UC San Diego
UC San Diego Electronic Theses and Dissertations

Title
Robust Distributed Control of Networked Systems with Linear Programming Objectives /

Permalink
https://escholarship.org/uc/item/13q5t307

Author
Richert, M.

Publication Date
2014

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA, SAN DIEGO

Robust Distributed Control of Networked Systems with Linear Programming Objectives

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Engineering Sciences (Mechanical Engineering)

by

Dean Richert

Committee in charge:

Professor Jorge Cortés, Chair
Professor Massimo Franceschetti
Professor Miroslav Krstic
Professor Sonia Martinez
Professor Jiawang Nie

2014

Copyright
Dean Richert, 2014
All rights reserved.

The dissertation of Dean Richert is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2014

DEDICATION

To my wife, my partner in crime for all of our adventures.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Thanks must first be directed to Prof. Jorge Cortés, my mentor for the past four years and doctoral committee chair. Jorge, your dedication to your students is unparalleled. The results of this thesis were made possible through the countless hours you devoted to meeting, teaching, and advising me. You truly embody the characteristics of both an excellent educator and a productive researcher. I sincerely appreciate all of the behind-the-scenes work you do for your students, such as securing funding, weekend Skype meetings, and Sunday svn commits.

The educational experience at the University of California, San Diego was top-notch. I attribute this to the world-class faculty, with a special mention of the other members on my doctoral committee: Prof. Sonia Martinez, Prof. Miroslav Krstic, Prof. Massimo Franceschetti, and Prof. Jiawang Nie.

Many insightful discussions with fellow students contributed to the ideas expressed in this thesis. To this end, I thank everyone in the Cortés-Martinez group, specifically Mike, Cameron, Bahman, Ashish, Andres, Eduardo, David, Hamed, Evan, and Beth.

Finally, I would like to acknowledge the unconditional support of my family. To Jane, you worked hard for four years so that I could pursue my degree, you took care of me when I couldn't take care of myself, and most importantly you loved me through it all. Mom and Dad, you provided me with more opportunities than I was able to take advantage of. To my in-laws, your persistent encouragement was much appreciated.

This work could never have been accomplished without the generous support of AFOSR Award FA9550-10-1-0499, NSF award CMMI-0908508, a Jacobs Fellowship, and an NSERC Postgraduate Scholarship.

Chapter 3, in part, is a reprint of the material [80] as it appears in 'Distributed linear programming and bargaining in exchange networks' by D. Richert and J. Cortés in the proceedings of the 2013 American Control Conference as well as [81] as it appears in 'Integral input-to-state stable saddle-point dynamics for distributed linear programming' by D. Richert and J. Cortés in the proceedings of the 2013 IEEE Conference on Decision and Control as well as [83] as it appears

in ‘Robust distributed linear programming’ by D. Richert and J. Cortés which was submitted to the IEEE Transactions on Automatic Control. The dissertation author was the primary investigator and author of these papers.

Chapter 4, in part, is a reprint of the material [85] as it appears in ‘Distributed event-triggered optimization for linear programming’ by D. Richert and J. Cortés which was submitted to the 2014 IEEE Conference on Decision and Control as well as [86] as it appears in ‘Distributed linear programming with event-triggered communication’ by D. Richert and J. Cortés which was submitted to the SIAM Journal of Control and Optimization. The dissertation author was the primary investigator and author of these papers.

Chapter 5, in part, is a reprint of the material [80] as it appears in ‘Distributed linear programming and bargaining in exchange networks’ by D. Richert and J. Cortés in the proceedings of the 2013 American Control Conference as well as the material [84] as it appears in ‘Distributed bargaining in dyadic-exchange networks’ by D. Richert and J. Cortés which was submitted to the IEEE Transactions on Control of Network Systems. The dissertation author was the primary investigator and author of this paper and unpublished material.

Chapter 6, in part, is a reprint of the material [79] as it appears in ‘Optimal leader allocation in UAV formation pairs under no-cost switching’ by D. Richert and J. Cortés in the proceedings of the 2012 American Control Conference as well as [78] as it appears in ‘Optimal leader allocation in UAV formation pairs under costly switching’ by D. Richert and J. Cortés in the proceedings of the 2012 IEEE Conference on Decision and Control as well as [82] as it appears in ‘Optimal leader allocation in UAV formation pairs ensuring cooperation’ by D. Richert and J. Cortés in Automatica. The dissertation author was the primary investigator and author of these papers.

VITA

2008	Bachelor of Science in Electrical Engineering, University of Calgary
2010	Master of Science in Electrical Engineering, University of Calgary
2014	Doctor of Philosophy in Engineering Science (Mechanical Engineering), University of California, San Diego

PUBLICATIONS

Journal publications:

D. Richert and J. Cortés, "Distributed bargaining in dyadic-exchange networks", *IEEE Transactions on Control of Network Systems*, submitted, 2014.

D. Richert and J. Cortés, "Distributed linear programming with event-triggered communication", *SIAM Journal of Control and Optimization*, submitted, 2014.

D. Richert and J. Cortés, "Robust distributed linear programming", *IEEE Transactions on Automatic Control*, under revision, 2014.

D. Richert and J. Cortés, "Optimal leader allocation in UAV formation pairs ensuring cooperation", *Automatica*, vol. 49, no. 11, 2013, pp. 3189-3198, 2013.

D. Richert, K. Masuad, and C. J. B. Macnab, "Discrete-time weight updates in neural adaptive control", *Soft Computing*, vol. 17, no. 3, 2013, pp. 431-444.

D. Richert, C. Macnab, and J. Pieper, "Adaptive haptic control for telerobotics transitioning between free, soft, and hard environments", *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 42, no. 3, 2012, pp. 558-570.

Conference proceedings:

D. Richert and J. Cortés, "Distributed event-triggered optimization for linear programming", *IEEE Conference on Decision and Control*, Los Angeles, California, USA, submitted, 2014.

D. Richert and J. Cortés, "Integral input-to-state stable saddle-point dynamics for distributed linear programming", *IEEE Conference on Decision and Control*, Florence, Italy, 2013, pp. 7480-7485.

D. Richert and J. Cortés, "Distributed linear programming and bargaining in exchange networks", *American Control Conference*, Washington, D.C., USA, 2013, pp. 4624-4629.

D. Richert and J. Cortés, "Optimal leader allocation in UAV formation pairs under costly switching", *IEEE Conference on Decision and Control*, Maui, Hawaii, USA, 2012, pp. 831-836.

D. Richert and J. Cortés, "Optimal leader allocation in UAV formation pairs under no-cost switching", *American Control Conference*, Montréal, Canada, 2012, pp. 3297-3302.

D. Richert, C. Macnab, and J. Pieper, "Adaptive control for haptics with time-delay", *IEEE Conference on Decision and Control*, Atlanta, Georgia, USA, 2010, pp. 3650-3655.

D. Richert, C. Macnab, and J. Pieper, "Force-force bilateral haptic control using adaptive backstepping with tuning functions", *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Montréal, Canada, 2010, pp. 341-346.

D. Richert, A. Beirami, and C. Macnab, "Neural-adaptive control of robotic manipulators using a supervisory inertia matrix", *International Conference on Autonomous Robots and Agents*, Wellington, New Zealand, 2009, pp. 634-639.

D. Richert and C. Macnab, "Direct adaptive force feedback for haptic control with time delay", *IEEE Toronto International Conference on Science and Technology for Humanity*, Toronto, Canada, 2009, pp. 893-897.

ABSTRACT OF THE DISSERTATION

Robust Distributed Control of Networked Systems with Linear Programming Objectives

by

Dean Richert

Doctor of Philosophy in Engineering Sciences (Mechanical Engineering)

University of California, San Diego, 2014

Professor Jorge Cortés, Chair

The pervasiveness of networked systems in modern engineering problems has stimulated the recent research activity in distributed control. Under this paradigm, individual agents, having access to partial information and subject to real-world disturbances, locally interact to achieve a common goal. Here, we consider network objectives formulated as a linear program where the individual agents' states correspond to components of the decision vector in the optimization problem. To this end, the first contribution we make is the development of a robust distributed continuous-time dynamics to solve linear programs. We systematically argue that the robustness properties we establish for this dynamics are as strong as can be expected for linear programming algorithms. The next contribution we

make is the design of a distributed event-triggered communication protocol for the aforementioned algorithm. We establish various state-based rules for agents to determine when they should broadcast their state, allowing us to relax the need for continual information flow between agents.

Turning our attention to a specific network control problem for which our algorithm can be applied, we consider distributed bargaining in exchange networks. In this scenario, agents autonomously form coalitions of size two (called a match) and agree on how to split a payoff between them. We emphasize fair and stable bargaining outcomes, whereby matched agents benefit equally from the collaboration and cannot improve their allocation by unilaterally deviating from the outcome. We synthesize distributed algorithms that converge to such outcomes. Finally, we focus on cooperation-inducing mechanisms to ensure that agents in a bargaining outcome effectively realize the payoff they were promised. As an illustrative example, we study how to allocate the leader role in unmanned aerial vehicle (UAV) formation pairs. We show how agents can strategically decide when to switch from leading to following in the formation to ensure that the other UAV cooperates.

Throughout the thesis, we emphasize the development of provably correct algorithms, making use of tools from the controls systems community such as Lyapunov analysis and the Invariance Principle. Simulations in distributed optimal control, multi-agent task assignment, channel access control in wireless communication networks, and UAV formations illustrate our results.

Chapter 1

Introduction

Networked systems, characterized by the interconnection of multiple components,[1] manifest themselves in many modern engineering applications as well as model various social, economic, and biological processes. To highlight the importance and prevalence of such systems let us outline some factors which evoke a network structure, beginning with the presence of a physical infrastructure. For example, the sparse distribution of physical generators, transmission lines, and loads in the electrical power grid constrain the supply, transmission, and use of power, inducing a network structure. Likewise, the roads that make up the transportation system admit limited paths between sources and destinations, dictating the flow of cars. Interestingly, a physical infrastructure need not be man-made, the human brain being an example where interactions between neurons occur via synapses. The availability of resources, or lack thereof, is another cause of a network structure. This is the case in wireless communication networks where power-limited devices cannot communicate with far away devices. Information, on the other hand, may be a scarce resource in a sensor network where the sensing ranges of each device are limited to local spatial neighborhoods. Of particular interest in the modern era is user privacy which may regulate the exchange of information to between trusting parties.

From a research perspective, the study of networked systems can be (for the

[1] In this thesis, we use the term component, subsystem, agent, player, and node interchangeably.

sake of this discussion) divided into modeling, design, and control. Scientists in sociology, biology, and economics are especially interested in the modeling aspect of a networked system as it allows them to predict future behaviors and better understand the underlying mechanisms that affect a network. For instance, in economic scenarios, game theory uses various behavioral models of players to predict the outcomes in strategic interactions. With regards to design, any network defined by a physical infrastructure provides an illustrative example. Building upon our previous example of electrical power systems, engineers must decide which components must be installed to meet consumer demand and maximize profit. Where to place sensors and actuators is another network design problem. Finally, the control of a networked system refers to the implementation of algorithms to drive the system to a desired state or behavior.

In this thesis, our focus is predominantly on the control of, and thus provable correct algorithm design for, networked systems. Broadly speaking, a network control algorithm can be understood as a policy that uses the available information in the network to dictate the behavior of subsystems. Technically speaking, it is a mapping from the available information to the network state evolution. To this end, two main architectures exist for the control of networked systems. On one hand, a centralized approach identifies a single entity to generate and transmit the control signal for all other components in the system. Consequently, the implementation of a centralized controller requires that all the information in the network be made accessible to the controller and there must exist direct communication between the controller and each subsystem. The advantage of a centralized control lies in the relative simplicity of designing provably correct algorithms. Essentially, the control design may ignore the inherent network structure of the system and simply views it as a high-dimensional dynamical system (to the extent, at least, that any physical constraints remain satisfied). For some applications, this is a perfectly valid approach, but for others it may be inefficient, infeasible, or simply undesirable. On the other hand, a distributed approach, as we consider in this thesis, allows each subsystem to compute and implement its own control signal based on locally available information. Distributed control algorithms enjoy the property of being naturally scalable with respect to the network size. That is, the complexity from the perspective of any given subsystem does not increase even when the dimension of the network state does. On a related note, networks controlled by distributed algorithms are naturally adaptable to changes in the network structure; the desired global behavior of the network being preserved when nodes are added or deleted. This phenomena is seen in robotic deployment applications where agents naturally adapt to achieve optimal deployment when some robots fail or new robots join the network. Finally, a distributed approach allows for a more efficient (and in many cases, a feasible) use of resources. Of course, the main challenge faced by a controls engineer in developing a distributed algorithm is precisely that the control signal for each subsystem may only be generated using locally available information. Interestingly, there is rarely a straightforward or intuitive mapping between the local behavior of a single component and the emergent global behavior of the network. This make the design and analysis of distributed control algorithms a particularly rich area of research, promising to be relevant for years to come.

The remainder of this introduction outlines the specific problems we consider in this thesis, places these problems in the context of the current literature, and summarizes the contributions made.

1.1 Specific problems considered

The overarching problem we explore in this thesis is how to systematically generate robust, distributed, and implementable control algorithms to achieve a desired network objective. We then ask to what extent the approach we develop can be applied in network bargaining scenarios. Our investigation into this second problem brings up some independently interesting questions that we also address.

The design of a network control algorithm must begin with a well-posed description of the network objective. That is, given a criteria and a current state, it should be possible to decisively conclude whether or not the network objective has been met. For this purpose, a particularly powerful and widespread framework is mathematical optimization. An optimization problem effectively captures the network objective in terms of a performance measure while enforcing the physical system constraints. It can be argued that most, if not all, well-posed network control problems can be formulated as an optimization problem. Within the field of optimization, linear programming considers linear objectives minimized over a convex polyhedron of feasible states. The use of linear programming to solve many real-life problems has been exhaustively demonstrated in the past century. Moreover, linear programming is a fundamental tool in mathematical optimization that demands specific attention. Thus, our starting point in this thesis is a network objective formulated as a linear program. The first problem we consider is, under certain assumptions on the network structure, how to design a distributed continuous-time dynamics to drive the network state to a solution of general linear programs. In this context, we consider the state of each subsystem as a subset of the decision vector in the linear program. As a follow-up to this first problem, we consider the case when various disturbances affect the network and explore under what conditions we can guarantee that the network objective is still achieved.

Motivated by implementation considerations for our continuous-time algorithm, we then turn our attention to developing a realistic communication protocol that agents may use to broadcast their state information to their neighbors. Our goal is to relax the need for continuous information flow between agents in the network. Simply deriving the Euler discretization of the continuous-time algorithm is not satisfactory to us for two reasons. One, the selection of the stepsizes to guarantee convergence has to take into account worst-case situations, leading to an inefficient use of the network resources. Second, the synchronization of broadcasts between agents is not realistic. Rather, we seek to design opportunistic event-triggered and asynchronous communication among agents.

The second half of this thesis considers more specific network control problems to which the developments in the first half are directly applicable. Our motivation is resource-constrained networks where collaboration between subsystems gives rise to a more efficient use of these resources. To this end, we view each subsystem as a player in a type of coalitional game where neighboring players are interested in forming coalitions of size two, called a match. To each potential matching between agents there is an associated transferable utility which can be divided between them. The problem we consider is how agents can decide with whom to collaborate with and how to allocate the utility. We call an algorithmic solution to this problem *bargaining between agents*. The type of outcomes we are interested in are called Nash bargaining outcomes, which combine the notion of both stability and fairness. In this context, a stable outcome means that none of the agents benefit by unilaterally deviating from their match. However, even within the set of stable outcomes, there may be some outcomes that benefit certain agents more than others. Thus, a fair outcome is one where every two matched agents benefit equally from the match.

Our study of network bargaining brings up the question of how agents may realistically realize the portion of the utility that they were promised. In particular, we look for mechanisms that can incentive matched agents to honor their promises. As an illustrative example, we consider pairwise formations between unmanned aerial vehicles (UAVs), where an agent gains a fuel benefit by flying in the wake of another (i.e., a reduction in aerodynamic drag). When agents are noncooperative, the potential benefits of flying in formation bring up the issue of how to distribute the leader task. The goal then is to develop cooperation-inducing leader allocations so that UAVs are able to realize the potential benefits of collaboration.

1.2 Literature review

Here we review the current state of the art in the area of network control, making specific emphasis on the literature that pertains to the problems we have outlined above. Accordingly, the topics covered in this review include distributed optimization algorithms, robustness notions, event-triggered implementations, and bargaining in networks.

1.2.1 Distributed optimization algorithms

Mathematical optimization plays an important role in a wide variety of networked systems applications, see e.g., [14, 34, 55, 59, 95] and references therein. Linear programs, in particular, can describe many network objectives including perimeter patrolling [2], task assignment [51, 64], operator placement [27], process control [56], routing in communication networks [100], and portfolio optimization [96].

This relevance has historically driven the design of efficient methods to solve general convex optimization problems [15, 23] and linear programs in particular [20, 35]. More specifically related to this work, the design of distributed optimization algorithms has seen significant interest in recent decades [18, 22, 68, 116]. Dual-decomposition [92, 113] is a popular method that exploits the natural separability of the dual problem to allow for a distributed implementation. In order for each iteration to be well-defined, strict convexity of the objective function is required for this method. Conversely, the alternating direction method of multipliers [58, 109] is applicable to non-strictly convex problem. However, each iteration in this approach requires a projection of the decision variable onto the feasible set and thus only amenable to a distributed implementation in some special cases. Auction algorithms [11, 16] are capable of computing in a distributed way approximate solutions of optimization problems. Subgradient projection algorithms [52, 67, 117] build on consensus-based dynamics [24, 63, 73, 76] whereby individual agents agree on the global solution to the optimization problem. This is a major difference with respect to our work here, in which each individual agent computes only its own component of the solution vector by communicating with its neighbors. In our work the messages transmitted over the network are independent of the size of the solution vector, and hence scalable, a property which would not be shared by a consensus-based distributed optimization method. Finally, the saddle-point approach, as we consider in this thesis, was applied in [39, 41, 106].

For the reasons discussed above, many of the aforementioned distributed optimization algorithms are, interestingly, not applicable to general linear programs. Moreover, centralized methods to solve linear programs, such as the simplex method [35], are not easily distributed. The works [25, 72, 112] propose algorithms specifically designed for distributed linear programming. As in consensus-based approaches, in these works the goal is for agents to agree on the global solution. Closer to our approach, although without considering equality constraints, the works [8, 39] build on the saddle-point dynamics of a smooth Lagrangian function to propose an algorithm for linear programming. The resulting dynamics is discontinuous in both the primal and dual variables because of the projections taken to keep the evolution within the feasible set. Both works establish convergence in the primal variables under the assumption that the solution of the linear program is unique [8] or that Slater's condition is satisfied [39], but do not characterize the properties of the final convergence point in the dual variables, which might indeed not be a solution of the dual problem. To our knowledge, the rigorous investigation into algorithmic robustness for distributed linear programming does not exist in the literature.

1.2.2 Robustness notions

Another point of connection of the present treatment with the literature is the body of work on robustness of dynamical systems against disturbances. In general, disturbances may affect both a system's state or its dynamics and, in our context, may model (among others) communication noise, modeling errors, link failures, or actuator limitations. For linear systems, the systematic design of robust controllers is well documented (see e.g., [115]). We explore the properties of our proposed dynamics with respect to notions appropriate for nonlinear systems, such as robust asymptotic stability [26], input-to-state stability (ISS) [97], and integral input-to-state stability (iISS) [6]. Considering robustness notions relevant to networked systems, [101] studies the convergence of discrete-time gradient algorithms under asynchronous communication and link failures. The work [74] established various robustness results for consensus dynamics in networks with switching topologies, including the deletion and addition of agents, and time delays. Regarding distributed optimization, [66] developed algorithms for time-varying directed graphs. We note that the term 'robust optimization' often employed in the literature, see e.g. [19], refers instead to worst-case optimization problems where uncertainty in the data is explicitly included in the problem formulation. Thus, 'robust' in that setting refers to the problem formulation and not to the actual algorithm employed to solve the optimization which is our interest in this thesis.

1.2.3 Event-triggered implementations

All of the algorithms mentioned previously are implemented in either continuous or discrete time, the latter with time-dependent stepsizes that are independent of the network state. Instead, event-triggered control seeks to opportunistically adapt the execution to the network state by trading computation and decision-making for less communication, sensing, or actuation effort while guaranteeing a desired level of performance, see e.g., [46, 62, 108]. Self-triggered control [7, 107] is a related approach where, rather than continuously measuring the state to detect an event, the next control update is pre-determined based on the most recent state measurement. In both cases, a key design objective, besides asymptotic convergence, is to ensure the lack of an infinite number of updates in any finite time interval of the resulting strategy. The design of event-triggered control algorithms for networked systems requires that triggers are locally detectable. Some works in this area of distributed event-triggered control include [37, 62]. One challenge faced in these works is that the agent broadcasts become asynchronous. A few works [53, 104] have explored the design of distributed event-triggered optimization algorithms for multi-agent systems. A major difference between event-triggered stabilization and optimization is that in the former the equilibrium is known a priori, whereas in the latter the determination of the equilibrium point is the objective itself. Adding to the complexity, the algorithm that we seek to design an event-triggered implementation for is a state-dependent switched dynamical systems. To our knowledge, this thesis is the first to consider such problems.

1.2.4 Bargaining in networks

In a coalitional game, two fundamental questions are (i) which coalitions will form and (ii) how will a payoff associated to each coalition be distributed amongst its members. We call a process whereby agents seek to autonomously answer these questions *bargaining*, and in a network the set of feasible coalitions is defined by the network structure. In engineering, examples of where network bargaining problems arise are plentiful for communication networks [3, 4, 5, 61, 70, 71, 91, 99, 114] and also exist for mobile robot coordination [75, 103], formations of UAVs [82], and large-scale data processing [87]. Several sociology [28] and economic applications, such as matching in labor markets [88], also exist. Bargaining problems of the type we consider are posed on dyadic-exchange networks, so called because agents can match with at most one other agent [30]. Bipartite matching and assignment problems [14] are special cases of the dyadic-exchange network. Nash bargaining outcomes, as we consider in this thesis, are an extension to network games of the classical two-player Nash bargaining solution [65]. Centralized methods for finding such outcomes were developed in [12, 54]. In terms of discrete-time distributed implementations, the work [10] provides dynamics that, given a matching, converge to fair (balanced) allocations. On the other hand, [13] provides discrete-time dynamics that converge to Nash outcomes. Our work here looks independently at continuous-time dynamics for each of stable, balanced, and Nash outcomes.

An element of the Nash bargaining problem that is often ignored is the mechanism by which agents use to transfer, or realize, the share of the payoff that they were promised. An illustrative example of when this question is relevant is in UAV formation flying where the payoff of a formation is the fuel savings. The energy savings of flying in formation are apparent in flocks of birds [50, 110]. In theory, the same benefits exist for formations among UAVs [21, 42]. Moreover, recent improvements in technology make it possible to realize these fuel savings [94, 102]. Here, we take inspiration from [77], who study formation creation in groups of UAVs, and we examine how collaboration can be enforced among (not necessarily cooperative) agents through appropriately designed protocols. In our model for UAV behavior, agents are incentivized to remain in formation, analogously to marginal cost pricing schemes in game theory [69].

1.3 Contributions made

This section summarizes the contributions made by this thesis to the body of research on the control of networked systems. In keeping with the theme of this introduction, we categorize our contributions into those related to robust distributed linear programming, event-triggered optimization, network bargaining, and cooperation inducing mechanisms.

1.3.1 Robust distributed linear programming

Regarding robust distributed linear programming, our first contribution is the design of a continuous-time saddle-point dynamics (that is, gradient descent in one variable, gradient ascent in the other) associated with a novel nonsmooth modified Lagrangian. It should be noted that, in general, saddle points are only guaranteed to be stable (and not necessarily asymptotically stable) for the corresponding saddle-point dynamics. Nevertheless, in our case, we are able to establish the global asymptotic stability of the (possibly unbounded) set of primal-dual solutions of general linear programs and, moreover, the pointwise convergence of the trajectories. Our proof of convergence reveals that knowledge of a global parameter is necessary to guarantee the convergence of the saddle-point dynamics. To circumvent this need, we propose an alternative discontinuous saddle-point dynamics that does not require such knowledge and is fully distributed over a network. We show that the discontinuous dynamics share the same convergence properties of the regular saddle-point dynamics by establishing that, for sufficiently large values of the global parameter, the trajectories of the former are also trajectories of the latter. The two central advantages to our methodology is that (i) we are able prove asymptotic stability of the discontinuous dynamics without establishing any regularity conditions on the switching behavior of the system and (ii) it allows for the characterization of novel and relevant robustness properties. This latter point bring us to our next contribution, which pertains to the characterization of the robustness properties of the discontinuous saddle-point dynamics against disturbances and link failures. We establish that no continuous-time algorithm that solves general linear programs can be input-to-state stable (ISS). As our technical approach shows, this fact is due to the intrinsic properties of the primal-dual solution sets of linear programs. Nevertheless, when the set of primal-dual solutions is compact, we show that our discontinuous saddle-point dynamics possesses an ISS-like property against small constant disturbances and, more importantly, is integral input-to-state stable (iISS) – and thus robust to finite energy disturbances. We conclude that one cannot expect better disturbance rejection properties from a linear programming algorithm than those we establish for our discontinuous saddle-point dynamics. These results allow us to establish the robustness of our dynamics against disturbances of finite variation as well as communication failures between agents modeled by recurrently connected graphs.

1.3.2 Distributed event-triggered optimization

Here we summarize the contributions made towards distributed event-triggered optimization. We assume that agents implement the continuous-time dynamics that we previously developed, but instead of continuous information flow between agents they use a sample-and-hold value of their neighbors' state. Thus, our main contribution is the synthesis of suitable criteria used by agents to opportunistically determine when they should update their state information to ensure convergence. Because of the technical complexity involved in solving this challenge, we first design a centralized criteria and then extend them to distributed ones. The characterization of the convergence properties of the centralized implementation is challenging because the original continuous-time dynamics is discontinuous in the agents' state and its final convergence value (being the solution of the optimization problem) is not known a priori, which further complicates the identification of a common smooth Lyapunov function.

In our distributed event-triggered communication law, agents use local information to determine when to broadcast their individual state. Our strategy to accomplish this is to investigate to what extent the centralized triggers can be implemented in a distributed way and modify them when necessary. In doing so, we face the additional difficulty posed by the fact that the mode switches associated to the discontinuity of the original dynamics are not locally detectable by individual agents. Moreover, the distributed character of the agent triggers leads to asynchronous state broadcasts, which poses an additional challenge for both design and analysis. The main result we prove establishes the asymptotic convergence of the distributed implementation and identifies sufficient conditions for executions to be persistently flowing (that is, state broadcasts that are separated by a uniform time infinitely often). As a byproduct of using a hybrid systems modeling framework in our technical approach, we are also able to guarantee that the global asymptotic stability of the proposed distributed algorithm is robust to small enough perturbations.

1.3.3 Network bargaining

Next, we contribute a provably correct distributed continuous-time dynamics to find Nash bargaining solutions and show its application to a wireless communication scenario. In our design process we develop dynamics to find each of stable, balanced, and ultimately Nash outcomes. The problem formulation we provide reveals that finding a stable outcome is combinatorial in the number of edges in the network. Nevertheless, we prove a correspondence between the existence of stable outcomes and the solutions of a linear program, making the problem tractable. We show how the application of the distributed linear programming algorithm discussed in Section 1.3.1 can be used by agents to find stable outcomes using only local information. Turning our attention to balanced outcomes, we show how finding them requires agents to solve a system of coupled nonlinear equations. We define local (with respect to 2-hop information) error functions that measure how far matched agents' allocations are from being balanced. Our proposed algorithm has agents adjust their allocations based on the negative of their balancing errors. We combine the two aforementioned dynamics in such a way that their aggregate finds Nash outcomes. Since the dynamics we propose satisfy certain regularity conditions, its robustness to perturbations is guaranteed. Also, our presentation of the material is done from a control and dynamical systems perspective. Thus, we provide a new understanding of the problem and contribute a robust framework for extensions to this work.

1.3.4 Cooperation inducing mechanisms

Turning our attention then to mechanisms that induce cooperation between matched agents, we consider a specific UAV application. Our contributions to this end pertain to the modeling, analysis, and design of UAV formation pairs for optimal point-to-point reconfiguration. Regarding modeling, we introduce the notion of a UAV formation pair as a collection of distances (or leader allocations) in a line along which each one must lead. We also define a cost-to-target function that measures the total fuel consumed along the trajectory. We model the compliance of a UAV to a leader allocation via a parameter, $\varepsilon \geq 0$, which quantifies the cost gain that the UAV will forgo before breaking a formation. With these elements, we formulate the problem of finding an optimal leader allocation among those that induce cooperation. This problem is nonconvex in one variable (finding the optimal leader allocation given a fixed number of switches) and combinatorial in the other (finding the optimal number of leader switches).

The remaining contributions concern the analysis of this optimization problem and the design of algorithms that converge to a solution. When switching the lead has no cost, we design the COST REALIZATION ALGORITHM to determine an optimal cooperation-inducing leader allocation. When switching the lead is costly, we restrict the feasible set of leader allocations to mimic those of the solution provided by the COST REALIZATION ALGORITHM. Remarkably, we show that the restriction convexifies the feasible set of the original nonconvex problem (for finding the optimal leader allocation given a fixed number of switches) while maintaining its optimal value. Turning our attention then to the problem of finding the optimal number of leader switches, we establish a quasiconvexity-like property of the optimal value of the problem as a function of the number of leader switches. This property allows us to design the BINARY SEARCH ALGORITHM, which finds the optimal number in logarithmic time.

Throughout the thesis, various simulations demonstrate our algorithms and verify our results. We specifically look at a distributed optimal control example, a distributed task assignment example, and a wireless communication example.

1.4 Organization

In Chapter 2, we introduce some notation and basic notions required to understand the technical analysis in this thesis. In Chapter 3, we introduce the problem of solving a linear program in a network system. The development of a robust and distributed linear programming dynamics happens in this chapter. Next, Chapter 4 develops an event-triggered implementation of the previously proposed linear programming dynamics. As an application of the distributed linear programming dynamics, in Chapter 5 we consider the network bargaining problem. Here, dynamics to find stable, balanced, and Nash outcomes are derived. To conclude the technical portion of this thesis, Chapter 6 considers the problem of UAVs executing a formation while inducing cooperation. Chapter 7 summarizes our contributions and highlights some interesting areas of future work.

Chapter 2

Preliminaries

This chapter contains introductory material on notation and basic notions on mathematical analysis, set-valued and nonsmooth analysis, set-valued dynamical systems, hybrid systems, optimization, and graph theory.

2.1 Notation and basic notions on mathematical analysis

The sets of real, natural, even, and odd numbers are given by $\mathbb{R}, \mathbb{N}, \mathbb{E}, \mathbb{O}$, respectively. We let $\mathbb{1}_p \in \mathbb{R}^p$ denote the vector of ones. For $x \in \mathbb{R}^p$, $x \geq 0$ (resp. $x > 0$) means that all components of x are nonnegative (resp. positive) and we define $\max\{0, x\} = (\max\{0, x_1\}, \ldots, \max\{0, x_p\}) \in \mathbb{R}^p$. We use $\|x\|$ and $\|x\|_\infty$ to denote the 2- and ∞-norms in $\mathbb{R}^p$. Given $x, y \in \mathbb{R}^p$, the Euclidean distance between them is denoted $d(x, y) := \|x - y\|$. The Euclidean distance from a point $x \in \mathbb{R}^p$ to a set $S \subseteq \mathbb{R}^p$ is defined as

$$\|x\|_S = \min_{y \in S} \|x - y\|.$$

The following is a useful inequality for bounding bilinear forms.

Lemma 2.1.1. (Young's inequality [45]). *Let $x \in \mathbb{R}^{p_1}$, $A \in \mathbb{R}^{p_1 \times p_2}$, $y \in \mathbb{R}^{p_2}$. Then, for any $\varepsilon > 0$,*

$$x^T A y \leq \frac{\varepsilon}{2} x^T A^T A x + \frac{1}{2\varepsilon} y^T y.$$

For $n \in \mathbb{R}$ and set $S \subset \mathbb{R}^p$, the shorthand notation $S_{\geq n}$ (resp. $S_{>n}$) is used to denote the set $\{s \in S : s \geq n\}$ (resp. $\{s \in S : s \geq n\}$). For a set $S \subseteq \mathbb{R}^p$, its intersection with the nonnegative orthant is denoted $S_+ := S \cap \mathbb{R}^p_{\geq 0}$. This notation is applied analogously to vectors and scalars. The boundary of S is denoted $\text{bd}(S) \subseteq \mathbb{R}^p$ and $\text{int}(S) \subseteq \mathbb{R}^p$ is its interior. We use $\bar{S}$ to denote the closure of S. The set S is convex if it fully contains the segment connecting any two points in S. The convex hull of S is the smallest convex set that fully contains S and is denoted $\text{co}(S)$. If S is a finite set, the number of elements in S is given by $|S|$. Given sets $S_1, S_2 \subseteq \mathbb{R}^p$, we use $S_1 \setminus S_2$ to denote the points in S_1 that are not in S_2. The set $\mathbb{B}(x, \delta) \subseteq \mathbb{R}^p$ is the open ball centered at $x \in \mathbb{R}^p$ with radius $\delta > 0$.

Given a matrix $A \in \mathbb{R}^{p_1 \times p_2}$, for $i \in \{1, \ldots, p_2\}$ its i^{th} row is denoted $A_i \in \mathbb{R}^{p_1}$ and for $j \in \{1, \ldots, p_1\}$ its (i,j)-element is denoted $a_{i,j} \in \mathbb{R}$. When $A \in \mathbb{R}^{p \times p}$ is square, its spectral radius is

$$\rho(A) := \max_{i \in \{1,\ldots,p\}} |\lambda_i|,$$

where $\lambda_1, \ldots, \lambda_p$ are the eigenvalues of A. We call $A = A^T \in \mathbb{R}^{p \times p}$ positive semi-definite (resp. positive definite) if $x^T A x \geq 0$ (resp. $x^T A x > 0$) for all $x \in \mathbb{R}^p \setminus \{0\}$ and we write $A \succeq 0$ (resp. $A \succ 0$). If $A \succeq 0$, then $x^T A x \leq \rho(A) x^T x$. Moreover, for any matrix $A \in \mathbb{R}^{p_1 \times p_2}$, $\rho(A^T A) = \rho(A A^T)$. The following is a useful result characterizing the eigenvalues of a matrix when some of its rows and columns are zeroed out.

Theorem 2.1.2. (Cauchy Interlacing Theorem [48, Theorem 4.3.15]). *For a matrix $0 \preceq A \in \mathbb{R}^{p \times p}$, let $0 \leq \lambda_1 \leq \cdots \leq \lambda_p$ denote its eigenvalues. For $d \in \{1, \ldots, p\}$, let A_d be the matrix obtained by zeroing out the d^{th} row and column of A, and let $0 = \mu_1 \leq \cdots \leq \mu_p$ denote its eigenvalues. Then $\mu_1 \leq \lambda_1 \leq \mu_2 \leq \lambda_2 \leq \cdots \leq \mu_p \leq \lambda_p$.*

The domain of a function $f : X \to Y$ is denoted $\text{dom}(f) := X$. A function $f : \mathbb{R}^p \to \mathbb{R}$ is positive definite with respect to a set $S \subset \mathbb{R}^p$ if $f(x) = 0$ for all $x \in S$ and $f(x) > 0$ for all $x \notin S$. If $S = \{0\}$, we refer to f as positive definite. f is radially unbounded with respect to S if $f(x) \to \infty$ when $\|x\|_S \to \infty$. If $S = \{0\}$, we refer to f as radially unbounded. We call f proper with respect to S if it is both positive definite and radially unbounded with respect to S. For $c \in \mathbb{R}$, we denote by $f^{-1}(\leq c) = \{x \in \text{dom}(f) \mid f(x) \leq c\}$ the c-sublevel set of f. If f is radially unbounded then $f^{-1}(\leq c)$ is compact. The function $f : X \to \mathbb{R}$ is convex if $X \subseteq \mathbb{R}^p$ is convex and $f(\mu x + (1-\mu)y) \leq \mu f(x) + (1-\mu)f(y)$ for all $x, y \in X$ and $\mu \in [0, 1]$. f is concave if $-f$ is convex. The function $L : X \times Y \to \mathbb{R}$ defined on the convex set $X \times Y \subseteq \mathbb{R}^{p_1} \times \mathbb{R}^{p_2}$ is convex-concave if $x \mapsto L(x, y)$ is convex and $y \mapsto L(x, y)$ is concave. A point $(\bar{x}, \bar{y}) \in X \times Y$ is a saddle point of L if $L(x, \bar{y}) \geq L(\bar{x}, \bar{y}) \geq L(\bar{x}, y)$ for all $(x, y) \in X \times Y$. The class of $\mathcal{K}$ functions is composed by functions of the form $[0, \infty) \to [0, \infty)$ that are continuous, zero at zero, and strictly increasing. The subset of class $\mathcal{K}$ functions that are unbounded are called class $\mathcal{K}_\infty$. A class $\mathcal{KL}$ function $[0, \infty) \times [0, \infty) \to [0, \infty)$ is class $\mathcal{K}$ in its first argument and continuous, decreasing, and converging to zero in its second argument.

2.2 Set-valued and nonsmooth analysis

An understanding of set-valued and nonsmooth functions will prove to be instrumental in both the derivation and analysis of various algorithms we introduce in this paper. In particular, certain extensions of the derivative will allow us to design gradient algorithms based on functions that are otherwise not differentiable (that is, nonsmooth). Incidentally, the resulting dynamics is set-valued which highlights a connection between these two notions.

We begin by introducing some notions on set-valued analysis, following closely the exposition of [9]. A set-valued map $F : \mathbb{R}^p \rightrightarrows \mathbb{R}^p$ maps elements in $\mathbb{R}^p$ to subsets of $\mathbb{R}^p$. F is locally bounded if for every $x \in \mathbb{R}^p$ there exists an $\varepsilon > 0$ such that $F(\mathbb{B}(x, \varepsilon))$ is bounded. The set-valued map $F : X \subset \mathbb{R}^p \rightrightarrows \mathbb{R}^p$ is upper semi-continuous if for all $x \in X$ and $\varepsilon > 0$ there exists $\delta_x \geq 0$ such that $F(y) \subseteq F(x) + \mathbb{B}(0, \varepsilon)$ for all $y \in \mathbb{B}(x, \delta_x)$. Upper semi-continuity, as we will see in the following section, endows certain regularity on dynamics defined by set-valued maps. Conversely, F is lower semi-continuous if for all $x \in X$, $\varepsilon \geq 0$, and any open set S intersecting $F(x)$ there exists a $\delta \geq 0$ such that $F(y)$ intersects S for all $y \in \mathbb{B}(x, \delta)$. If F is both upper and lower semi-continuous then it is continuous.

Let us now review some basic notions in nonsmooth analysis. Our presentation follows [29]. A function $f : \mathbb{R}^p \to \mathbb{R}$ is locally Lipschitz at $x \in \mathbb{R}^p$ if there exist $\delta_x > 0$ and $L_x \geq 0$ such that $|f(y_1) - f(y_2)| \leq L_x \|y_1 - y_2\|$ for $y_1, y_2 \in \mathbb{B}(x, \delta_x)$. If f is locally Lipschitz at all $x \in \mathbb{R}^p$, we refer to f as locally Lipschitz. If f is convex, then it is locally Lipschitz. A locally Lipschitz function is differentiable almost everywhere. Let $\Omega_f \subseteq \mathbb{R}^p$ be then the set of points where f is not differentiable. A natural extension of the derivative of f at $x \in \Omega_f$, called the generalized gradient, is the convex combination of the set of limit points of the gradient of f at neighboring points where f is differentiable. Formally speaking, the generalized gradient of f at $x \in \mathbb{R}^p$ is

$$\partial f(x) = \mathrm{co}\Big\{ \lim_{i \to \infty} \nabla f(x_i) : x_i \to x, x_i \notin S \cup \Omega_f \Big\},$$

where $S \subset \mathbb{R}^p$ is any set with zero Lebesgue measure. The definition of the generalized gradient is independent of the choice of set S, and this set is typically chosen to simplify computation. The following result states some useful properties of the generalized gradient.

Lemma 2.2.1. (Properties of the generalized gradient). *If $f : \mathbb{R}^p \to \mathbb{R}$ is locally Lipschitz at $x \in \mathbb{R}^p$, then $\partial f(x)$ is nonempty, convex, and compact. Moreover, $x \mapsto \partial f(x)$ is locally bounded and upper semi-continuous.*

The above result will allow us, in subsequent sections, to ensure the existence of solutions for gradient algorithms based on locally Lipschitz functions. A critical point $x \in \mathbb{R}^p$ of f satisfies $0 \in \partial f(x)$. For a convex function f, the first-order condition of convexity states that $f(y) \geq f(x) + (y - x)^T g$ for all $g \in \partial f(x)$ and $x, y \in \mathbb{R}^p$. For $f : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ and $(\bar{x}, \bar{y}) \in \mathbb{R}^p \times \mathbb{R}^p$, we use $\partial_x f(\bar{x}, \bar{y})$ and $\partial_y f(\bar{x}, \bar{y})$ to denote the generalized gradients of the maps $x \mapsto V(x, \bar{x})$ at $\bar{x}$ and $y \mapsto V(\bar{x}, y)$ at $\bar{y}$, respectively.

2.3 Set-valued dynamical systems

Some of the dynamics we propose in this thesis are defined by set-valued flow fields, so we introduce some basic concepts for these types of systems. The tutorial [32] provides an in depth look at this field.

A time-invariant set-valued dynamical system is represented by the differential inclusion

$$\dot{x} \in F(x), \tag{2.1}$$

where $F : \mathbb{R}^p \rightrightarrows \mathbb{R}^p$ is a set valued map. If F is locally bounded, upper semi-continuous and takes nonempty, convex, and compact values, then from any initial condition in $\mathbb{R}^p$, there exists an absolutely continuous curve $x : \mathbb{R}_{\geq 0} \to \mathbb{R}^p$ satisfying (2.1) almost everywhere. In this case, we call $t \mapsto x(t)$ either a solution or a trajectory of (2.1). The solution is maximal if it cannot be extended forward in time. The set of equilibria of F is defined as $\{x \in \mathbb{R}^p \mid 0 \in F(x)\}$. A set $\mathcal{M}$ is strongly (resp. weakly) invariant with respect to (2.1) if, for each $x_0 \in \mathcal{M}$, $\mathcal{M}$ contains all (resp. at least one) maximal solution(s) of (2.1) with initial condition x_0. The Lie derivative is a powerful mathematical tool that captures the evolution of a real-valued function along the trajectories of a set-valued dynamical system. Consider first the simple case when the dynamics is given by a differential equation, $\dot{x} = f(x)$, with $f : \mathbb{R}^p \to \mathbb{R}^p$. The evolution of a differentiable function $V : \mathbb{R}^p \to \mathbb{R}$ along the trajectories of this dynamics is the Lie derivative of V along f,

$$\mathcal{L}_f V(x) = \nabla V(x)^T f(x).$$

In this case, it is easy to see that the Lie derivative is equivalent to the directional derivative of V along f. We are interested in the extension of this concept when $V : \mathbb{R}^p \to \mathbb{R}$ is locally Lipschitz and the dynamics are set-valued, as in (2.1). In this case, the Lie derivative of V along the trajectories of (2.1) is defined as

$$\mathcal{L}_F V(x) = \{a \in \mathbb{R} : \exists v \in F(x) \text{ s.t. } v^T \zeta = a \; \forall \zeta \in \partial V(x)\}.$$

Again, if V is differentiable, then the above definition simplifies to

$$\mathcal{L}_F V(x) = \{\nabla V(x)^T v : v \in F(x)\}.$$

If $\mathcal{L}_F V(x) \subset (-\infty, 0]$ for all $x \in \mathbb{R}^p$, then $t \mapsto F(x(t))$ is monotonically non-increasing where $t \mapsto x(t)$ is any trajectory of (2.1). When $\mathcal{L}_F V(x) \subset (-\infty, 0]$ for all $x \in \mathbb{R}^p$ the set $V^{-1}(x(0))$ is strongly invariant with respect to (2.1). The following result helps establish the asymptotic set-convergence of (2.1) by way of the Lie derivative and the invariance, with respect to (2.1), of certain sets.

Theorem 2.3.1. (Set-valued LaSalle Invariance Principle). *Let $X \subset \mathbb{R}^p$ be compact and strongly invariant with respect to* (2.1). *Assume $V : \mathbb{R}^p \to \mathbb{R}$ is differentiable and F is locally bounded, upper semi-continuous and takes nonempty, convex, and compact values. If $\mathcal{L}_F V(x) \subset (-\infty, 0]$ for all $x \in X$, then any solution of* (2.1) *starting in X converges to the largest weakly invariant set $\mathcal{M}$ contained in $\overline{\{x \in X : 0 \in \mathcal{L}_F V(x)\}}$.*

Differential inclusions are especially useful to handle differential equations with discontinuities. For example, let $f : X \subset \mathbb{R}^p \to \mathbb{R}^p$ be a piecewise continuous vector field and consider

$$\dot{x} = f(x). \tag{2.2}$$

The classical notion of solution is not applicable to (2.2) because of the discontinuities. Instead, consider the Filippov set-valued map associated to f, defined by

$$\mathcal{F}[f](x) := \overline{\text{co}}\Big\{ \lim_{i\to\infty} f(x_i) : x_i \to x, x_i \notin \Omega_f \Big\}, \tag{2.3}$$

where Ω_f are the points where f is discontinuous. One can show that the set-valued map $\mathcal{F}[f]$ is locally bounded, upper semi-continuous and takes nonempty, convex, and compact values, and hence solutions exist to

$$\dot{x} \in \mathcal{F}[f](x), \tag{2.4}$$

starting from any initial condition. The solutions of (2.2) in the sense of Filippov are, by definition, the solutions of the differential inclusion (2.4).

2.4 Hybrid systems

These basic notions on hybrid systems follow closely the exposition found in [44]. A hybrid (or cyber-physical) system is a dynamical system whose state may evolve according to (i) a differential equation $\dot{x} = f(x)$ when its state is in some subset, C, of the state-space and (ii) a difference equation $x^+ = g(x)$ when its state is in some other subset, D, of the state-space. Thus, we may represent a hybrid system by the tuple $H = (f, g, C, D)$ where $f : \mathbb{R}^p \to \mathbb{R}^p$ (resp. $g : \mathbb{R}^p \to \mathbb{R}^p$) is called the *flow map* (resp. *jump map*) and $C \subseteq \mathbb{R}^p$ (resp. $D \subseteq \mathbb{R}^p$) is called the *flow set* (resp. *jump set*). Formally speaking, the evolution of the states of H are governed by the following equations

$$\dot{x} = f(x), \quad x \in C, \tag{2.5a}$$

$$x^+ = g(x), \quad x \in D. \tag{2.5b}$$

A *compact hybrid time domain* is a subset of $\mathbb{R}_{\geq 0} \times \mathbb{N}$ of the form

$$E = \cup_{j=0}^{J-1}([t_j, t_{j+1}], j),$$

for some finite sequence of times $0 = t_0 \leq t_1 \leq \cdots \leq t_J$. It is a hybrid time domain if for all $(T, J) \in E$, $E \cap ([0, T] \times \{0, \ldots, J\})$ is a compact hybrid time domain. A function ψ is a solution to the hybrid system (2.5) if

(i) for all $j \in \mathbb{N}$ such that $I^j := \{t : (t, j) \in \mathrm{dom}(\psi)\}$ has non-empty interior

$$\begin{aligned} \psi(t, j) &\in C, && \forall t \in \mathrm{int}(I^j), \\ \dot{\psi}(t, j) &= f(\psi(t, j)), && \text{for almost all } t \in I^j. \end{aligned}$$

(ii) for all $(t, j) \in \mathrm{dom}(\psi)$ such that $(t, j + 1) \in \mathrm{dom}(\psi)$

$$\begin{aligned} \psi(t, j) &\in D, \\ \psi(t, j + 1) &= g(\psi(t, j)). \end{aligned}$$

In (i) above, we say that ψ is *flowing* and in (ii) we say that ψ is *jumping*. We call ψ *persistently flowing* if it is eventually continuous or if there exists a uniform time constant τ_P whereby ψ flows for τ_P seconds infinitely often. Formally speaking, ψ is persistently flowing if

(PFi) $([t_J, \infty), J) \subset \text{dom}(\psi)$ for some $J \in \mathbb{N}$, or

(PFii) there exists $\tau_P > 0$ and an increasing sequence $\{j_k\}_{k=0}^{\infty} \subset \mathbb{N}$ such that $([t_{j_k}, t_{j_k} + \tau_P], j_k) \subset \text{dom}(\psi)$ for each $k \in \mathbb{N}$.

2.5 Optimization

Here we introduce some basic definitions and results regarding mathematical optimization. A detailed exposition on these topics can be found in [23]. In general, a mathematical optimization problem of the form

$$\min_{x \in X} f(x), \tag{2.6}$$

where $f : \mathbb{R}^p \to \mathbb{R} \cup \{-\infty, +\infty\}$ is called the objective function and $X \subseteq \mathbb{R}^p$ is called the feasibility set. The minimum value of f on X is called the optimal value of (2.6). A point $x_* \in X$ for which $f(x)$ attains its minimum value on X is called a solution of (2.6). By definition, a solution x_* of (2.6) satisfies $f(x_*) \leq f(x)$ for all $x \in X$. If $X \neq \emptyset$ then we say that (2.6) is feasible. Otherwise, it is infeasible and its optimal value is $+\infty$. A quadratic optimization problem, as we consider in this thesis, can be denoted by

$$\min \quad c^T x + \frac{1}{2} x^T \mathcal{E} x \tag{2.7a}$$

$$\text{s.t.} \quad Ax = b, \quad x \geq 0, \tag{2.7b}$$

where for $p_1, p_2 \in \mathbb{N}$, $c, x \in \mathbb{R}^{p_1}$, $0 \preceq \mathcal{E} = \mathcal{E}^T \in \mathbb{R}^{p_1 \times p_1}$, $b \in \mathbb{R}^{p_2}$, and $A \in \mathbb{R}^{p_2 \times p_1}$. We call (2.7) the *primal* problem and its associated *dual* is defined as

$$\max_z q(z),$$

where $q : \mathbb{R}^{p_2} \to \mathbb{R}$ is given by

$$q(z) := \min_x \left\{ -\frac{1}{2} x^T \mathcal{E} x - b^T z : c + \mathcal{E} x + A^T z \geq 0 \right\}.$$

The solutions to the primal and the dual can be related through the so-called Karush-Kuhn-Tucker (KKT) conditions which we define next.

Definition 2.5.1. (KKT conditions). *A point* $(x_*, z_*) \in \mathbb{R}^{p_1} \times \mathbb{R}^{p_2}$ *satisfies the KKT conditions for* (2.7) *if*

$$c + \mathcal{E}x_* + A^T z_* \geq 0, \quad Ax_* = b, \quad x_* \geq 0,$$
$$(c + \mathcal{E}x_* + A^T z_*)^T x_* = 0. \qquad \bullet$$

The equality $(c+\mathcal{E}x_* + A^T z_*)^T x_* = 0$ is called the *complementary slackness* condition. When the primal is feasible, we have the following equivalence.

Theorem 2.5.2. (Optimality conditions). *Suppose that* (2.7) *is feasible. Then, the optimal value of the primal is the optimal value of the dual. Moreover, a point* (x_*, z_*) *satisfies the KKT conditions for* (2.7) *if and only if* x_* *(resp.* z_**) is a solution to the primal (resp. the dual).*

The property that the optimal value of the primal and dual coincide, that is $c^T x_* = -b^T z_*$, is called *strong duality*. One remarkable consequence of Theorem 2.5.2 is that the points that satisfy the KKT conditions must define a convex set since, by equivalence, the primal-dual solution set is convex. This fact is not obvious since the complementary slackness condition is not affine in the variables x and z. This observation will allow us to use a simplified version of Danskin's Theorem, stated next, in the proof of a key result of Chapter 3.

Theorem 2.5.3. (Danskin's Theorem [15, Proposition B.25]). *Let* $Y \subset \mathbb{R}^{p_1}$ *be compact and convex. Given* $g : \mathbb{R}^{p_2} \times Y \to \mathbb{R}$*, suppose that* $x \mapsto g(x, y)$ *is differentiable for every* $y \in Y$*,* $\partial_x g$ *is continuous on* $\mathbb{R}^n \times Y$*, and* $y \mapsto g(x, y)$ *is strictly convex and continuous for every* $x \in \mathbb{R}^{p_2}$*. Define* $f : \mathbb{R}^{p_2} \to \mathbb{R}$ *by* $f(x) = \min_{y \in Y} g(x, y)$*. Then,* $\nabla f(x) = \partial_x g(x, y)|_{y=y_*(x)}$*, where* $y_*(x) = \operatorname{argmin}_{y \in Y} g(x, y)$*.*

A special instance of the problem (2.7) is a linear program, which corresponds to the case when $\mathcal{E} = 0$,

$$\min \quad c^T x \tag{2.8a}$$
$$\text{s.t.} \quad Ax = b, \quad x \geq 0. \tag{2.8b}$$

The above linear program is in standard form. Applying the definition of the dual yields,

$$\max \quad -b^T z \tag{2.9a}$$

$$\text{s.t.} \quad A^T z \geq c. \tag{2.9b}$$

Linear programming (that is, the design of algorithms to solve linear programs) is one of the most fundamental fields in mathematical optimization and thus demands particular attention.

2.6 Graph Theory

Graphs, as we use them in this thesis, are a concise way to model the interactions and information flow between certain subsystems in a network. We do not require any advanced knowledge of graph theory beyond the basic modeling framework. To this end, an undirected graph is a tuple $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$, where $\mathcal{V} = \{1, \ldots, n\}$ is a set of vertices, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of edges, and $W \in \mathbb{R}^{|\mathcal{E}|}$ is a vector of weights, indexed by edges in $\mathcal{G}$. For any given vertex $i \in \mathcal{V}$, the neighbor set of i is denoted $\mathcal{N}_i := \{j \in \mathcal{V} : (i,j) \in \mathcal{E}\}$. Given a matrix $A \in \mathbb{R}^{p \times n}$, we call a graph *connected with respect to* A if for each $\ell \in \{1, \ldots, p\}$ such that $a_{\ell,i} \neq 0 \neq a_{\ell,j}$, it holds that $(i,j) \in \mathcal{E}$.

Chapter 3

Robust distributed linear programming

In this chapter, we are interested in both the synthesis of distributed algorithms that can solve standard form linear programs and the characterization of their robustness properties. Our interest is motivated by network control scenarios that give rise to linear programs with an intrinsic distributed nature. Here, we consider scenarios where individual agents interact with their neighbors and are only responsible for computing their own component of the solution vector of the linear program.

The algorithm we design builds on the characterization of the solutions of the linear program as saddle points of a modified nonsmooth Lagrangian function. We show that the resulting continuous-time saddle-point algorithm is provably correct by relying on the set-valued LaSalle Invariance Principle and, in particular, a careful use of the properties of weakly and strongly invariant sets of the saddle-point dynamics. However, in general, this dynamics is not distributed because of a global parameter associated with the nonsmooth exact penalty function employed to encode the inequality constraints of the linear program. This motivates the design of a discontinuous saddle-point dynamics that, while enjoying the same convergence guarantees, is fully distributed and scalable with the dimension of the solution vector.

We then turn our attention to the characterization of the algorithm's robustness against disturbances and link failures. Specifically, we are able to show that it is integral-input-to-state stable but not input-to-state stable. Our proof method of the former is based on identifying a suitable iISS Lyapunov function, which we build by combining the Lyapunov function used in our LaSalle argument and results from converse Lyapunov theory. The fact that the dynamics is not input-to-state stable is a consequence of a more general result, that we also establish, which states that no algorithmic solution for linear programming is input-to-state stable when uncertainty in the problem data affects the dynamics as a disturbance. These robustness characterizations further allow us to establish the resilience of the proposed distributed dynamics to disturbances of finite variation and recurrently disconnected communication among the agents.

Simulations in an optimal control application conclude this chapter.

3.1 Problem statement

In this chapter, we consider feasible standard form linear programs (2.8), repeated here for convenience,

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b, \quad x \geq 0. \end{aligned}$$

We denote the set of solutions to (2.8) by $\mathcal{X} \subset \mathbb{R}^n$ and the set of solutions to its dual (2.9) by $\mathcal{Z} \subset \mathbb{R}^m$. Although the formulation of a linear program is concise and descriptive, it gives little insight into how one might go about finding solutions to the problem. Towards a more functional description, one can relate the solutions of a linear program to the saddle points of a modified Lagrangian function. The next result establishes this connection.

Proposition 3.1.1. (Solutions of linear program as saddle points). *For $K \geq 0$, let $L^K : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ be defined by*

$$L^K(x, z) = c^T x + \frac{1}{2}(Ax - b)^T(Ax - b) + z^T(Ax - b) + K\mathbb{1}_n^T \max\{0, -x\}. \quad (3.1)$$

Then, L^K is convex in x and concave (in fact, linear) in z. Moreover, *(i) if* $x_* \in \mathbb{R}^n$ *is a solution of* (2.8) *and* $z_* \in \mathbb{R}^m$ *is a solution of* (2.9), *then the point* (x_*, z_*) *is a saddle point of* L^K *for any* $K \geq \|A^T z_* + c\|_\infty$,

(ii) if $(\bar{x}, \bar{z}) \in \mathbb{R}^n \times \mathbb{R}^m$ *is a saddle point of* L^K *with* $K > \|A^T z_* + c\|_\infty$ *for some* $z_* \in \mathbb{R}^m$ *solution of* (2.9), *then* $\bar{x} \in \mathbb{R}^n$ *is a solution of* (2.8).

Proof. One can readily see from (3.1) that L^K is a convex-concave function. Let x_* be a solution of (2.8) and let z_* be a solution of (2.9). To show (i), using the characterization of $\mathcal{X} \times \mathcal{Z}$ described in Theorem 2.5.2 and the fact that $K \geq \|A^T z_* + c\|_\infty$, we can write for any $x \in \mathbb{R}^n$,

$$\begin{aligned}
L^K(x, z_*) &= c^T x + (Ax - b)^T (Ax - b) + z_*^T (Ax - b) + K \mathbb{1}_n^T \max\{0, -x\}, \\
&\geq c^T x + z_*^T (Ax - b) + (A^T z_* + c)^T \max\{0, -x\}, \\
&\geq c^T x + z_*^T (Ax - b) - (A^T z_* + c)^T x, \\
&= c^T x + z_*^T A(x - x_*) - (A^T z_* + c)^T (x - x_*), \\
&= c^T x - c^T (x - x_*) = c^T x_* = L^K(x_*, z_*).
\end{aligned}$$

The fact that $L^K(x_*, z) = L^K(x_*, z_*)$ for any $z \in \mathbb{R}^m$ is immediate. These two facts together imply that (x_*, z_*) is a saddle point of L^K.

We prove (ii) by contradiction. Let $(\bar{x}, \bar{z})$ be a saddle point of L^K with $K > \|A^T z_* + c\|_\infty$ for some $z_* \in \mathcal{Z}$, but suppose $\bar{x}$ is not a solution of (2.8). Let $x_* \in \mathcal{X}$. Since for fixed x, $z \mapsto L^K(x, z)$ is concave and differentiable, a necessary condition for $(\bar{x}, \bar{z})$ to be a saddle point of L^K is that $A\bar{x} - b = 0$. Using this fact, $L^K(x_*, \bar{z}) \geq L^K(\bar{x}, \bar{z})$ can be expressed as

$$c^T x_* \geq c^T \bar{x} + K \mathbb{1}_n^T \max\{0, -\bar{x}\}. \tag{3.2}$$

Now, if $\bar{x} \geq 0$, then $c^T x_* \geq c^T \bar{x}$, and thus $\bar{x}$ would be a solution of (2.8). If, instead, $\bar{x} \ngeq 0$,

$$\begin{aligned}
c^T \bar{x} &= c^T x_* + c^T (\bar{x} - x_*), \\
&= c^T x_* - z_*^T A(\bar{x} - x_*) + (A^T z_* + c)^T (\bar{x} - x_*), \\
&= c^T x_* - z_*^T (A\bar{x} - b) + (A^T z_* + c)^T \bar{x}, \\
&> c^T x_* - K \mathbb{1}_n^T \max\{0, -\bar{x}\},
\end{aligned}$$

which contradicts (3.2), concluding the proof. □

The relevance of Proposition 3.1.1 is two-fold. On the one hand, it justifies searching for the saddle points of L^K instead of directly solving the constrained optimization problem (2.8). On the other hand, given that L^K is convex-concave, a natural approach to find the saddle points is via the associated saddle-point dynamics. However, for an arbitrary function, such dynamics is known to render saddle points only stable, not asymptotically stable. Indeed, it is well known (see e.g., [38, 39]) that the saddle-point dynamics derived using the standard Lagrangian for a linear program does not converge to a solution of the linear program. Interestingly, using penalty functions associated with the constraints to augment the cost function has been observed to improve the convergence of the saddle-point dynamics [8]. In our case, we augment the linear cost function $c^T x$ with a quadratic penalty for the equality constraints and a nonsmooth penalty function for the inequality constraints. This results in the nonlinear optimization problem,

$$\min_{Ax=b} c^T x + ||Ax - b||^2 + K \mathbb{1}_n^T \max\{0, -x\},$$

whose standard Lagrangian is equivalent to L^K. The nonsmooth penalty function is required (among other reasons that we expand on elsewhere) to ensure that there is an exact equivalence between saddle points of L^K and the solutions of (2.8). On the other hand, the use of any smooth penalty function, such as the logarithmic barrier function used in [106], will not enjoy this property. It is worth noticing that the lower bounds on K in Proposition 3.1.1 are characterized by certain dual solutions, which are unknown a priori. Nevertheless, our discussion later shows that this problem can be circumvented and that knowledge of such bounds is not necessary for the design of robust, distributed algorithms that solve linear programs.

In the next section, we show that indeed the saddle-point dynamics of L^K asymptotically converges to saddle points.

3.2 Saddle-point dynamics for distributed linear programming

In this section, we design a continuous-time algorithm to find the solutions of (2.8) and discuss its distributed implementation in a multi-agent system. Building on the result in Proposition 3.1.1, we consider the saddle-point dynamics (gradient descent in one argument, gradient ascent in the other) of the modified Lagrangian L^K. The nonsmooth character of L^K means that its saddle-point dynamics takes the form of the following differential inclusion,

$$\dot{x} + c + A^T(z + Ax - b) \in -K\partial \max\{0, -x\}, \tag{3.3a}$$

$$\dot{z} = Ax - b. \tag{3.3b}$$

For notational convenience, we use $F^K_{\text{sdl}} : \mathbb{R}^n \times \mathbb{R}^m \rightrightarrows \mathbb{R}^n \times \mathbb{R}^m$ to refer to (3.3). The following result characterizes the asymptotic convergence of (3.3) to the set of solutions to (2.8)-(2.9).

Theorem 3.2.1. (Asymptotic convergence to the primal-dual solution set). *Let $(x_*, z_*) \in \mathcal{X} \times \mathcal{Z}$ and define $V : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ as*

$$V(x, z) = \frac{1}{2}(x - x_*)^T(x - x_*) + \frac{1}{2}(z - z_*)^T(z - z_*).$$

For $\infty > K \geq \|A^T z_* + c\|_\infty$, it holds that $\mathcal{L}_{F^K_{sdl}} V(x, z) \subset (-\infty, 0]$ for all $(x, z) \in \mathbb{R}^n \times \mathbb{R}^m$ and any trajectory $t \mapsto (x(t), z(t))$ of (3.3) converges asymptotically to the set $\mathcal{X} \times \mathcal{Z}$.

Proof. Our proof strategy is based on verifying the hypotheses of the LaSalle Invariance Principle, cf. Theorem 2.3.1, and identifying the set of primal-dual solutions as the corresponding largest weakly invariant set. First, note that Lemma 2.2.1 implies that F^K_{sdl} is locally bounded, upper semi-continuous and takes nonempty, convex, and compact values. By Proposition 3.1.1(i), (x_*, z_*) is a saddle point of L^K when $K \geq \|A^T z_* + c\|_\infty$. Consider the quadratic function V defined in the theorem statement, which is continuously differentiable and radially unbounded. Let $a \in \mathcal{L}_{F^K_{\text{sdl}}} V(x, z)$. By definition, there exists

$v = (-c - A^T(z + Ax - b) - g_x, Ax - b) \in F^K_{\text{sdl}}(x,z)$, with $g_x \in K\partial \max\{0, -x\}$, such that

$$a = v^T \nabla V(x,z) = (x - x_*)^T(-c - A^T(z + Ax - b) - g_x) + (z - z_*)(Ax - b). \quad (3.4)$$

Since L^K is convex in its first argument, and $c + A^T(z + Ax - b) + g_x \in \partial_x L^K(x,z)$, using the first-order condition of convexity, we have

$$L^K(x,z) \leq L^K(x_*,z) + (x - x_*)^T\big(c + A^T(z + Ax - b) + g_x\big).$$

Since L^K is linear in z, we have $L^K(x,z) = L^K(x,z_*) + (z - z_*)^T(Ax - b)$. Using these facts in (3.4), we get

$$a \leq L^K(x_*,z) - L^K(x,z_*) = L^K(x_*,z) - L^K(x_*,z_*) + L^K(x_*,z_*) - L^K(x,z_*) \leq 0,$$

since (x_*, z_*) is a saddle point of L^K. Since a is arbitrary, we deduce that $\mathcal{L}_{F^K_{\text{sdl}}} V(x,z) \subset (-\infty, 0]$. For any given $\rho \geq 0$, this implies that the set $V^{-1}(\leq \rho)$ is strongly invariant with respect to (3.3). Since V is radially unbounded, $V^{-1}(\leq \rho)$ is also compact. The conditions of Theorem 2.3.1 are then satisfied with $X = V^{-1}(\leq \rho)$, and therefore any trajectory of (3.3) starting in $V^{-1}(\leq \rho)$ converges to the largest weakly invariant set $\mathcal{M}$ in $\{(x,z) \in V^{-1}(\leq \rho) : 0 \in \mathcal{L}_{F^K_{\text{sdl}}} V(x,z)\}$ (note that for any initial condition (x_0, z_0) one can choose a ρ such that $(x_0, z_0) \in V^{-1}(\leq \rho)$). This set is closed, which can be justified as follows. Since F^K_{sdl} is upper semi-continuous and V is continuously differentiable, the map $(x,z) \mapsto \mathcal{L}_{F^K_{\text{sdl}}} V(x,z)$ is also upper semi-continuous. Closedness then follows from [9, Convergence Theorem]. We now show that $\mathcal{M} \subseteq \mathcal{X} \times \mathcal{Z}$. To start, take $(x', z') \in \mathcal{M}$. Then $L^K(x_*, z_*) - L^K(x', z_*) = 0$, which implies

$$\tilde{L}^K(x', z_*) - (Ax' - b)^T(Ax' - b) = 0, \quad (3.5)$$

where $\tilde{L}^K(x', z_*) = c^T x_* - c^T x' - z_*^T(Ax' - b) - K\mathbb{1}_n^T \max\{0, -x'\}$. Using strong duality, the expression of $\tilde{L}^K$ can be simplified to $\tilde{L}^K(x', z_*) = -(A^T z_* + c)^T x' - K\mathbb{1}_n^T \max\{0, -x'\}$. In addition, $A^T z_* + c \geq 0$ by dual feasibility. Thus, when $K \geq \|A^T z_* + c\|_\infty$, we have $\tilde{L}^K(x, z_*) \leq 0$ for all $(x,z) \in V^{-1}(\leq \rho)$. This implies that $(Ax' - b)^T(Ax' - b) = 0$ for (3.5) to be true, which further implies that

$Ax' - b = 0$. Moreover, from the definition of $\tilde{L}^K$ and the bound on K, one can see that if $x' \ngeq 0$, then $\tilde{L}^K(x', z_*) < 0$. Therefore, for (3.5) to be true, it must be that $x' \geq 0$. Finally, from (3.5), we get that $\tilde{L}^K(x', z_*) = c^T x_* - c^T x' = 0$. In summary, if $(x', z') \in \mathcal{M}$ then $c^T x_* = c^T x'$, $Ax' - b = 0$, and $x' \geq 0$. Therefore, x' is a solution of (2.8).

Now, we show that z' is a solution of (2.9). Because $\mathcal{M}$ is weakly invariant, there exists a trajectory starting from (x', z') that remains in $\mathcal{M}$. The fact that $Ax' = b$ implies that $\dot{z} = 0$, and hence $z(t) = z'$ is constant. For any given $i \in \{1, \ldots, n\}$, we consider the cases (i) $x'_i > 0$ and (ii) $x'_i = 0$. In case (i), the dynamics of the ith component of x is $\dot{x}_i = -(c + A^T z')_i$ where $(c + A^T z')_i$ is constant. It cannot be that $-(c + A^T z')_i > 0$ because this would contradict the fact that $t \mapsto x_i(t)$ is bounded. Therefore, $(c + A^T z')_i \geq 0$. If $\dot{x}_i = -(c + A^T z')_i < 0$, then $x_i(t)$ will eventually become zero, which we consider in case (ii). In fact, since the solution remains in $\mathcal{M}$, without loss of generality, we can assume that (x', z') is such that either $x'_i > 0$ and $(c + A^T z')_i = 0$ or $x'_i = 0$ for each $i \in \{1, \ldots, n\}$. Consider now case (ii). Since $x_i(t)$ must remain non-negative in $\mathcal{M}$, it must be that $\dot{x}_i(t) \geq 0$ when $x_i(t) = 0$. That is, in $\mathcal{M}$, we have $\dot{x}_i(t) \geq 0$ when $x_i(t) = 0$ and $\dot{x}_i(t) \leq 0$ when $x_i(t) > 0$. Therefore, for any trajectory $t \mapsto x_i(t)$ in $\mathcal{M}$ starting at $x'_i = 0$, the unique Filippov solution is that $x_i(t) = 0$ for all $t \geq 0$. As a consequence, $(c + A^T z')_i \in [0, K]$ if $x'_i = 0$. To summarize cases (i) and (ii), we have

- $Ax' = b$ and $x' \geq 0$ (primal feasibility),
- $A^T z' + c \geq 0$ (dual feasibility),
- $(A^T z' + c)_i = 0$ if $x'_i > 0$ and $x'_i = 0$ if $(A^T z' + c)_i > 0$ (complementary slackness),

which is sufficient to show that $z \in \mathcal{Z}$ (cf. Theorem 2.5.2). Hence $\mathcal{M} \subseteq \mathcal{X} \times \mathcal{Z}$. Since the trajectories of (3.3) converge to $\mathcal{M}$, this completes the proof. □

Using a slightly more complicated lower bound on the parameter K, we are able to show point-wise convergence of the saddle-point dynamics. We state this result next.

Corollary 3.2.2. (Point-wise convergence of saddle-point dynamics). *Let $\rho > 0$. Then, with the notation of Theorem 3.2.1, for*

$$\infty > K \geq \max_{(x,z)\in(\mathcal{X}\times\mathcal{Z})\cap V^{-1}(\leq\rho)} \|A^T z + c\|_\infty, \tag{3.6}$$

it holds that any trajectory $t \mapsto (x(t), z(t))$ of (3.3) starting in $V^{-1}(\leq \rho)$ converges asymptotically to a point in $\mathcal{X} \times \mathcal{Z}$.

Proof. If K satisfies (3.6), then in particular $K \geq \|A^T z_* + c\|_\infty$. Thus, $V^{-1}(\leq \rho)$ is strongly invariant under (3.3) since $\mathcal{L}_{F^K_{\text{sdl}}} V(x,z) \subset (-\infty, 0]$ for all $(x,z) \in V^{-1}(\leq \rho)$ (cf. Theorem 3.2.1). Also, $V^{-1}(\leq \rho)$ is bounded because V is quadratic. Therefore, by the Bolzano-Weierstrass theorem [89, Theorem 3.6], there exists a subsequence $(x(t_k), z(t_k)) \in V^{-1}(\leq\rho)$ that converges to a point $(\tilde{x}, \tilde{z}) \in (\mathcal{X} \times \mathcal{Z}) \cap V^{-1}(\leq\rho)$. Given $\varepsilon > 0$, let k^* be such that $\|(x(t_{k^*}), z(t_{k^*})) - (\tilde{x}, \tilde{z})\| \leq \varepsilon$. Consider the function $\tilde{V}(x,z) = \frac{1}{2}(x-\tilde{x})^T(x-\tilde{x}) + \frac{1}{2}(z-\tilde{z})^T(z-\tilde{z})$. When K satisfies (3.6), again it holds that $K \geq \|A^T\tilde{z}+c\|_\infty$. Applying Theorem 3.2.1 once again, $\tilde{V}^{-1}(\leq \rho)$ is strongly invariant under (3.3). Consequently, for $t \geq t_{k^*}$, we have $(x(t), z(t)) \in \tilde{V}^{-1}(\leq \tilde{V}(x(t_{k^*}), z(t_{k^*}))) = \overline{\mathbb{B}}\big((\tilde{x},\tilde{z}), \|(x(t_{k^*}), z(t_{k^*}))-(\tilde{x},\tilde{z})\|\big) \subset \overline{\mathbb{B}}((\tilde{x},\tilde{z}),\varepsilon)$. Since ε can be taken arbitrarily small, this implies that $(x(t), z(t))$ converges to the point $(\tilde{x}, \tilde{z}) \in \mathcal{X} \times \mathcal{Z}$. □

Remark 3.2.3. (Choice of parameter K). The bound (3.6) for the parameter K depends on (i) the primal-dual solution set $\mathcal{X} \times \mathcal{Z}$ as well as (ii) the initial condition, since the result is only valid when the dynamics start in $V^{-1}(\leq \rho)$. However, if the set $\mathcal{X}\times\mathcal{Z}$ is compact, the parameter K can be chosen independently of the initial condition since the maximization in (3.6) would be well defined when taken over the whole set $\mathcal{X} \times \mathcal{Z}$. We should point out that, in Section 3.2.1 we introduce a discontinuous version of the saddle-point dynamics which does not involve K. •

3.2.1 Discontinuous saddle-point dynamics

Here, we propose an alternative dynamics to (3.3) that does not rely on knowledge of the parameter K and also converges to the solutions of (2.8)-(2.9).

We begin by defining the *nominal flow function* $f : \mathbb{R}^n_{\geq 0} \times \mathbb{R}^m \to \mathbb{R}^n$ by

$$f(x,z) := -c - A^T(z + Ax - b). \tag{3.7}$$

This definition is motivated by the fact that, for $(x,z) \in \mathbb{R}^n_{>0} \times \mathbb{R}^m$, the set $\partial_x L^K(x,z)$ is the singleton $\{-f(x,z)\}$. The *discontinuous saddle-point dynamics* is, for $i \in \{1,\dots,n\}$,

$$\dot{x}_i = \begin{cases} f_i(x,z), & \text{if } x_i > 0, \\ \max\{0, f_i(x,z)\}, & \text{if } x_i = 0, \end{cases} \tag{3.8a}$$

$$\dot{z} = Ax - b. \tag{3.8b}$$

When convenient, we use the notation $f_{\text{dis}} : \mathbb{R}^n_{\geq 0} \times \mathbb{R}^m \to \mathbb{R}^n \times \mathbb{R}^m$ to refer to the discontinuous dynamics (3.8). Note that the discontinuous function that defines the dynamics (3.8a) is simply the positive projection operator. That is, when $x_i = 0$, its flow is given by the projection of $f_i(x,z)$ onto $\mathbb{R}_{\geq 0}$. We understand the solutions of (3.8) in the Filippov sense. We begin our analysis by establishing a relationship between the Filippov set-valued map of f_{dis} and the saddle-point dynamics F^K_{sdl} which allows us to relate the trajectories of (3.8) and (3.3).

Proposition 3.2.4. (Trajectories of the discontinuous saddle-point dynamics are trajectories of the saddle-point dynamics). *Let $\rho > 0$ and $(x_*, z_*) \in \mathcal{X} \times \mathcal{Z}$ be given and the function V be defined as in Theorem 3.2.1. Then, for any*

$$\infty > K \geq K_1 := \max_{(x,z) \in V^{-1}(\leq \rho)} \|f(x,z)\|_\infty,$$

the inclusion $\mathcal{F}[f_{dis}](x,z) \subseteq F^K_{sdl}(x,z)$ holds for every $(x,z) \in V^{-1}(\leq \rho)$. Thus, the trajectories of (3.8) starting in $V^{-1}(\leq \rho)$ are also trajectories of (3.3).

Proof. The projection onto the i^{th} component of the Filippov set-valued map $\mathcal{F}[f_{\text{dis}}]$ is

$$\text{proj}_i(\mathcal{F}[f_{\text{dis}}](x,z)) = \begin{cases} \{f_i(x,z)\}, & \text{if } i \in \{1,\dots,n\} \text{ and } x_i > 0, \\ [f_i(x,z), \max\{0, f_i(x,z)\}], & \text{if } i \in \{1,\dots,n\} \text{ and } x_i = 0, \\ \{(Ax-b)_i\}, & \text{if } i \in \{n+1,\dots,n+m\}. \end{cases}$$

As a consequence, for any $i \in \{n+1, \dots, n+m\}$, we have

$$\text{proj}_i(F_{\text{sdl}}^K(x,z)) = (Ax-b)_i = \text{proj}_i(\mathcal{F}[f_{\text{dis}}](x,z)),$$

and, for any $i \in \{1, \dots, n\}$ such that $x_i > 0$, we have

$$\text{proj}_i(F_{\text{sdl}}^K(x,z)) = (-c - A^T(Ax-b+z))_i = \{f_i(x,z)\} = \text{proj}_i(\mathcal{F}[f_{\text{dis}}](x,z)).$$

Thus, let us consider the case when $x_i = 0$ for some $i \in \{1, \dots, n\}$. In this case, note that

$$\begin{aligned} \text{proj}_i(\mathcal{F}[f_{\text{dis}}](x,z)) &= [f_i(x,z), \max\{0, f_i(x,z)\}] \subseteq [f_i(x,z), f_i(x,z) + |f_i(x,z)|], \\ \text{proj}_i(F_{\text{sdl}}^K(x,z)) &= [f_i(x,z), f_i(x,z) + K]. \end{aligned}$$

The choice $K \geq |f_i(x,z)|$ for each $i \in \{1, \dots, n\}$ makes $\mathcal{F}[f_{\text{dis}}](x,z) \subseteq F_{\text{sdl}}^K(x,z)$. More generally, since $V^{-1}(\rho)$ is compact and f is continuous, the choice

$$\infty > K \geq \max_{(x,z) \in V^{-1}(\rho)} \|f(x,z)\|_\infty,$$

guarantees $\mathcal{F}[f_{\text{dis}}](x,z) \subseteq F_{\text{sdl}}^K(x,z)$ for all $(x,z) \in V^{-1}(\rho)$. By Theorem 3.2.1, we know that V is non-increasing along (3.3), implying that $V^{-1}(\leq \rho)$ is strongly invariant with respect to (3.3), and hence (3.8) too. Therefore, any trajectory of (3.8) starting in $V^{-1}(\leq \rho)$ is a trajectory of (3.3). □

To shed some light on the intuitive relationship between the trajectories of (3.3) and (3.8), Figure 3.1 illustrates the effect that increasing K has on the trajectories of (3.3). From a given initial condition, at some point the value of K is large enough, cf. Proposition 3.2.4, to make the trajectory of (3.8) (which never leave $\mathbb{R}^n_{\geq 0} \times \mathbb{R}^m$) also a trajectory of (3.3). Building on Proposition 3.2.4, the next result characterizes the asymptotic convergence of (3.8).

Corollary 3.2.5. (Asymptotic convergence of the discontinuous saddle-point dynamics). *The trajectories of* (3.8) *starting in* $\mathbb{R}^n_{\geq 0} \times \mathbb{R}^m$ *converge asymptotically to a point in* $\mathcal{X} \times \mathcal{Z}$.

Proof. Let V be defined as in Theorem 3.2.1. Given any initial condition $(x_0, z_0) \in \mathbb{R}^n \times \mathbb{R}^m$, let $t \mapsto (x(t), z(t))$ be a trajectory of (3.8) starting from (x_0, z_0) and

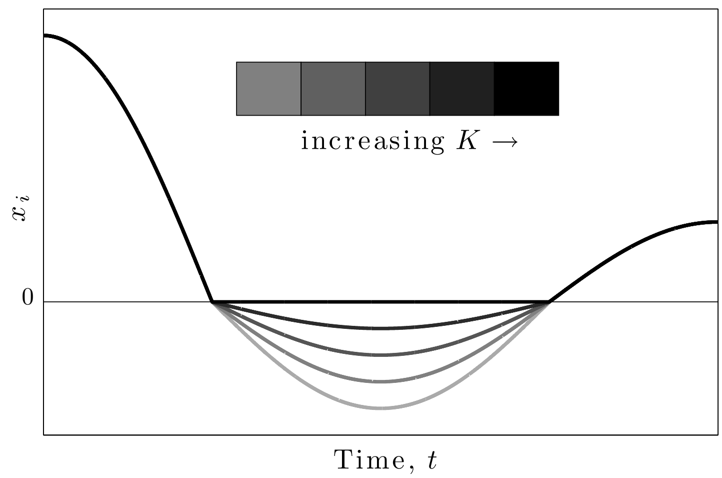


Figure 3.1: [Illustration of the effect that increasing K has on the trajectories of (3.3). For a given initial condition, the trajectories of (3.3) have increasingly smaller "incursions" into the region where $x_i < 0$ as K increases, until a finite value is reached where the corresponding trajectory of (3.8) is also a trajectory of (3.3).

let $\rho = V(x_0, z_0)$. Note that $t \mapsto (x(t), z(t))$ does not depend on K because (3.8) does not depend on K. Proposition 3.2.4 establishes that $t \mapsto (x(t), z(t))$ is also a trajectory of (3.3) for $K \geq K_1$. Imposing the additional condition that

$$\infty > K \geq \max\left\{K_1, \max_{(x_*, z_*) \in (\mathcal{X} \times \mathcal{Z}) \cap V^{-1}(\leq \rho)} \|A^T z_* + c\|_\infty\right\},$$

Corollary 3.2.2 implies that the trajectories of (3.3) (and thus $t \mapsto (x(t), z(t))$ converge asymptotically to a point in $\mathcal{X} \times \mathcal{Z}$. □

Establishing the above convergence of (3.8) directly (i.e., without a comparison to the trajectories of (3.3)) would require that certain regularity conditions hold for the switching behavior of the system. On the other hand, our proof of convergence for the set-valued saddle-point dynamics (3.3) made use of powerful stability tools for set-valued dynamics which did not rely on any such regularity. Therefore, arriving at Corollary 3.2.5 by means of comparing the trajectories of (3.8) and (3.3) proved to be an elegant way of accounting for the complexity of the dynamics (3.8). Moreover, the interpretation of the trajectories of (3.8) in the Filippov sense is instrumental for our analysis in Section 3.3 where we study the robustness against disturbances using powerful Lyapunov-like tools for differential inclusions.

Remark 3.2.6. (Comparison to existing dynamics for linear programming). Though a central motivation for the development of our linear programming algorithm is the establishment of various robustness properties which we study next, the dynamics (3.8) and associated convergence results of this section are both novel and have distinct contributions. The work [39] also builds on the saddle-point dynamics of a smooth Lagrangian function to introduce an algorithm for linear programs in inequality form. Instead of exact penalty functions, this approach uses projections to keep the evolution within the feasible set, resulting in a discontinuous dynamics in both the primal and dual variables. Convergence in the dual variables is to some unknown point which is not shown to be a solution to the dual problem. This is to be contrasted with the convergence properties of the dynamics (3.8) stated in Corollary 3.2.5. Also, we do not require Slater's condition to be satisfied. •

3.2.2 Distributed implementation

An important advantage of the dynamics (3.8) over other linear programming methods is that it is well-suited for distributed implementation. To make this statement precise, consider a scenario where each component of $x \in \mathbb{R}^n$ corresponds to the state of an independent decision maker or agent and the interconnection between the agents is modeled by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. To see under what conditions the dynamics (3.8) can be implemented by this multi-agent system, let us express it component-wise. First, the nominal flow function in (3.8a) for agent $i \in \{1, \dots, n\}$ is,

$$f_i(x,z) = -c_i - \sum_{\ell=1}^{m} a_{\ell,i}\Big[z_\ell + \sum_{k=1}^{n} a_{\ell,k}x_k - b_\ell\Big] = -c_i - \sum_{\{\ell\,:\,a_{\ell,i}\neq 0\}} a_{\ell,i}\Big[z_\ell + \sum_{\{k\,:\,a_{\ell,k}\neq 0\}} a_{\ell,k}x_k - b_\ell\Big],$$

and the dynamics (3.8b) for each $\ell \in \{1, \dots, m\}$ is

$$\dot{z}_\ell = \sum_{\{i\,:\,a_{\ell,i}\neq 0\}} a_{\ell,i}x_i - b_\ell. \tag{3.9}$$

According to these expressions, in order for agent $i \in \{1, \ldots, n\}$ to be able to implement its corresponding dynamics in (3.8a), it also needs access to certain components of z (specifically, those components z_ℓ for which $a_{\ell,i} \neq 0$), and therefore needs to implement their corresponding dynamics (3.9). We say that the dynamics (3.8) is *distributed over* $\mathcal{G}$ when the following holds

(D1) for each $i \in \mathcal{V}$, agent i knows

  (a) $c_i \in \mathbb{R}$,

  (b) every $b_\ell \in \mathbb{R}$ for which $a_{\ell,i} \neq 0$,

  (c) the non-zero elements of every row of A for which the i^{th} component, $a_{\ell,i}$, is non-zero,

(D2) agent $i \in \mathcal{V}$ has control over the variable $x_i \in \mathbb{R}$,

(D3) $\mathcal{G}$ is connected with respect to A, and

(D4) agents have access to the variables controlled by neighboring agents.

Note that (D3) guarantees that the agents that implement (3.9) for a particular $\ell \in \{1, \ldots, m\}$ are neighbors in $\mathcal{G}$. At times, it will be useful to view the dynamics of component of z as being implemented by virtual agents.

Remark 3.2.7. (Scalability of the nominal saddle-point dynamics). A different approach to solve (2.8) is the following: reformulate the optimization problem as the constrained minimization of a sum of convex functions all of the form $\frac{1}{n}c^T x$ and use the algorithms developed in, for instance, [25, 68, 72, 106, 116], for distributed convex optimization. However, this approach would lead to agents storing and communicating with neighbors estimates of the entire solution vector in $\mathbb{R}^n$, and hence would not scale well with the number of agents of the network. In contrast, to execute the discontinuous saddle-point dynamics, agents only need to store the component of the solution vector that they control and communicate it with neighbors. Therefore, the dynamics scales well with respect to the number of agents in the network. •

3.3 Robustness against disturbances

Here we explore the robustness properties of the discontinuous saddle-point dynamics (3.8) against disturbances. Such disturbances may correspond to noise, unmodeled dynamics, or incorrect agent knowledge of the data defining the linear program. Note that the global asymptotic stability of $\mathcal{X} \times \mathcal{Z}$ under (3.8) characterized in Section 3.2 naturally provides a robustness guarantee on this dynamics: when $\mathcal{X} \times \mathcal{Z}$ is compact, sufficiently small perturbations do not destroy the global asymptotic stability of the equilibria, cf. [26]. Our objective here is to go beyond this qualitative statement to obtain a more precise, quantitative description of robustness. To this end, we consider the notions of input-to-state stability (ISS) and integral-input-to-state stability (iISS). In Section 3.3.1 we show that, when the disturbances correspond to uncertainty in the problem data, no dynamics for linear programming can be ISS. This motivates us to explore the weaker notion of iISS. In Section 3.3.2 we show that (3.8) with additive disturbances is iISS.

Remark 3.3.1. (Robust dynamics versus robust optimization). We make a note of the distinction between the notion of algorithm robustness, which is what we study here, and the term robust (or worst-case) optimization, see e.g., [19]. The latter refers to a type of problem formulation in which some notion of variability (which models uncertainty) is explicitly included in the problem statement. Mathematically,

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & f(x, \omega) \leq 0, \ \forall \omega \in \Omega, \end{aligned}$$

where ω is an uncertain parameter. Building on the observation that one only has to consider the worst-case values of ω, one can equivalently cast the optimization problem with constraints that only depend on x, albeit at the cost of a loss of structure in the formulation. •

Without explicitly stating it from here on, we make the following assumption throughout the section:

(A) The solution sets to (2.8) and (2.9) are compact (i.e., $\mathcal{X} \times \mathcal{Z}$ is compact).

The justification for this assumption is twofold. On the technical side, our study of the iISS properties of (3.10) in Section 3.3.2 builds on a Converse Lyapunov Theorem [26] which requires the equilibrium set to be compact (the question of whether the Converse Lyapunov Theorem holds when the equilibrium set is not compact and the dynamics is discontinuous is an open problem). On the practical side, one can add box-type constraints to (2.8), ensuring that (A) holds.

We now formalize the disturbance model considered in this section. Let $w = (w_x, w_z) : \mathbb{R}_{\geq 0} \to \mathbb{R}^n \times \mathbb{R}^m$ be locally essentially bounded and enter the dynamics as follows,

$$\dot{x}_i = \begin{cases} f_i(x,z) + (w_x)_i, & \text{if } x_i > 0, \\ \max\{0, f_i(x,z) + (w_x)_i\}, & \text{if } x_i = 0, \end{cases} \quad \forall i \in \{1, \dots, n\}, \tag{3.10a}$$

$$\dot{z} = Ax - b + w_z. \tag{3.10b}$$

For notational purposes, we use $f_{\text{dis}}^{w} : \mathbb{R}^{2(n+m)} \to \mathbb{R}^{n+m}$ to denote (3.10). As mentioned above, the additive disturbance w captures unmodeled dynamics, and both measurement and computation noise. In addition, any error in an agent's knowledge of the problem data (A, b and c) can be interpreted as a specific manifestation of w. For example, if agent $i \in \{1, \dots, n\}$ uses an estimate $\hat{c}_i$ of c_i when computing its dynamics, this can be modeled in (3.10) by considering $(w_x(t))_i = c_i - \hat{c}_i$. To make precise the correspondence between the disturbance w and uncertainties in the problem data, we provide the following convergence result when the disturbance is constant.

Corollary 3.3.2. (Convergence under constant disturbances). *For constant* $\overline{w} = (\overline{w}_x, \overline{w}_z) \in \mathbb{R}^n \times \mathbb{R}^m$, *consider the* perturbed linear program,

$$\min \quad (c - \overline{w}_x - A^T \overline{w}_z)^T x \tag{3.11a}$$

$$\text{s.t.} \quad Ax = b - \overline{w}_z, \quad x \geq 0, \tag{3.11b}$$

and, with a slight abuse in notation, let $\mathcal{X}(\overline{w}) \times \mathcal{Z}(\overline{w})$ *be its primal-dual solution set. Suppose that* $\mathcal{X}(\overline{w}) \times \mathcal{Z}(\overline{w})$ *is nonempty. Then each trajectory of* (3.10) *starting in* $\mathbb{R}^n_{\geq 0} \times \mathbb{R}^m$ *with constant disturbance* $w(t) = \overline{w} = (\overline{w}_x, \overline{w}_z)$ *converges asymptotically to a point in* $\mathcal{X}(\overline{w}) \times \mathcal{Z}(\overline{w})$.

Proof. Note that (3.10) with disturbance $\overline{w}$ corresponds to the undisturbed dynamics (3.8) for the perturbed problem (3.11). Since $\mathcal{X}(\overline{w}) \times \mathcal{Z}(\overline{w}) \neq \emptyset$, Corollary 3.2.5 implies the result. □

3.3.1 No dynamics for linear programming is input-to-state stable

The notion of input-to-state stability (ISS) is a natural starting point to study the robustness of dynamical systems against disturbances. Informally, if a dynamics is ISS, then bounded disturbances give rise to bounded deviations from the equilibrium set. Here we show that any dynamics that (i) solve any feasible linear program and (ii) where uncertainties in the problem data (A, b, and c) enter as disturbances is not input-to-state stable (ISS). Our analysis relies on the properties of the solution set of a linear program. To make our discussion precise, we begin by recalling the definition of input-to-state stability.

Definition 3.3.3. (Input-to-state stability [97]). *The dynamics* (3.10) *is ISS with respect to* $\mathcal{X} \times \mathcal{Z}$ *if there exist* $\beta \in \mathcal{KL}$ *and* $\gamma \in \mathcal{K}$ *such that, for any trajectory* $t \mapsto (x(t), z(t))$ *of* (3.10)*, one has*

$$\|(x(t), z(t))\|_{\mathcal{X}\times\mathcal{Z}} \leq \beta(\|(x(0), z(0)\|_{\mathcal{X}\times\mathcal{Z}}, t) + \gamma(\|w\|_\infty),$$

for all $t \geq 0$*. Here,* $\|w\|_\infty := \operatorname{esssup}_{s\geq 0} \|w(s)\|$ *is the essential supremum of* $w(t)$*.*

Our method to show that no dynamics is ISS is constructive. We find a constant disturbance such that the primal-dual solution set to some perturbed linear program is unbounded. Since any point in this unbounded solution set is a stable equilibrium by assumption, this precludes the possibility of the dynamics from being ISS. This argument is made precise next.

Theorem 3.3.4. (No dynamics for linear programming is ISS). *Consider the generic dynamics*

$$(\dot{x}, \dot{z}) = \Phi(x, z, v), \tag{3.12}$$

with disturbance $t \mapsto v(t)$. Assume uncertainties in the problem data are modeled by v. That is, there exists a surjective function $g = (g_1, g_2) : \mathbb{R}^{n+m} \to \mathbb{R}^n \times \mathbb{R}^m$ with $g(0) = (0, 0)$ such that, for $\bar{v} \in \mathbb{R}^{n+m}$, the primal-dual solution set $\mathcal{X}(\bar{v}) \times \mathcal{Z}(\bar{v})$ of the linear program

$$\min \quad (c + g_1(\bar{v}))^T x \tag{3.13a}$$

$$\text{s.t.} \quad Ax = b + g_2(\bar{v}), \quad x \geq 0, \tag{3.13b}$$

is the stable equilibrium set of $(\dot{x}, \dot{z}) = \Phi(x, z, \bar{v})$ whenever $\mathcal{X}(\bar{v}) \times \mathcal{Z}(\bar{v}) \neq \emptyset$. Then, the dynamics (3.12) *is not ISS with respect to $\mathcal{X} \times \mathcal{Z}$.*

The proof of the above result requires the following technical Lemma.

Lemma 3.3.5. (Property of feasible set). *If $\{Ax = b, x \geq 0\}$ is non-empty and bounded then $\{A^T z \geq 0\}$ is unbounded.*

Proof. We start by proving that there exists an $\nu \in \mathbb{R}^m$ such that $\{Ax = b+\nu, x \geq 0\}$ is empty. Define the vector $s \in \mathbb{R}^n$ component-wise as $s_i = \max_{\{Ax=b,x\geq 0\}} x_i$. Since $\{Ax = b, x \geq 0\}$ is compact and non-empty, s is finite. Next, fix $\varepsilon > 0$ and let $\nu = -A(s+\varepsilon\mathbb{1}_n)$. Note that $Ax = b+\nu$ corresponds to $A(x+s+\varepsilon\mathbb{1}_n) = b$, which is a shift by $s+\varepsilon\mathbb{1}_n$ in each component of x. By construction, $\{Ax = b+\nu, x \geq 0\}$ is empty. Then, the application of Farkas' Lemma [23, pp. 263] yields that there exists $\hat{z} \in \mathbb{R}^m$ such that $A^T\hat{z} \geq 0$ and $(b+\nu)^T\hat{z} < 0$ (in particular, $(b+\nu)^T\hat{z} < 0$ implies that $\hat{z} \neq 0$). For any $\lambda \in \mathbb{R}_{\geq 0}$, it holds that $A^T(\lambda\hat{z}) \geq 0$, and thus $\lambda\hat{z} \in \{A^T z \geq 0\}$, which implies the result. □

We are now ready to prove Theorem 3.3.4.

Proof of Theorem 3.3.4. We divide the proof in two cases depending on whether $\{Ax = b, x \geq 0\}$ is (i) unbounded or (ii) bounded. In both cases, we design a constant disturbance $v(t) = \bar{v}$ such that the equilibria of (3.12) contains points arbitrarily far away from $\mathcal{X} \times \mathcal{Z}$. This would imply that the dynamics is not ISS. Consider case (i). Since $\{Ax = b, x \geq 0\}$ is unbounded, convex, and polyhedral, there exists a point $\hat{x} \in \mathbb{R}^n$ and direction $\nu_x \in \mathbb{R}^n \setminus \{0\}$ such that $\hat{x} + \lambda\nu_x \in \text{bd}(\{Ax = b, x \geq 0\})$ for all $\lambda \geq 0$. Here $\text{bd}(\cdot)$ refers to the boundary of the set.

Let $\eta \in \mathbb{R}^n$ be such that $\eta^T \nu_x = 0$ and $\hat{x} + \varepsilon\eta \notin \{Ax = b, x \geq 0\}$ for any $\varepsilon > 0$ (geometrically, η is normal to and points out of $\{Ax = b, x \geq 0\}$ at $\hat{x}$). Now that these quantities have been defined, consider the following linear program,

$$\begin{aligned} \min \quad & \eta^T x \\ \text{s.t.} \quad & Ax = b, \quad x \geq 0. \end{aligned} \tag{3.14}$$

Because g is surjective, there exists $\bar{v}$ such that $g(\bar{v}) = (-c + \eta, 0)$. In this case, the program (3.14) is exactly the program (3.13), with primal-dual solution set $\mathcal{X}(\bar{v}) \times \mathcal{Z}(\bar{v})$. We show next that $\hat{x}$ is a solution to (3.14) and thus in $\mathcal{X}(\bar{v})$. Clearly, $\hat{x}$ satisfies the constraints of (3.14). Since $\eta^T \nu_x = 0$ and points outward of $\{Ax = b, x \geq 0\}$, it must be that $\eta^T(\hat{x} - x) \leq 0$ for any $x \in \{Ax = b, x \geq 0\}$, which implies that $\eta^T \hat{x} \leq \eta^T x$. Thus, $\hat{x}$ is a solution to (3.14). Moreover, $\hat{x} + \lambda\nu_x$ is also a solution to (3.14) for any $\lambda \geq 0$ since (i) $\eta^T(\hat{x} + \lambda\nu_x) = \eta^T \hat{x}$ and (ii) $\hat{x} + \lambda\nu_x \in \{Ax = b, x \geq 0\}$. That is, $\mathcal{X}(\bar{v})$ is unbounded. Therefore, there is a point $(x_0, z_0) \in \mathcal{X}(\bar{v}) \times \mathcal{Z}(\bar{v})$, which is also an equilibrium of (3.12) by assumption, that is arbitrarily far from the set $\mathcal{X} \times \mathcal{Z}$. Clearly, $t \mapsto (x(t), z(t)) = (x_0, z_0)$ is an equilibrium trajectory of (3.12) starting from (x_0, z_0) when $v(t) = \bar{v}$. The fact that (x_0, z_0) can be made arbitrarily far from $\mathcal{X} \times \mathcal{Z}$ precludes the possibility of the dynamics from being ISS.

Next, we deal with case (ii), when $\{Ax = b, x \geq 0\}$ is bounded. Consider the linear program

$$\begin{aligned} \min \quad & -b^T z \\ \text{s.t.} \quad & A^T z \geq 0. \end{aligned}$$

Since $\{Ax = b, x \geq 0\}$ is bounded, Lemma 3.3.5 implies that $\{A^T z \geq 0\}$ is unbounded. Using an analogous approach as in case (i), one can find $\eta \in \mathbb{R}^m$ such that the set of solutions to

$$\begin{aligned} \min \quad & \eta^T z \\ \text{s.t.} \quad & A^T z \geq 0, \end{aligned} \tag{3.15}$$

is unbounded. Because g is surjective, there exists $\bar{v}$ such that $g(\bar{v}) = (-c, -b - \eta)$. In this case, the program (3.15) is the dual to (3.13), with primal-dual solution set $\mathcal{X}(\bar{v}) \times \mathcal{Z}(\bar{v})$. Since $\mathcal{Z}(\bar{v})$ is unbounded, one can find equilibrium trajectories of (3.12) under the disturbance $v(t) = \bar{v}$ that are arbitrarily far away from $\mathcal{X} \times \mathcal{Z}$, which contradicts ISS. □

Note that, in particular, the perturbed problem (3.11) and (3.13) coincide when

$$g(\overline{w}) = g(\overline{w}_x, \overline{w}_z) = (-\overline{w}_x - A^T\overline{w}_z, -\overline{w}_z).$$

Thus, by Theorem 3.3.4, the discontinuous saddle-point dynamics (3.10) is not ISS. Nevertheless, one can establish an ISS-like result for this dynamics under small enough and constant disturbances. We state this result next, where we also provide a quantifiable upper bound on the disturbances in terms of the solution set of some perturbed linear program.

Proposition 3.3.6. (ISS of discontinuous saddle-point dynamics under small constant disturbances). *Suppose there exists $\delta > 0$ such that the primal-dual solution set $\mathcal{X}(\overline{w}) \times \mathcal{Z}(\overline{w})$ of the perturbed problem (3.11) is nonempty for $\overline{w} \in \overline{\mathbb{B}}(0, \delta)$ and $\cup_{\overline{w} \in \overline{\mathbb{B}}(0,\delta)} \mathcal{X}(\overline{w}) \times \mathcal{Z}(\overline{w})$ is compact. Then there exists a continuous, zero-at-zero, and increasing function $\gamma : [0, \delta] \to \mathbb{R}_{\geq 0}$ such that, for all trajectories $t \mapsto (x(t), z(t))$ of (3.10) with constant disturbance $\overline{w} \in \overline{\mathbb{B}}(0, \delta)$, it holds that*

$$\lim_{t\to\infty} \|(x(t), z(t))\|_{\mathcal{X}\times\mathcal{Z}} \leq \gamma(\|\overline{w}\|).$$

Proof. Let $\gamma : [0, \delta] \to \mathbb{R}_{\geq 0}$ be given by

$$\gamma(r) := \max\left\{ \|(x, z)\|_{\mathcal{X}\times\mathcal{Z}} : (x, z) \in \bigcup_{\overline{w}\in\overline{\mathbb{B}}(0,r)} \mathcal{X}(\overline{w}) \times \mathcal{Z}(\overline{w}) \right\}.$$

By hypotheses, γ is well-defined. Note also that γ is increasing and satisfies $\gamma(0) = 0$. Next, we show that γ is continuous. By assumption, $\mathcal{X}(\overline{w}) \times \mathcal{Z}(\overline{w})$ is nonempty and bounded for every $\overline{w} \in \overline{\mathbb{B}}(0, \delta)$. Moreover, it is clear that $\mathcal{X}(\overline{w}) \times \mathcal{Z}(\overline{w})$ is closed for every $\overline{w} \in \overline{\mathbb{B}}(0, \delta)$ since we are considering linear programs in standard form. Thus, $\mathcal{X}(\overline{w}) \times \mathcal{Z}(\overline{w})$ is nonempty and compact for every $\overline{w} \in \overline{\mathbb{B}}(0, \delta)$. By [111, Corollary 11], these two conditions are sufficient for the set-valued map $\overline{w} \mapsto$

$\mathcal{X}(\overline{w}) \times \mathcal{Z}(\overline{w})$ to be continuous on $\overline{\mathbb{B}}(0,\delta)$. Since $r \mapsto \overline{\mathbb{B}}(0,r)$ is also continuous, [9, Proposition 1, pp. 41] ensures that the following set-valued composition map

$$r \mapsto \bigcup_{\overline{w} \in \overline{\mathbb{B}}(0,r)} \mathcal{X}(\overline{w}) \times \mathcal{Z}(\overline{w}),$$

is continuous (with compact values, by assumption). Therefore, [9, Theorem 6, pp. 53] guarantees then that γ is continuous on $\overline{\mathbb{B}}(0,\delta)$. Finally, to establish the bound on the trajectories, recall from Corollary 3.3.2 that each trajectory $t \mapsto (x(t), z(t))$ of (3.10) with constant disturbance $\overline{w} \in \overline{\mathbb{B}}(0,\delta)$ converges asymptotically to a point in $\mathcal{X}(\overline{w}) \times \mathcal{Z}(\overline{w})$. The distance between $\mathcal{X} \times \mathcal{Z}$ and the point in $\mathcal{X}(\overline{w}) \times \mathcal{Z}(\overline{w})$ to which the trajectory converges is upper bounded by

$$\lim_{t\to\infty} \|(x(t), z(t))\|_{\mathcal{X}\times\mathcal{Z}} \leq \max\{\|(x,z)\|_{\mathcal{X}\times\mathcal{Z}} : (x,z) \in \mathcal{X}(\overline{w}) \times \mathcal{Z}(\overline{w})\} \leq \gamma(\|\overline{w}\|),$$

which concludes the proof. □

3.3.2 Discontinuous saddle-point dynamics is integral input-to-state stable

Here we establish that the dynamics (3.10) possess a notion of robustness weaker than ISS, namely, integral input-to-state stability (iISS). Informally, iISS guarantees that disturbances with small energy give rise to small deviations from the equilibria. This is stated formally next.

Definition 3.3.7. (Integral input-to-state stability [6]). *The dynamics* (3.10) *is iISS with respect to the set* $\mathcal{X} \times \mathcal{Z}$ *if there exist functions* $\alpha \in \mathcal{K}_\infty, \beta \in \mathcal{KL}$, *and* $\gamma \in \mathcal{K}$ *such that, for any trajectory* $t \mapsto (x(t), z(t))$ *of* (3.10) *and all* $t \geq 0$, *one has*

$$\alpha(\|(x(t), z(t))\|_{\mathcal{X}\times\mathcal{Z}}) \leq \beta(\|(x(0), z(0)\|_{\mathcal{X}\times\mathcal{Z}}, t) + \int_0^t \gamma(\|w(s)\|)ds. \tag{3.16}$$

Our ensuing discussion is based on a suitable adaptation of the exposition in [6] to the setup of asymptotically stable sets for discontinuous dynamics. A useful tool for establishing iISS is the notion of iISS Lyapunov function, whose definition we review next.

Definition 3.3.8. (iISS Lyapunov function). *A differentiable function* $V : \mathbb{R}^{n+m} \to \mathbb{R}_{\geq 0}$ *is an iISS Lyapunov function with respect to the set* $\mathcal{X} \times \mathcal{Z}$ *for dynamics* (3.10) *if there exist functions* $\alpha_1, \alpha_2 \in \mathcal{K}_\infty, \sigma \in \mathcal{K}$*, and a continuous positive definite function* α_3 *such that*

$$\alpha_1(\|(x,z)\|_{\mathcal{X}\times\mathcal{Z}}) \leq V(x,z) \leq \alpha_2(\|(x,z)\|_{\mathcal{X}\times\mathcal{Z}}), \tag{3.17a}$$

$$a \leq -\alpha_3(\|(x,z)\|_{\mathcal{X}\times\mathcal{Z}}) + \sigma(\|w\|), \tag{3.17b}$$

for all $a \in \mathcal{L}_{\mathcal{F}[f^w_{dis}]} V(x,z)$ *and* $x \in \mathbb{R}^n, z \in \mathbb{R}^m, w \in \mathbb{R}^{n+m}$.

Note that, since the set $\mathcal{X} \times \mathcal{Z}$ is compact by assumption, (3.17a) is equivalent to V being proper with respect to $\mathcal{X} \times \mathcal{Z}$. The existence of an iISS Lyapunov function is critical in establishing iISS, as the following result states.

Theorem 3.3.9. (iISS Lyapunov function implies iISS). *If there exists an iISS Lyapunov function with respect to* $\mathcal{X} \times \mathcal{Z}$ *for* (3.10)*, then the dynamics is iISS with respect to* $\mathcal{X} \times \mathcal{Z}$*.*

This result is stated in [6, Theorem 1] for the case of differential equations with locally Lipschitz right-hand side and asymptotically stable origin, but its extension to discontinuous dynamics and asymptotically stable sets, as considered here, is straightforward. We rely on Theorem 3.3.9 to establish that the discontinuous saddle-point dynamics (3.10) is iISS. Interestingly, the function V employed to characterize the convergence properties of the unperturbed dynamics in Section 3.2 is not an iISS Lyapunov function (in fact, our proof of Theorem 3.2.1 relies on the set-valued LaSalle Invariance Principle because, essentially, the Lie derivative of V is not negative definite). Nevertheless, in the proof of the next result, we build on the properties of this function with respect to the dynamics to identify a suitable iISS Lyapunov function for (3.10).

Theorem 3.3.10. (iISS of saddle-point dynamics). *The dynamics* (3.10) *is iISS with respect to* $\mathcal{X} \times \mathcal{Z}$*.*

Proof. We proceed by progressively defining functions V_{euc}, $V^{\text{rep}}_{\text{euc}}$, V_{CLF}, and $V^{\text{rep}}_{\text{CLF}} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$. The rationale for our construction is as follows. Our starting point is the squared Euclidean distance from the primal-dual solution set, denoted V_{euc}. The function $V_{\text{euc}}^{\text{rep}}$ is a reparameterization of V_{euc} (which remains radially unbounded with respect to $\mathcal{X} \times \mathcal{Z}$) so that state and disturbance appear separately in the (set-valued) Lie derivative. However, since V_{euc} is only a LaSalle-type function, this implies that only the disturbance appears in the Lie derivative of $V_{\text{euc}}^{\text{rep}}$. Nevertheless, via a Converse Lyapunov Theorem, we identify an additional function V_{CLF} whose reparameterization $V_{\text{CLF}}^{\text{rep}}$ has a Lie derivative where both state and disturbance appear. The function $V_{\text{CLF}}^{\text{rep}}$, however, may not be radially unbounded with respect to $\mathcal{X} \times \mathcal{Z}$. This leads us to the construction of the iISS Lyapunov function as $V = V_{\text{euc}}^{\text{rep}} + V_{\text{CLF}}^{\text{rep}}$.

We begin by defining the differentiable function V_{euc}

$$V_{\text{euc}}(x, z) = \min_{(x_*, z_*) \in \mathcal{X} \times \mathcal{Z}} \frac{1}{2}(x - x_*)^T (x - x_*) + \frac{1}{2}(z - z_*)^T (z - z_*).$$

Since $\mathcal{X} \times \mathcal{Z}$ is convex and compact, applying Theorem 2.5.3 one gets $\nabla V_{\text{euc}}(x, z) = (x - x_*(x, z), z - z_*(x, z))$, where

$$(x_*(x, z), z_*(x, z)) = \underset{(x_*, z_*) \in \mathcal{X} \times \mathcal{Z}}{\operatorname{argmin}} \frac{1}{2}(x - x_*)^T (x - x_*) + \frac{1}{2}(z - z_*)^T (z - z_*).$$

It follows from Theorem 3.2.1 and Proposition 3.2.4 that $\mathcal{L}_{\mathcal{F}[f_{\text{dis}}]} V_{\text{euc}}(x, z) \subset (-\infty, 0]$ for all $(x, z) \in \mathbb{R}^n_{\geq 0} \times \mathbb{R}^m$. Next, define the function $V_{\text{euc}}^{\text{rep}}$ by

$$V_{\text{euc}}^{\text{rep}}(x, z) = \int_0^{V_{\text{euc}}(x, z)} \frac{dr}{1 + \sqrt{2r}}.$$

Clearly, $V_{\text{euc}}^{\text{rep}}(x, z)$ is positive definite with respect to $\mathcal{X} \times \mathcal{Z}$. Also, $V_{\text{euc}}^{\text{rep}}(x, z)$ is radially unbounded with respect to $\mathcal{X} \times \mathcal{Z}$ because (i) $V_{\text{euc}}(x, z)$ is radially unbounded with respect to $\mathcal{X} \times \mathcal{Z}$ and (ii) $\lim_{y \to \infty} \int_0^y \frac{dr}{1+\sqrt{2r}} = \infty$. In addition, for any $a \in \mathcal{L}_{\mathcal{F}[f_{\text{dis}}^w]} V_{\text{euc}}^{\text{rep}}(x, z)$ and $(x, z) \in \mathbb{R}^n_{\geq 0} \times \mathbb{R}^m$, one has

$$a \leq \frac{\sqrt{2V_{\text{euc}}(x, z)}\|w\|}{1 + \sqrt{2V_{\text{euc}}(x, z)}} \leq \|w\|. \tag{3.18}$$

Next, we define the function V_{CLF}. Since $\mathcal{X} \times \mathcal{Z}$ is compact and globally asymptotically stable for (3.8) $(\dot{x}, \dot{z}) = \mathcal{F}[f_{\text{dis}}^w](x, z)$ when $w \equiv 0$ (cf. Corollary 3.2.5) the Converse Lyapunov Theorem [26, Theorem 3.13] ensures the existence of a smooth function $V_{\mathrm{CLF}} : \mathbb{R}^{n+m} \to \mathbb{R}_{\geq 0}$ and class $\mathcal{K}_\infty$ functions $\tilde{\alpha}_1$, $\tilde{\alpha}_2$, $\tilde{\alpha}_3$ such that

$$\begin{aligned}\tilde{\alpha}_1(\|(x,z)\|_{\mathcal{X}\times\mathcal{Z}}) \leq V_{\mathrm{CLF}}(x,z) \leq \tilde{\alpha}_2(\|(x,z)\|_{\mathcal{X}\times\mathcal{Z}}),\\ a \leq -\tilde{\alpha}_3(\|(x,z)\|_{\mathcal{X}\times\mathcal{Z}}),\end{aligned}$$

for all $a \in \mathcal{L}_{\mathcal{F}[f_{\mathrm{dis}}]}V_{\mathrm{CLF}}(x,z)$ and $(x,z) \in \mathbb{R}^n_{\geq 0} \times \mathbb{R}^m$. Thus, when $w \not\equiv 0$, for $a \in \mathcal{L}_{\mathcal{F}[f^w_{\mathrm{dis}}]}V_{\mathrm{CLF}}(x,z)$ and $(x,z) \in \mathbb{R}^n_{\geq 0} \times \mathbb{R}^m$, we have

$$\begin{aligned} a &\leq -\tilde{\alpha}_3(\|(x,z)\|_{\mathcal{X}\times\mathcal{Z}}) + \nabla V_{\mathrm{CLF}}(x,z)w,\\ &\leq -\tilde{\alpha}_3(\|(x,z)\|_{\mathcal{X}\times\mathcal{Z}}) + \|\nabla V_{\mathrm{CLF}}(x,z)\| \cdot \|w\|,\\ &\leq -\tilde{\alpha}_3(\|(x,z)\|_{\mathcal{X}\times\mathcal{Z}}) + (\|(x,z)\|_{\mathcal{X}\times\mathcal{Z}} + \|\nabla V_{\mathrm{CLF}}(x,z)\|) \cdot \|w\|,\\ &\leq -\tilde{\alpha}_3(\|(x,z)\|_{\mathcal{X}\times\mathcal{Z}}) + \lambda(\|(x,z)\|_{\mathcal{X}\times\mathcal{Z}}) \cdot \|w\|,\end{aligned}$$

where $\lambda : [0,\infty) \to [0,\infty)$ is given by

$$\lambda(r) = r + \max_{\|\eta\|_{\mathcal{X}\times\mathcal{Z}} \leq r} \|\nabla V_{\mathrm{CLF}}(\eta)\|.$$

Since V_{CLF} is smooth, λ is a class $\mathcal{K}$ function. Next, define

$$V^{\mathrm{rep}}_{\mathrm{CLF}}(x,z) = \int_0^{V_{\mathrm{CLF}}(x,z)} \frac{dr}{1 + \lambda \circ \tilde{\alpha}_1^{-1}(r)}.$$

Without additional information about $\lambda \circ \tilde{\alpha}_1^{-1}$, one cannot determine if $V^{\mathrm{rep}}_{\mathrm{CLF}}$ is radially unbounded with respect to $\mathcal{X} \times \mathcal{Z}$ or not. Nevertheless, $V^{\mathrm{rep}}_{\mathrm{CLF}}$ is positive definite with respect to $\mathcal{X} \times \mathcal{Z}$. Then for any $a \in \mathcal{L}_{\mathcal{F}[f^w_{\mathrm{dis}}]}V^{\mathrm{rep}}_{\mathrm{CLF}}(x,z)$ and $(x,z) \in \mathbb{R}^n_{\geq 0} \times \mathbb{R}^m$ we have,

$$\begin{aligned} a &\leq \frac{-\tilde{\alpha}_3(\|(x,z)\|_{\mathcal{X}\times\mathcal{Z}}) + \nabla V_{\mathrm{CLF}}(x,z)w}{1 + \lambda \circ \tilde{\alpha}_1^{-1}(V_{\mathrm{CLF}}(x,z))},\\ &\leq \frac{-\tilde{\alpha}_3(\|(x,z)\|_{\mathcal{X}\times\mathcal{Z}})}{1 + \lambda \circ \tilde{\alpha}_1^{-1} \circ \tilde{\alpha}_2(\|(x,z)\|_{\mathcal{X}\times\mathcal{Z}})} + \frac{\lambda(\|(x,z)\|_{\mathcal{X}\times\mathcal{Z}})}{1 + \lambda(\|(x,z)\|_{\mathcal{X}\times\mathcal{Z}})}\|w\|,\\ &\leq -\rho(\|(x,z)\|_{\mathcal{X}\times\mathcal{Z}}) + \|w\|, \end{aligned} \tag{3.19}$$

where ρ is the positive definite function given by

$$\rho(r) = \tilde{\alpha}_3(r)/(1 + \lambda \circ \tilde{\alpha}_1^{-1} \circ \tilde{\alpha}_2(r)).$$

We now show that $V = V^{\mathrm{rep}}_{\mathrm{euc}} + V^{\mathrm{rep}}_{\mathrm{CLF}}$ is an iISS Lyapunov function for (3.10) with respect to $\mathcal{X} \times \mathcal{Z}$. First, (3.17a) is satisfied because V is positive definite and radially unbounded with respect to $\mathcal{X} \times \mathcal{Z}$ since (i) $V_{\text{euc}}^{\text{rep}}$ is positive definite and radially unbounded with respect to $\mathcal{X} \times \mathcal{Z}$ and (ii) $V_{\text{CLF}}^{\text{rep}}$ is positive definite with respect to $\mathcal{X} \times \mathcal{Z}$. Second, (3.17b) is satisfied as a result of the combination of (3.18) and (3.19). Since V satisfies the conditions of Theorem 3.3.9, (3.10) is iISS. □

Based on the discussion in Section 3.3.1, the iISS property of (3.10) is an accurate representation of the robustness of the dynamics, not a limitation of our analysis. A consequence of iISS is that the asymptotic convergence of the dynamics is preserved under finite energy disturbances [98, Proposition 6]. In the case of (3.10), a stronger convergence property is true under finite variation disturbances (which do not have finite energy). The following formalizes this fact.

Corollary 3.3.11. (Finite variation disturbances). *Suppose* $w : \mathbb{R}_{\geq 0} \to \mathbb{R}^n \times \mathbb{R}^m$ *is such that* $\int_0^\infty \|w(s) - \overline{w}\| ds < \infty$ *for some* $\overline{w} = (\overline{w}_x, \overline{w}_z) \in \mathbb{R}^n \times \mathbb{R}^m$. *Assume that* $\mathcal{X}(\overline{w}) \times \mathcal{Z}(\overline{w})$ *is nonempty and compact. Then each trajectory of* (3.10) *under the disturbance* w *converges asymptotically to a point in* $\mathcal{X}(\overline{w}) \times \mathcal{Z}(\overline{w})$.

Proof. Let $f_{\text{dis,pert}}^{v}$ be the discontinuous saddle-point dynamics derived for the perturbed program (3.11) associated to $\overline{w}$ with additive disturbance $v : \mathbb{R}_{\geq 0} \to \mathbb{R}^n \times \mathbb{R}^m$. By Corollary 3.3.2, $\mathcal{X}(\overline{w}) \times \mathcal{Z}(\overline{w}) \neq \emptyset$ is globally asymptotically stable for $f_{\text{dis,pert}}^{0}$. Additionally, by Theorem 3.3.10 and since $\mathcal{X}(\overline{w}) \times \mathcal{Z}(\overline{w})$ is compact, $f_{\text{dis,pert}}^{v}$ is iISS. As a consequence, by [98, Proposition 6], each trajectory of $f_{\text{dis,pert}}^{v}$ converges asymptotically to a point in $\mathcal{X}(\overline{w}) \times \mathcal{Z}(\overline{w})$ if $\int_0^\infty \|v(s)\| ds < \infty$. The result now follows by noting that f_{dis}^{w} with disturbance w is exactly $f_{\text{dis,pert}}^{v}$ with disturbance $v = w - \overline{w}$ and that, by assumption, the latter disturbance satisfies $\int_0^\infty \|v(s)\| ds < \infty$. □

3.4 Robustness in recurrently connected graphs

In this section, we build on the iISS properties of the saddle-point dynamics (3.3) to study its convergence under communication link failures. As such, agents do not receive updated state information from their neighbors at all times and use the last known value of their state to implement the dynamics. The link failure model we considered is described by recurrently connected graphs (RCG), in which periods of communication loss are followed by periods of connectivity. We formalize this notion next.

Definition 3.4.1. (Recurrently connected graphs). *Given a strictly increasing sequence of times* $\{t_k\}_{k=0}^{\infty} \subset \mathbb{R}_{\geq 0}$ *and a base graph* $\mathcal{G}_b = (\mathcal{V}, \mathcal{E}_b)$*, we call* $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$ recurrently connected with respect to $\mathcal{G}_b$ and $\{t_k\}_{k=0}^{\infty}$ *if* $\mathcal{E}(t) \subseteq \mathcal{E}_b$ *for all* $t \in [t_{2k}, t_{2k+1})$ *while* $\mathcal{E}(t) \supseteq \mathcal{E}_b$ *for all* $t \in [t_{2k+1}, t_{2k+2})$*,* $k \in \mathbb{Z}_{\geq 0}$*.*

Intuitively, one may think of $\mathcal{G}_b$ as a graph over which (3.8) is distributed: during time intervals of the form $[t_{2k}, t_{2k+1})$, links are failing and hence the network cannot execute the algorithm properly, whereas during time intervals of the form $[t_{2k+1}, t_{2k+2})$, enough communication links are available to implement it correctly. In what follows, and for simplicity of presentation, we only consider the worst-case link failure scenario: i.e., if a link fails during the time interval $[t_{2k}, t_{2k+1})$, it remains down during its entire duration. The results stated here also apply to the general scenarios where edges may fail and reconnect multiple times within a time interval.

In the presence of link failures, the implementation of the evolution of the z variables, cf. (3.9), across different agents would yield in general different outcomes (given that different agents have access to different information at different times). To avoid this problem, we assume that, for each $\ell \in \{1, \dots, m\}$, the agent with minimum identifier index,

$$j = \mathbb{S}(\ell) := \min\{i \in \{1, \dots, n\} : a_{\ell,i} \neq 0\},$$

implements the z_ℓ-dynamics and communicates this value when communication is available to its neighbors. Incidentally, only neighbors of $j = \mathbb{S}(\ell)$ need to know z_ℓ. With this convention in place, we may describe the network dynamics under link failures. Let $\mathbb{F}(k)$ be the set of failing communication edges for $t \in [t_k, t_{k+1})$. In other words, if $(i, j) \in \mathbb{F}(k)$ then agents i and j do not receive updated state information from each other during the whole interval $[t_k, t_{k+1})$. The nominal flow function of i on a RCG for $t \in [t_k, t_{k+1})$ is

$$f_i^{\text{nom,RCG}}(x,z) = -c_i - \sum_{\substack{\ell=1 \\ (i,\mathbb{S}(\ell))\notin\mathbb{F}(k)}}^{m} a_{\ell,i} z_\ell - \sum_{\substack{\ell=1 \\ (i,\mathbb{S}(\ell))\in\mathbb{F}(k)}}^{m} a_{\ell,i} z_\ell(t_k)$$
$$- \sum_{\ell=1}^{m} a_{\ell,i} \Big[\sum_{\substack{j=1 \\ (i,j)\notin\mathbb{F}(k)}}^{n} a_{\ell,j} x_j + \sum_{\substack{j=1 \\ (i,j)\in\mathbb{F}(k)}}^{n} a_{\ell,j} x_j(t_k) - b_\ell \Big].$$

Thus the x_i-dynamics during $[t_k, t_{k+1})$ for $i \in \{1,\dots,n\}$ is

$$\dot{x}_i = \begin{cases} f_i^{\text{nom,RCG}}(x,z), & \text{if } x_i > 0, \\ \max\{0, f_i^{\text{nom,RCG}}(x,z)\}, & \text{if } x_i = 0. \end{cases} \tag{3.20a}$$

Likewise, the z-dynamics for $\ell \in \{1,\dots,m\}$ is

$$\dot{z}_\ell = \sum_{\substack{i=1 \\ (i,\mathbb{S}(\ell))\notin\mathbb{F}(k)}}^{n} a_{\ell,i} x_i + \sum_{\substack{i=1 \\ (i,\mathbb{S}(\ell))\in\mathbb{F}(k)}}^{n} a_{\ell,i} x_i(t_k) - b_\ell. \tag{3.20b}$$

It is worth noting that (3.20) and (3.8) coincide when $\mathbb{F}(k) = \emptyset$. The next result shows that the discontinuous saddle-point dynamics still converge under recurrently connected graphs.

Proposition 3.4.2. (Convergence of saddle-point dynamics under RCGs). *Let $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$ be recurrently connected with respect to $\mathcal{G}_b = (\mathcal{V}, \mathcal{E}_b)$ and $\{t_k\}_{k=0}^{\infty}$. Suppose that (3.20) is distributed over $\mathcal{G}_b$ and $T_{\text{disconnected}}^{\max} := \sup_{k\in\mathbb{Z}_{\geq 0}}(t_{2k+1} - t_{2k}) < \infty$. Let $t \mapsto (x(t), z(t))$ be a trajectory of (3.20). Then there exists $T_{\text{connected}}^{\min} > 0$ (depending on $T_{\text{disconnected}}^{\max}$, $x(t_0)$, and $z(t_0)$) such that $\inf_{k\in\mathbb{Z}_{\geq 0}}(t_{2k+2} - t_{2k+1}) > T_{\text{connected}}^{\min}$ implies that $\|(x(t_{2k}), z(t_{2k}))\|_{\mathcal{X}\times\mathcal{Z}} \to 0$ as $k \to \infty$.*

Proof. The proof method is to (i) show that trajectories of (3.20) do not escape in finite time and (ii) use a $\mathcal{KL}$ characterization of asymptotically stable dynamics [26] to find $T_{\text{connected}}^{\min}$ for which $\|(x(t_{2k}), z(t_{2k}))\|_{\mathcal{X}\times\mathcal{Z}} \to 0$ as $k \to \infty$. To prove (i), note that (3.20) represents a switched system of affine differential equations. The modes are defined by all κ-combinations of link failures (for $\kappa = 1,\dots,|\mathcal{E}_b|$) and all κ-combinations of agents (for $\kappa = 1,\dots,n$). Thus, the number of modes is

$d := 2^{|\mathcal{E}_b|+n}$. Assign to each mode a number in the set $\{1,\ldots,d\}$. Then, for any given $t \in [t_k, t_{k+1})$, the dynamics (3.20) is equivalently represented as

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = P_{\sigma(t)} \begin{bmatrix} x \\ z \end{bmatrix} + q_{\sigma(t)}(x(t_k), z(t_k)),$$

where $\sigma : \mathbb{R}_{\geq 0} \to \{1,\ldots,d\}$ is a switching law and $P_{\sigma(t)}$ (resp. $q_{\sigma(t)}$) is the flow matrix (resp. drift vector) of (3.20) for mode $\sigma(t)$. Let $\rho = \|(x(t_0), z(t_0))\|_{\mathcal{X}\times\mathcal{Z}}$ and define

$$\tilde{q} := \max_{\substack{p\in\{1,\ldots,d\} \\ \|(x,z)\|_{\mathcal{X}\times\mathcal{Z}}\leq\rho}} \|q_p(x,z)\|,$$

$$\tilde{\mu} := \max_{p\in\{1,\ldots,d\}} \mu(P_p),$$

where $\mu(P_p) = \lim_{h\to 0^+} \frac{\|I-hP_p\|-1}{h}$ is the logarithmic norm of P_p. Both $\tilde{q}$ and $\tilde{\mu}$ are finite. Consider an arbitrary interval $[t_{2k}, t_{2k+1})$ where $\|(x(t_{2k}), z(t_{2k}))\|_{\mathcal{X}\times\mathcal{Z}} \leq \rho$. In what follows, we make use of the fact that the trajectory of an affine differential equation $\dot{y} = \mathcal{A}y + \beta$ for $t \geq t_0$ is

$$y(t) = e^{\mathcal{A}(t-t_0)}y(t_0) + \int_{t_0}^{t} e^{\mathcal{A}(t-s)}\beta ds. \tag{3.21}$$

Applying (3.21), we derive the following bound,

$$\begin{aligned} &\|(x(t_{2k+1}), z(t_{2k+1})) - (x(t_{2k}), z(t_{2k}))\| \\ &\qquad\leq \|(x(t_{2k}), z(t_{2k}))\|(e^{\tilde{\mu}(t_{2k+1}-t_{2k})} - 1) + \int_{t_{2k}}^{t_{2k+1}} e^{\tilde{\mu}(t_{2k+1}-s)}\tilde{q}ds, \\ &\qquad\leq (\rho + \tilde{q}/\tilde{\mu})(e^{\tilde{\mu}T^{\max}_{\text{disconnected}}} - 1) =: M. \end{aligned}$$

In words, M bounds the distance that trajectories travel on intervals of link failures. Also, M is valid for all such intervals where $\|(x(t_{2k}), z(t_{2k}))\|_{\mathcal{X}\times\mathcal{Z}} \leq \rho$. Next, we address the proof of (ii) by designing $T^{\min}_{\text{connected}}$ to enforce this condition. By definition, $\|(x(t_0), z(t_0))\|_{\mathcal{X}\times\mathcal{Z}} = \rho$. Thus, $\|(x(t_1), z(t_1)) - (x(t_0), z(t_0))\| \leq M$. Given that $\mathcal{X} \times \mathcal{Z}$ is globally asymptotically stable for (3.20) if $\mathbb{F}(k) = \emptyset$ (cf. Corollary 3.2.5), [26, Theorem 3.13] implies the existence of $\beta \in \mathcal{KL}$ such that

$$\|(x(t), z(t))\|_{\mathcal{X}\times\mathcal{Z}} \leq \beta(\|(x(t_0), z(t_0))\|_{\mathcal{X}\times\mathcal{Z}}, t).$$

By [98, Proposition 7], there exist $\theta_1, \theta_2 \in \mathcal{K}_\infty$ such that $\beta(s,t) \leq \theta_1(\theta_2(s)e^{-t})$. Thus,

$$\begin{aligned}\alpha(\|(x(t_2), z(t_2))\|_{\mathcal{X}\times\mathcal{Z}}) &\leq \theta_1(\theta_2(\|(x(t_1), z(t_1))\|_{\mathcal{X}\times\mathcal{Z}})e^{-t_2+t_1}),\\ &\leq \theta_1(\theta_2(\rho + M)e^{-t_2+t_1}).\end{aligned}$$

Consequently, if

$$t_2 - t_1 > T^{\min}_{\text{connected}} := \ln\left(\frac{\theta_2(\rho + M)}{\theta_1^{-1}(\alpha(\rho))}\right) > 0,$$

then $\|(x(t_2), z(t_2))\|_{\mathcal{X}\times\mathcal{Z}} < \rho$. Repeating this analysis reveals that $\|(x(t_{2k+2}), z(t_{2k+2}))\|_{\mathcal{X}\times\mathcal{Z}} < \|(x(t_{2k}), z(t_{2k}))\|_{\mathcal{X}\times\mathcal{Z}}$ for all $k \in \mathbb{Z}_{\geq 0}$ when $t_{2k+2} - t_{2k+1} > T^{\min}_{\text{connected}}$. Thus $\|(x(t_{2k}), z(t_{2k}))\|_{\mathcal{X}\times\mathcal{Z}} \to 0$ as $k \to \infty$ as claimed. □

Remark 3.4.3. (More general link failures). Proposition 3.4.2 shows that, as long as the communication graph is connected with respect to A for a sufficiently long time after periods of failure, the discontinuous saddle-point dynamics converge. We have observed in simulations, however, that the dynamics is not robust to more general link failures such as when the communication graph is never connected with respect to A but its union over time is. We believe the reason is the lack of consistency in the $z-$dynamics for all time across agents in this case. •

3.5 Simulations

Here we illustrate the convergence and robustness properties of the discontinuous saddle-point dynamics. We consider a finite-horizon optimal control problem for a network of agents with coupled dynamics and underactuation. The network-wide dynamics is open-loop unstable and the aim of the agents is to find a control to minimize the actuation effort and ensure the network state remains small. To achieve this goal, the agents use the discontinuous saddle-point dynamics (3.8). Formally, consider the following finite-horizon optimal control problem,

$$\min \quad \sum_{\tau=0}^{T} \|x(\tau+1)\|_1 + \|u(\tau)\|_1 \tag{3.22a}$$

$$\text{s.t.} \quad x(\tau+1) = Gx(\tau) + Hu(\tau), \quad \tau = 0, \ldots T, \tag{3.22b}$$

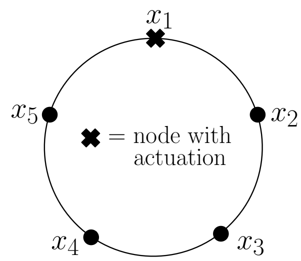


Figure 3.2: Network topology of the multi-agent system defined by dynamics (3.24). The presence of a communication link among every pair of agents whose dynamics are coupled in (3.24) ensures that the algorithm (3.8) is distributed over the communication graph.

where $x(\tau) \in \mathbb{R}^N$ and $u(\tau) \in \mathbb{R}^N$ is the network state and control, respectively, at time τ. The initial point $x_i(0)$ is known to agent i and its neighbors. The matrices $G \in \mathbb{R}^{N\times N}$ and $H = \text{diag}(h) \in \mathbb{R}^{N\times N}$, $h \in \mathbb{R}^N$, define the network evolution, and the network topology is encoded in the sparsity structure of G. We interpret each agent as a subsystem whose dynamics is influenced by the states of neighboring agents. An agent knows the dynamics of its own subsystem and its neighbor's subsystem, but does not know the entire network dynamics. A solution to (3.22) is a time history of optimal controls $(u_*(0), \dots, u_*(T)) \in (\mathbb{R}^N)^T$.

To express this problem in standard linear programming form (2.8), we split the states into their positive and negative components, $x(\tau) = x^+(\tau) - x^-(\tau)$, with $x^+(\tau), x^-(\tau) \geq 0$ (and similarly for the inputs $u(\tau)$). Then, (3.22) can be equivalently formulated as the following linear program,

$$\min \quad \sum_{\tau=0}^{T}\sum_{i=1}^{N} x_i^+(\tau+1) + x_i^-(\tau+1) + u_i^+(\tau) + u_i^-(\tau) \tag{3.23a}$$

$$\text{s.t.} \quad x^+(\tau+1) - x^-(\tau) = G(x^+(\tau) - x^-(\tau)) + H(u^+(\tau) - u^-(\tau)), \quad \tau = 0, \dots, T, \tag{3.23b}$$

$$x^+(\tau+1), x^-(\tau+1), u^+(\tau), u^-(\tau) \geq 0, \quad \tau = 0, \dots, T. \tag{3.23c}$$

The optimal control for (3.22) at time τ is then $u_*(\tau) = u_*^+(\tau) - u_*^-(\tau)$, where the vector $(u_*^+(0), u_*^-(0), \dots, u_*^+(T), u_*^-(T))$ is a solution to (3.23), cf. [36, Lemma 6.1].

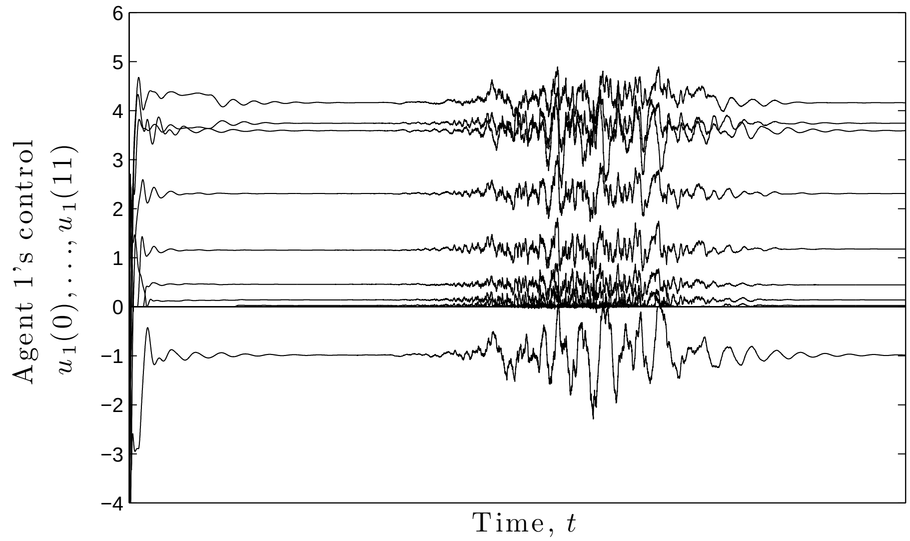


Figure 3.3: Trajectories of the discontinuous saddle-point dynamics (3.10) for agent 1 as it computes its time history of optimal controls. The noise depicted in Figure 3.4 was applied to the dynamics.

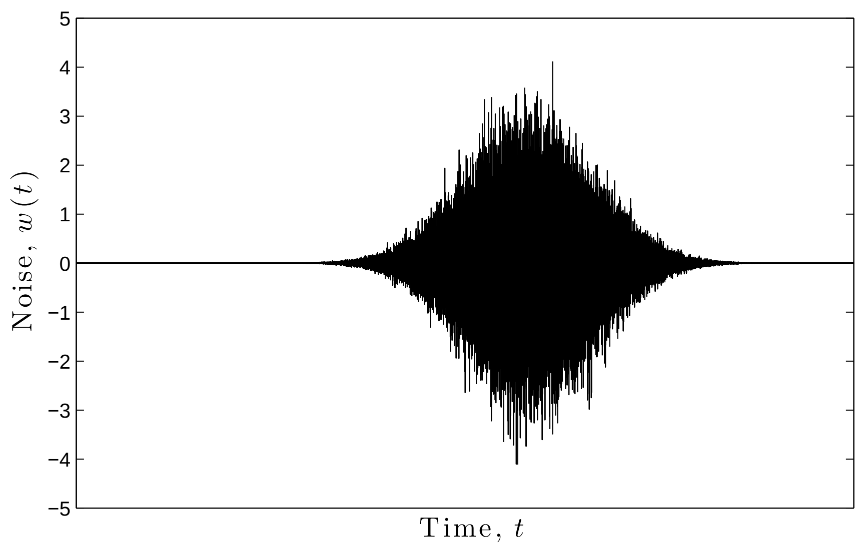


Figure 3.4: Noise that was applied to agents' dynamics resulting in the disturbances observed in the trajectoriies of Figure 3.3.

We implement the discontinuous saddle-point dynamics (3.8) for problem (3.23) over a network of 5 agents described by the dynamics

$$x(\tau+1) = \begin{bmatrix} 0.5 & 0 & 0 & 0 & 0.7 \\ 0.7 & 0.5 & 0 & 0 & 0 \\ 0 & 0.7 & 0.5 & 0 & 0 \\ 0 & 0 & 0.7 & 0.5 & 0 \\ 0 & 0 & 0 & 0.7 & 0.5 \end{bmatrix} x(\tau) + \mathrm{diag}\left(\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}\right) u(\tau). \tag{3.24}$$

Note that this dynamics is underactuated and open-loop unstable but controllable. Figure 3.2 reveals the network topology induced by this dynamics. Note that, when implementing this dynamics, agent $i \in \{1, \ldots, 5\}$ computes the time history of its optimal control, $u_i^-(0), u_i^+(0), \ldots, u_i^-(T), u_i^+(T)$, as well as the time history of its states, $x_i^-(1), x_i^+(1), \ldots, x_i^-(T+1), x_i^+(T+1)$. With respect to the solution of the optimal control problem, the time history of states are auxiliary variables used in the discontinuous dynamics and can be discarded after the control is determined. Figure 3.3 shows the results of the implementation of (3.8) when a finite energy noise signal, shown in Figure 3.4, disturbs the agents' execution. Convergence is achieved initially in the absence of noise. Then, the finite energy noise signal enters each agents' dynamics and disrupts this convergence, albeit not significantly due to the iISS property of (3.10) characterized in Theorem 3.3.10. Once the noise disappears, convergence ensues. The time horizon of the problem is $T = 11$. The 12 trajectories represent agent 1's evolving estimates of the optimal controls $u_1(0), \ldots, u_1(11)$. The steady-state values achieved by these trajectories correspond to the solution of (3.22). Once determined, these controls are implemented by agent 1, resulting in the network evolution depicted in Figure 3.5. Clearly, agent 1 is able to drive the open-loop unstable system state to zero.

Figure 3.6 shows the results of the implementation of (3.8) when the communication graph is recurrently connected, where convergence is achieved in agreement with Proposition 3.4.2. The link failure model here is a random number of random links failing during times of disconnection. The graph is repeatedly connected for $1s$ and then disconnected for $4s$ (i.e., the ratio $T^{\max}_{\text{disconnected}} : T^{\min}_{\text{connected}}$ is $4 : 1$). The fact that convergence is still achieved under this unfavorable ratio highlights the strong robustness properties of the algorithm.

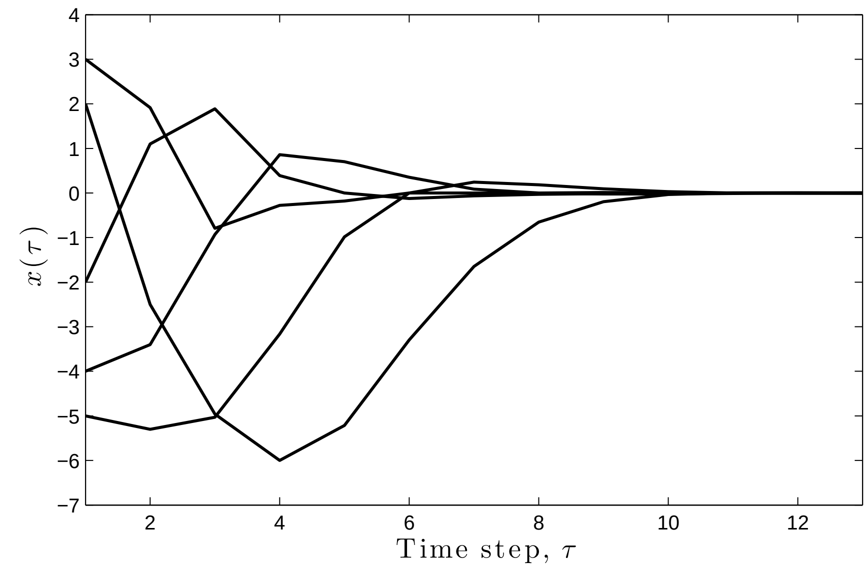


Figure 3.5: Once the optimal control is determined (the steady state values depicted in Figure 3.3 and 3.6), these controls are then implemented by agent 1 and result in the network evolution (3.22b) depicted above.

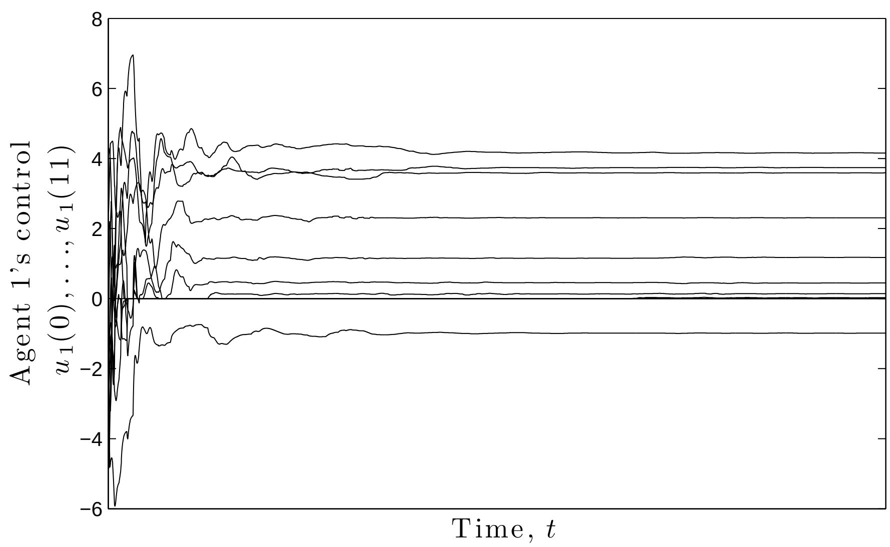


Figure 3.6: Trajectories of the dynamics under a recurrently connected communication graph where a random number of random links failed during periods of disconnection.

This concludes our study of robust distributed linear programming.

Chapter 3, in part, is a reprint of the material [80] as it appears in 'Distributed linear programming and bargaining in exchange networks' by D. Richert and J. Cortés in the proceedings of the 2013 American Control Conference as well as [81] as it appears in 'Integral input-to-state stable saddle-point dynamics for distributed linear programming' by D. Richert and J. Cortés in the proceedings of the 2013 IEEE Conference on Decision and Control as well as [83] as it appears in 'Robust distributed linear programming' by D. Richert and J. Cortés which was submitted to the IEEE Transactions on Automatic Control. The dissertation author was the primary investigator and author of these papers.

Chapter 4

Distributed event-triggered linear programming

From an analysis viewpoint, the availability of powerful concepts and tools from stability analysis made the development of the continuous-time coordination algorithm (3.8) of the previous chapter appealing. However, its implementation requires the continuous flow of information among agents. To relax this requirement, the goal of this chapter is to synthesize realistic communication protocols for the exchange of information between agents. One plausible approach would be to design the Euler discretization of the continuous-time algorithm, determine an appropriate stepsize to ensure convergence, and, based on it, have agents synchronously and periodically broadcast their states. Besides requiring synchronous state broadcasting, making this solution unattractive is the fact that the stepsize has to take into account worst-case situations, leading to an inefficient use of the network resources. Rather, our preferred solution is to develop a set of state-based rules, termed triggers, that individual agents use to determine when to opportunistically broadcast their state to neighboring agents while ensuring asymptotic convergence to a solution of the linear program.

As a preliminary development, we design a centralized trigger law whereby agents use global knowledge of the network to determine when to synchronously broadcast their state. The characterization of the convergence properties of the centralized implementation is challenging for two reasons: (i) the original continuous-time dynamics is discontinuous in the agents' state and (ii) a common smooth Lyapunov function is unknown. Regarding (ii), the LaSalle function used in the previous chapter is inadequate to prove convergence because there are certain points in the state space where arbitrarily fast state broadcasting is required to ensure its monotonicity. Nevertheless, using concepts and tools from switched and hybrid systems, we are able to overcome these obstacles by introducing a discontinuous Lyapunov function and examining its evolution during time intervals where state broadcasts do not occur.

We then investigate to what extent the centralized triggers can be implemented in a distributed way and modify them when necessary. In doing so, we face the additional difficulty posed by the fact that the mode switches associated to the discontinuity of the original dynamics are not locally detectable by individual agents. To address this challenge, we bound the evolution of the Lyapunov function under mode mismatch and, based on this understanding, design the distributed triggers so that any potential increase of the Lyapunov function due to the mismatch is compensated by the decrease in its value before the mismatch took place. Moreover, the distributed character of the agent triggers leads to asynchronous state broadcasts, which poses an additional challenge for both design and analysis. Our main result establishes the asymptotic convergence of the distributed implementation and identifies sufficient conditions for executions to be persistently flowing (that is, state broadcasts are separated by a uniform time infinitely often). We show that the asynchronous state broadcasts cannot be the cause of non-persistently flowing executions and we conjecture that all executions are in fact persistently flowing. As a byproduct of using a hybrid systems modeling framework in our technical approach, we are also able to guarantee that the global asymptotic stability of the proposed distributed algorithm is robust to small perturbations. This chapter concludes with simulation results of a distributed assignment problem.

We would like to remark that our contributions in this chapter are also relevant to the field of switched and hybrid systems [44, 47, 57]. To the authors' knowledge, this thesis is the first to consider event-triggered implementations of state-dependent switched dynamical systems. As already eluded to in this introduction, a unique challenge that must be overcome is the fact that the use of outdated state information may cause the system to miss a mode switch and in turn may affect the overall stability and performance.

4.1 Problem Statement

Let us formally state the problem of distributed linear programming with event-triggered communication using the notion of hybrid systems described in Chapter 2.

Problem Statement 4.1.1. (Distributed linear programming with event-triggered communication). *Design a hybrid system that, for each* $i \in \{1, \dots, n\}$*, takes the form,*

$$\dot{x}_i = g_i(\hat{x}), \quad \text{if } (x, \hat{x}) \notin \mathcal{T}_i, \tag{4.1a}$$

$$\hat{x}_i^+ = x_i, \quad \text{if } (x, \hat{x}) \in \mathcal{T}_i, \tag{4.1b}$$

where $g_i : \mathbb{R}^n \to \mathbb{R}$ *is agent* i*'s flow map and* $\mathcal{T}_i \subseteq \mathbb{R}^n \times \mathbb{R}^n$ *is agent* i*'s trigger set, which determines when* i *should broadcast its state, such that*

(i) g_i *is computable by agent* i *and the inclusion* $(x, \hat{x}) \in \mathcal{T}_i$ *is detectable by agent* i *using local information and communication, and*

(ii) the aggregate of the agents' states converge to a solution of (2.8).

The interpretation of Problem 4.1.1 is as follows. Equation (4.1a) models the fact that agent $i \in \{1, \dots, n\}$ uses the last broadcast states from neighboring agents and itself to compute the continuous-time flow g_i governing the evolution of its state. In-between two consecutive broadcasts of agent i (i.e., while flowing), there is no dynamics for its last broadcast state $\hat{x}_i$. Formally, $\dot{\hat{x}}_i = 0$ if $(x, \hat{x}) \notin \mathcal{T}_i$. For this reason, the state evolution is quite easy to compute since it changes according to a constant rate during continuous flow. Our use of the term "continuous-time flow" is motivated by the fact that we model the event-triggered design in the hybrid system framework. Moreover, viewing the dynamics (4.1a) as a continuous-time flow will aid our analysis in subsequent sections. Equation (4.1b) models the broadcast procedure. The condition $(x, \hat{x}) \in \mathcal{T}_i$ is a state-based trigger used by agent i to determine when to broadcast its current state x_i to its neighbors. Since communication is instantaneous, $x_i^+ = x_i$ if $(x, \hat{x}) \in \mathcal{T}_i$. The dynamical coupling between different agents is through the broadcast states in $\hat{x}$ only. Note that an agent cannot pre-determine the time of its next state broadcast because it cannot predict if or when it will receive a broadcast from a neighbor. For this reason, we call the strategy outlined in Problem 4.1.1 *event-triggered* as opposed to self-triggered.

4.2 Re-design of the continuous-time algorithm

Our initial aim was to simply use the continuous-time dynamics (3.8) to define the agents' state dynamics between broadcasts (i.e., the g_i in (4.1a), albeit with auxiliary states z). However, it turns out that this dynamics is not directly amenable to an event-triggered implementation (we make this point clearer in Remark 4.3.4 later). Nevertheless a slight modification of that algorithm deems it suitable. In this section we motivate and introduce that modification. We will also state and prove the analogs of Propositions 3.1.1 and 3.2.4 for this new dynamics as they will be necessary for our technical analysis later.

The dynamics we introduce here are derived in a parallel way as done in Chapter 3 but for a regularized linear program instead. More specifically, consider the following quadratic regularization of (2.8),

$$\min \quad \gamma c^T x + \frac{1}{2} x^T x \tag{4.2a}$$

$$\text{s.t.} \quad Ax = b, \quad x \geq 0, \tag{4.2b}$$

where $\gamma \geq 0$. The following result reveals that this regularization is exact for suitable values of γ. The result is a modification of [60, Theorem 1] for linear programs in standard form rather than in inequality form.

Lemma 4.2.1. (Exact regularization). *There exists* $\gamma_{\min} > 0$ *such that, for*

$\gamma \geq \gamma_{\min}$, *the solution to the regularization* (4.2) *is a solution to the linear program* (2.8).

Proof. We use the fact that a point $x_* \in \mathbb{R}^n$ (resp. $z_* \in \mathbb{R}^m$) is a solution to (2.8) (resp. the dual of (2.8)) if and only if it satisfies the KKT conditions for (2.8),

$$c + A^T z_* \geq 0, \quad Ax_* = b, \quad x_* \geq 0, \quad (c + A^T z_*)^T x_* = 0. \tag{4.3}$$

We also consider the optimization problem

$$\min \quad \frac{1}{2} x^T x \tag{4.4a}$$

$$\text{s.t.} \quad Ax = b, \quad c^T x = p, \quad x \geq 0, \tag{4.4b}$$

where p is the optimal value of (2.8). Note that the solution to the above problem is a solution to (2.8) by construction of the constraints. Likewise, $(\bar{x}, \bar{z}, \bar{w}) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}$ are primal-dual solutions to (4.4) if and only if they satisfy the KKT conditions for (4.4)

$$\bar{x} + A^T \bar{z} + c\bar{w} \geq 0, \quad A\bar{x} = b, \quad c^T \bar{x} = p, \quad \bar{x} \geq 0, \quad (\bar{x} + A^T \bar{z} + c\bar{w})^T \bar{x} = 0. \tag{4.5}$$

Since $\bar{x}$ is a solution to (2.8), without loss of generality, we suppose that $x_* = \bar{x}$. We consider the cases when (i) $\bar{w} = 0$, (ii) $\bar{w} > 0$, and (iii) $\bar{w} < 0$. In case (i), combining (4.3) and (4.5), one can obtain for any $\gamma \geq 0$,

$$\gamma c + x_* + A^T(\gamma z_* + \bar{z}) \geq 0, \quad Ax_* = b, \quad x_* \geq 0,$$
$$(\gamma c + x_* + A^T(\gamma z_* + \bar{z}))^T x_* = 0.$$

The above conditions reveal that $(x_*, \gamma z_* + \bar{z})$ satisfy the KKT conditions for (4.2). Thus, x_* (which is a solution to (2.8)) is the solution to (4.2) and this would complete the proof. Next, consider case (ii). If $\gamma = \gamma_{\min} := \bar{w} > 0$, the conditions (4.5) can be manipulated to give

$$\gamma c + x_* + A^T \bar{z} \geq 0, \quad Ax_* = b, \quad x_* \geq 0, \quad (\gamma c + x_* + A^T \bar{z})^T x_* = 0.$$

This means that $(x_*, \bar{z})$ satisfy the KKT conditions for (4.2). Thus, x_* (which is a solution to (2.8)) is the solution to (4.2) and this would complete the proof.

Now, for any $\gamma \geq \gamma_{\min}$, there exists an $\eta \geq 0$ such that $\gamma = \gamma_{\min} + \eta = \bar{w} + \eta$. Combining (4.3) and (4.5), one can obtain

$$\gamma c + x_* + A^T(\eta z_* + \bar{z}) \geq 0, \quad Ax_* = b, \quad x_* \geq 0,$$
$$(\gamma c + x_* + A^T(\eta z_* + \bar{z})^T x_* = 0.$$

This means that $(x_*, \eta z_* + \bar{z})$ satisfy the KKT conditions for (4.2). Thus, x_* (which is a solution to (2.8)) is the solution to (4.2) and this would complete the proof. Case (iii) can be considered analogously to case (ii) with $\gamma_{\min} := -\bar{w}$. This completes the proof. □

The actual value of $\gamma_{\min}$ in Lemma 4.2.1 is somewhat generic in the following sense: if one replaces c in (4.2) by $\bar{\gamma} c$ for some $\bar{\gamma} > 0$, then the regularization is exact for $\gamma \geq \frac{\gamma_{\min}}{\bar{\gamma}}$. Therefore, to ease notation, we make the following standing assumption,

$$\textbf{SA \#1:} \quad \gamma_{\min} \leq 1.$$

This justifies our focus on the case $\gamma = 1$. Our next result establishes the correspondence between the solution of (4.2) and the saddle points of an associated augmented Lagrangian function. This result can be seen as the analog of Proposition 3.1.1.

Lemma 4.2.2. (Solutions of (4.2) **as saddle points).** *For $K \geq 0$, consider the augmented Lagrangian function $L^K_{QR} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ associated to the quadratically regularized optimization problem* (4.2) *with $\gamma = 1$,*

$$L^K_{QR}(x, z) = c^T x + \frac{1}{2} x^T x + \frac{1}{2}(Ax - b)^T(Ax - b) + z^T(Ax - b) + K\mathbb{1}^T \max\{0, -x\}.$$

Then, L^K_{QR} is convex in x and concave in z. Let $x_* \in \mathbb{R}^n$ (resp. $z_* \in \mathbb{R}^m$) be the solution to (4.2) *(resp. a solution to the dual of* (4.2)*). Then, for $K > \|c + x_* + A^T z_*\|_\infty$, the following holds,*

(i) (x_*, z_*) is a saddle point of L^K_{QR},

(ii) if $(\bar{x}, \bar{z})$ is a saddle point of L^K_{QR}, then $\bar{x} = x_*$ is the solution of (4.2).

Proof. For any $x \in \mathbb{R}^n$,

$$\begin{aligned}
L_{QR}^K(x, z_*) &= c^T x + \frac{1}{2}x^T x + \frac{1}{2}(Ax-b)^T(Ax-b) + z_*^T(Ax-b) + K\mathbb{1}^T \max\{0, -x\}, \\
&\geq c^T x + \frac{1}{2}x^T x + z_*^T(Ax-b) + \|c + x_* + A^T z_*\|_\infty x, \\
&\geq c^T x + \frac{1}{2}x^T x + z_*^T(Ax-b) - (c + x_* + A^T z_*)^T x, \\
&\geq c^T x + \frac{1}{2}x^T x + z_*^T A(x - x_*) - (c + x_* + A^T z_*)^T (x - x_*), \\
&\geq c^T x + \frac{1}{2}x^T x - (c + x_*)^T (x - x_*), \\
&\geq c^T x_* + \frac{1}{2}(x - x_*)^T (x - x_*) + \frac{1}{2}x_*^T x_*, \\
&\geq c^T x_* + \frac{1}{2}x_*^T x_* = L_{QR}^K(x_*, z_*). \qquad (4.6)
\end{aligned}$$

For any z, it is easy to see that $L_{QR}^K(x_*, z) = L_{QR}^K(x_*, z_*)$. Thus (x_*, z_*) is a saddle point of L_{QR}^K.

Let us now prove (ii). As a necessary condition for $(\bar{x}, \bar{z})$ to be a saddle point of L_{QR}^K, it must be that $\partial_z L_{QR}^K(\bar{x}, \bar{z}) = A\bar{x} - b = 0$ as well as $L_{QR}^K(x_*, \bar{x}) \geq L_{QR}^K(\bar{x}, \bar{z})$ which means that

$$c^T x_* + \frac{1}{2}x_*^T x_* \geq c^T \bar{x} + \frac{1}{2}\bar{x}^T \bar{x} + K\mathbb{1}^T \max\{0, -\bar{x}\}. \qquad (4.7)$$

If $\bar{x} \geq 0$ then $c^T x_* + \frac{1}{2}x_*^T x_* \geq c^T \bar{x} + \frac{1}{2}\bar{x}^T \bar{x}$ and thus $\bar{x}$ would be a solution to (4.2). Consider then that $\bar{x} \ngeq 0$. Then,

$$\begin{aligned}
c^T \bar{x} + \frac{1}{2}\bar{x}^T \bar{x} &= c^T x_* + \frac{1}{2}x_*^T x_* + (c + x_*)^T (\bar{x} - x_*) + \frac{1}{2}(\bar{x} - x_*)^T (\bar{x} - x_*), \\
&\geq c^T x_* + \frac{1}{2}x_*^T x_* + (c + x_*)^T (\bar{x} - x_*), \\
&\geq c^T x_* + \frac{1}{2}x_*^T x_* - z_*^T A(\bar{x} - x_*) + (c + x_* + A^T z_*)^T (\bar{x} - x_*), \\
&\geq c^T x_* + \frac{1}{2}x_*^T x_* - z_*^T (A\bar{x} - b) + (c + x_* + A^T z_*)^T \bar{x}, \\
&\geq c^T x_* + \frac{1}{2}x_*^T x_* - \|c + x_* + A^T z_*\|_\infty \max\{0, -\bar{x}\}, \\
&> c^T x_* + \frac{1}{2}x_*^T x_* - K\mathbb{1}^T \max\{0, -\bar{x}\},
\end{aligned}$$

contradicting (4.7). Thus, $\bar{x} \geq 0$ and must be the solution to (4.2). ☐

Then, as in Chapter 3, a sensible strategy to find a solution of (2.8) is via the saddle point dynamics,

$$\dot{x} \in -\partial_x L^K_{QR}(x,z), \tag{4.8a}$$

$$\dot{z} = \partial_z L^K_{QR}(x,z). \tag{4.8b}$$

This dynamics is well-defined since L^K_{QR} is a locally Lipschitz function. In a similar fashion to our approach in Chapter 3, we consider instead the discontinuous dynamics

$$\dot{x}_i = \begin{cases} f^{QR}_i(x,z), & \text{if } x_i > 0, \\ \max\{0, f^{QR}_i(x,z)\}, & \text{if } x_i = 0, \end{cases} \tag{4.9a}$$

$$\dot{z} = Ax - b, \tag{4.9b}$$

for each $i \in \{1,\dots,n\}$ where $f^{QR} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is defined as

$$f^{QR}(x,z) := -(A^T z + c) - A^T(Ax - b) - x = f(x,z) - x.$$

Note the $-x$ term, appearing due to the quadratic regularization, appears in $f^{QR}(x,z)$ when compared with $f(x,z)$.

The next result shows that the above discontinuous dynamics simply represent a certain selection of elements from the set-valued saddle-point dynamics (4.8).

Lemma 4.2.3. (Generalized gradients of the Lagrangian). *Given a compact set $X \times Z \subset \mathbb{R}^n_{\geq 0} \times \mathbb{R}^m$, let*

$$K_*(X \times Z) := \max_{(x,z) \in X \times Z} \|f^{QR}(x,z)\|_\infty.$$

Then, if $K \geq K_*(X \times Z)$, for each $(x,z) \in X \times Z$, $\partial_z L^K_{QR}(x,z) = \{Ax - b\}$ and there exists $a \in -\partial_x L^K_{QR}(x,z) \subset \mathbb{R}^n$ such that, for each $i \in \{1,\dots,n\}$,

$$a_i = \begin{cases} f^{QR}_i(x,z), & \text{if } x_i > 0, \\ \max\{0, f^{QR}_i(x,z)\}, & \text{if } x_i = 0. \end{cases}$$

Proof. Let $(x, z) \in \mathbb{R}^n \times \mathbb{R}^m$. It is straightforward to see that $\partial_z L^K_{QR}(x, z) = \{Ax - b\}$ for any K. Next, note that for any $a \in -\partial_x L^K_{QR}(x, z)$, we have

$$-a - (A^T z + c + x) - A^T(Ax - b) \in K\partial \max\{0, -x\}, \tag{4.10}$$

or, equivalently, $-a + f^{QR}(x, z) \in K\partial \max\{0, -x\}$. For any $i \in \{1, \dots, n\}$ such that $x_i > 0$, the corresponding set in the right-hand side of (4.10) is the singleton 0 and therefore $a_i = f^{QR}_i(x, z)$. On the other hand, if $x_i = 0$, then

$$-a_i + f^{QR}_i(x, z) \in [-K, 0].$$

If $f^{QR}_i(x, z) \geq 0$, the choice $a_i = f^{QR}_i(x, z)$ satisfies the equation. Conversely, if $f^{QR}_i(x, z) < 0$, then $a_i = 0$ satisfies the equation for all $K \geq K_*(X \times Z)$ by definition of $K_*(X \times Z)$. This completes the proof. $\square$

The above result can be understood as the analog of Proposition 3.2.4 but with a slightly different emphasis to fit our application in this chapter. Specifically, this result reveals that, on a compact set, the trajectories of the dynamics (4.9) are trajectories of (4.8). In other words, any bounded trajectory of (4.9) is also a trajectory of (4.8).

The dynamics (4.9) is precisely the one we use to design an event-triggered implementation for in the next sections. Besides the standard considerations in designing an event-triggered implementation (such as ensuring convergence and preventing arbitrarily fast broadcasting), we face several unique challenges including the fact that the equilibria of (4.9) are not known a priori as well as having to account for the switched nature of the dynamics. However, we note that the distributed implementation of (4.9) is equivalent to that of (3.8). In the context of this chapter, it is useful to think of each z_ℓ as the state of a virtual agent with identifier $n + \ell$.

We conclude this section by stating a final standing assumptions that will simplify the technical exposition in subsequent sections. Namely, without loss of generality, we assume

$$\textbf{SA \#2:} \quad \rho(A^T A) \leq 1.$$

Two reasons justify the generality of **SA #2**: the results are easily extensible to the case $\rho(A^T A) > 1$ and there exists a $O(m)$ distributed algorithm, that we explain briefly here, to ensure the assumption holds. It can be viewed as a simple pre-processing algorithm based on max-consensus:

For each row a_ℓ of the matrix A, the virtual agent $n+\ell$ can compute the ℓ^{th} row of $A^T A$. Then, this agent stores the following estimate of the spectral radius,

$$\hat{\rho}_{n+\ell} = (A^T A)_{(\ell,\ell)} + \sum_{i \in \{1,\dots,n\} \setminus \{\ell\}} |(A^T A)_{(\ell,i)}|.$$

The virtual agents use these estimates as an initial point in the standard max-consensus algorithm [33]. In $O(m)$ steps, the max-consensus converges to a point $\rho_* \geq \rho(A^T A)$, where the inequality is a consequence of the Gershgorin Circle Theorem [48, Corollary 6.1.5]. Then, each virtual agent scales its corresponding row of A and entry of b by $1/\rho_*$, and communicates this new data to its neighbors. The resulting linear program is $\min\{c^T x : \tilde{A}x = \tilde{b},\ x \geq 0\}$, with $\tilde{A} = A/\rho_*$ and $\tilde{b} = b/\rho_*$. Both the solutions and optimal value of the new linear program are the same as the original linear program and, by construction, $\rho(\tilde{A}^T \tilde{A}) \leq 1$.

4.3 Algorithm design with centralized event-triggered communication

Here, we build on the discussion of Section 4.2 to address the main objective of this chapter as outlined in Problem 4.1.1. Our starting point is the distributed continuous-time algorithm (4.9), which requires continuous-time communication. Our approach is divided in two steps because of the complexity of the challenges (e.g., asymptotic convergence, asynchronism, and persistence of solutions) involved in going from continuous-time to opportunistic discrete-time communication. First, we design a centralized event-triggered scheme that the network can use to determine in an opportunistic way when information should be updated. Then, in the next section, we build on this development to design a distributed event-triggered communication scheme that individual agents can employ to determine when to share information with their neighbors.

The problem we solve in this section can be formally stated as follows.

Problem Statement 4.3.1. (Linear programming with centralized event-triggered communication). *Identify a set* $\mathcal{T}^c \subseteq \mathbb{R}^n_{\geq 0} \times \mathbb{R}^m \times \mathbb{R}^n_{\geq 0} \times \mathbb{R}^m$ *such that the hybrid system, for* $i \in \{1, \ldots, n\}$*, of the form,*

$$\dot{x}_i = \begin{cases} f_i^{QR}(\hat{x}, \hat{z}), & \hat{x}_i > 0, \\ \max\{0, f_i^{QR}(\hat{x}, \hat{z})\}, & \hat{x}_i = 0, \end{cases} \tag{4.11a}$$

$$\dot{z} = A\hat{x} - b, \tag{4.11b}$$

if $(x, z, \hat{x}, \hat{z}) \notin \mathcal{T}^c$ *and*

$$(\hat{x}^+, \hat{z}^+) = (x, z), \tag{4.11c}$$

if $(x, z, \hat{x}, \hat{z}) \in \mathcal{T}^c$*, makes the aggregate* $x \in \mathbb{R}^n$ *of the real agents' states converge to a solution of the linear program* (2.8)*.*

We refer to the set $\mathcal{T}^c$ in Problem 4.3.1 as the *centralized trigger set*. Note that, in this centralized formulation of the problem, we do not require individual agents, but rather the network as a whole, to be able to detect whether $(x, z, \hat{x}, \hat{z}) \in \mathcal{T}^c$. In addition, when this condition is enabled, state broadcasts among agents are performed synchronously, as described by (4.11c). Our strategy to design $\mathcal{T}^c$ is to first identify a candidate Lyapunov function and study its evolution along the trajectories of (4.11). We then synthesize $\mathcal{T}^c$ based on the requirement that our Lyapunov function decreases along the trajectories of (4.11) and conclude with a result showing that the desired convergence properties are attained.

Before we introduce the candidate Lyapunov function, we present an alternative representation of (4.11a)-(4.11b) that will be useful in our analysis later. Given $(\hat{x}, \hat{z}) \in \mathbb{R}_{\geq 0} \times \mathbb{R}^m$, let $\sigma(\hat{x}, \hat{z})$ be the set of agents i for which $\dot{x}_i = f_i^{QR}(\hat{x}, \hat{z})$ in (4.11a). Formally,

$$\sigma(\hat{x}, \hat{z}) = \big\{ i \in \{1, \ldots, n\} : f_i^{QR}(\hat{x}, \hat{z}) \geq 0 \text{ or } \hat{x}_i > 0 \big\}.$$

Next, let $I_{\sigma(\hat{x}, \hat{z})} \in \mathbb{R}^{n \times n}$ be defined by

$$(I_{\sigma(\hat{x}, \hat{z})})_{i,j} = \begin{cases} 0, \text{ if } i \neq j \text{ or } i \notin \sigma(\hat{x}, \hat{z}), \\ 1, \text{ otherwise.} \end{cases}$$

Note that this matrix is an identity-like matrix with a zero (i,i)-element if $i \notin \sigma(\hat{x},\hat{z})$. The matrix $I_{\sigma(\hat{x},\hat{z})}$ has the following properties,

$$I_{\sigma(\hat{x},\hat{z})} \succeq 0, \quad I_{\sigma(\hat{x},\hat{z})} = I_{\sigma(\hat{x},\hat{z})}^T, \quad I_{\sigma(\hat{x},\hat{z})}^2 = I_{\sigma(\hat{x},\hat{z})}, \quad \rho(I_{\sigma(\hat{x},\hat{z})}) \leq 1.$$

Then, a compact representation of (4.11a)-(4.11b) is

$$(\dot{x},\dot{z}) = F(\hat{x},\hat{z}) := (I_{\sigma(\hat{x},\hat{z})} f^{QR}(\hat{x},\hat{z}), A\hat{x} - b),$$

where $F = (F_x, F_z) : \mathbb{R}^n_{\geq 0} \times \mathbb{R}^m \to \mathbb{R}^n \times \mathbb{R}^m$.

4.3.1 Candidate Lyapunov function and its evolution

Now let us define and analyze the candidate Lyapunov function that we use to design the trigger set $\mathcal{T}^c$. The overall Lyapunov function is the sum of 2 separate functions V_1 and V_2, that we introduce next. To define $V_1 : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_{\geq 0}$, fix $\mathcal{K} > \|c + x_* + A^T z_*\|_\infty$ where x_* (resp. z_*) is the solution to (4.2) (resp. any solution of the dual of (4.2)) and let $(\bar{x},\bar{z})$ be a saddle-point of $L_{QR}^{\mathcal{K}}$. Then

$$V_1(x,z) = \frac{1}{2}(x-\bar{x})^T(x-\bar{x}) + \frac{1}{2}(z-\bar{z})^T(z-\bar{z}).$$

Note that $V_1 \geq 0$ is smooth with compact sublevel sets. Next, $V_2 : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ is given by

$$V_2(x,z) = \frac{1}{2} f^{QR}(x,z)^T I_{\sigma(x,z)} f^{QR}(x,z) + \frac{1}{2}(Ax-b)^T(Ax-b).$$

Note that $V_2 \geq 0$ but, due to the state-dependent matrix $I_{\sigma(x,z)}$, is only piecewise smooth. In this sense V_2 can be viewed as a collection of multiple (smooth) Lyapunov functions, each defined on a neighborhood where σ is constant. Also, $V_2^{-1}(0)$ is, by definition, the set of saddle-points of $L_{QR}^{\mathcal{K}}$ (cf. Lemma 4.2.3). It turns out that the negative terms in the Lie derivative of V_1 alone are insufficient to ensure that V_1 is always decreasing given any practically implementable trigger design (by *practically implementable* we mean a trigger design that does not demand arbitrarily fast state broadcasting). The analogous statement regarding V_2 is also true which is why we consider instead a candidate Lyapunov function $V : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ that is their sum

$$V(x,z) = (V_1 + V_2)(x,z).$$

We would like to establish an upper bound on $\mathcal{L}_F V$ in terms of the state errors in x and z. However, because V_2 is discontinuous, we are unable to apply the definition of Lie-derivative as revealed in Chapter 2. Rather, we use the following definition: The Lie derivative of V along F at (x, z) is

$$\mathcal{L}_F V(x,z) := \lim_{\alpha \to 0} \frac{V((x,z) + \alpha F(x,z)) - V(x,z)}{\alpha}. \tag{4.12}$$

We say that $\mathcal{L}_F V(x,z)$ exists when the limit in (4.12) exist. When V is differentiable at (x,z), then we recover the standard Lie derivative, $\mathcal{L}_F V(x,z) = \nabla V(x,z)^T F(x,z)$. We may now state the result.

Proposition 4.3.2. (Evolution of V). *Let $X \times Z \subseteq \mathbb{R}^n_{\geq 0} \times \mathbb{R}^m$ be compact and suppose that $(x, z, \hat{x}, \hat{z}) \in X \times Z \times X \times Z$ is such that $\sigma(\hat{x}, \hat{z}) \subseteq \sigma(x, z)$ and*

$$\sigma(x,z) = \lim_{\alpha \to 0} \sigma(x + \alpha F_x(\hat{x}, \hat{z}), z + \alpha F_z(\hat{x}, \hat{z})). \tag{4.13}$$

Then $\mathcal{L}_F V(x,z)$ exists and

$$\begin{aligned}\mathcal{L}_F V(x,z) \leq& -\frac{1}{2} f^{QR}(\hat{x},\hat{z})^T I_{\sigma(\hat{x},\hat{z})} f^{QR}(\hat{x},\hat{z}) - \frac{1}{4}(A\hat{x} - b)^T (A\hat{x} - b) + 40 e_x^T e_x \\ &+ 20 e_z^T e_z + 15 f^{QR}(x,z)^T I_{\sigma(x,z)\setminus\sigma(\hat{x},\hat{z})} f^{QR}(x,z),\end{aligned} \tag{4.14}$$

where $e_x = x - \hat{x}$ and $e_z = z - \hat{z}$.

Proof. For convenience, we use the shorthand notation $p = \sigma(x,z)$ and $\hat{p} = \sigma(\hat{x}, \hat{z})$. Consider first V_1, which is differentiable and thus $\mathcal{L}_F V_1(x,z)$ exists,

$$\begin{aligned}\mathcal{L}_F V_1(x,z) &= (x - \bar{x})^T I_{\hat{p}} f^{QR}(\hat{x},\hat{z}) + (z - \bar{z})^T (A\hat{x} - b), \\ &= (\hat{x} - \bar{x})^T I_{\hat{p}} f^{QR}(\hat{x},\hat{z}) + (\hat{z} - \bar{z})^T (A\hat{x} - b) + e_x^T I_{\hat{p}} f^{QR}(\hat{x},\hat{z}) \\ &\quad + e_z^T (A\hat{x} - b).\end{aligned} \tag{4.15}$$

Since $X \times Z$ is compact, without loss of generality assume that $\mathcal{K} \geq K_*(X \times Z)$ so that $-I_{\hat{p}} f^{QR}(\hat{x},\hat{z}) \in \partial_x L^{\mathcal{K}}_{QR}(\hat{x},\hat{z})$, cf. Lemma 4.2.3. This, together with the fact that $L^{\mathcal{K}}_{QR}$ is convex in its first argument, implies

$$L^{\mathcal{K}}_{QR}(\hat{x},\hat{z}) \leq L^{\mathcal{K}}_{QR}(\bar{x},\hat{z}) - (\hat{x} - \bar{x})^T I_{\hat{p}} f^{QR}(\hat{x},\hat{z}).$$

Since $L_{QR}^{\mathcal{K}}$ is linear in z, we can write

$$L_{QR}^{\mathcal{K}}(\hat{x}, \hat{z}) = L_{QR}^{\mathcal{K}}(\hat{x}, \bar{z}) + (\hat{z} - \bar{z})^T (A\hat{x} - b).$$

Substituting these expressions into (4.15), we get

$$\begin{aligned}
\mathcal{L}_F V_1(x, z) &\leq L_{QR}^{\mathcal{K}}(\bar{x}, \hat{z}) - L_{QR}^{\mathcal{K}}(\hat{x}, \bar{z}) + e_x^T I_{\hat{p}} f^{QR}(\hat{x}, \hat{z}) + e_z^T (A\hat{x} - b), \\
&\leq c^T \bar{x} + \frac{1}{2}\bar{x}^T \bar{x} - c^T \hat{x} - \frac{1}{2}\sum_{i=1}^{n} \hat{x}^T \hat{x} - \bar{z}^T (A\hat{x} - b) - \mathcal{K}\mathbb{1}^T \max\{0, -\hat{x}\} \\
&\quad - \frac{1}{2}(A\hat{x} - b)^T (A\hat{x} - b) + e_x^T I_{\hat{p}} f^{QR}(\hat{x}, \hat{z}) + e_z^T (A\hat{x} - b).
\end{aligned}$$

From the analysis in the proof of Lemma 4.2.2, inequality (4.6) showed that

$$\begin{aligned}
&c^T \bar{x} + \frac{1}{2}\bar{x}^T \bar{x} \\
&\quad \leq c^T \hat{x} + \frac{1}{2}\hat{x}^T \hat{x} + \bar{z}^T (A\hat{x} - b) + \frac{1}{2}(A\hat{x} - b)^T (A\hat{x} - b) + \mathcal{K}\mathbb{1}^T \max\{0, -\hat{x}\},
\end{aligned}$$

where we use the fact that $\bar{x}$ is also a solution to (4.2), cf. Lemma 4.2.2. Therefore,

$$\begin{aligned}
\mathcal{L}_F V_1(x, z) &\leq -\frac{1}{2}(A\hat{x} - b)^T (A\hat{x} - b) + e_x^T I_{\hat{p}} f^{QR}(\hat{x}, \hat{z}) + e_z^T (A\hat{x} - b), \\
&\leq -\frac{1}{2}(A\hat{x} - b)^T (A\hat{x} - b) + \frac{\kappa}{2}(A\hat{x} - b)^T (A\hat{x} - b) \\
&\quad + \frac{\kappa}{2} f^{QR}(\hat{x}, \hat{z})^T I_{\hat{p}} f^{QR}(\hat{x}, \hat{z}) + \frac{1}{2\kappa} e_x^T e_x + \frac{1}{2\kappa} e_z^T e_z, \qquad (4.16)
\end{aligned}$$

where we have used Lemma 2.1.1. Next, let us consider V_2. We begin by showing that (4.13) is sufficient for $\mathcal{L}_F V_2(x, z)$ to exist. Since σ is a discrete set of indices, for the limit in (4.13) to exist, there must exist an $\bar{\alpha} > 0$ such that

$$\sigma(x, z) = \sigma(x + \alpha F_x(\hat{x}, \hat{z}), z + \alpha F_z(\hat{x}, \hat{z})),$$

for all $\alpha \in [0, \bar{\alpha}]$. This means that one can substitute $I_{\sigma(x,z)}$ for $I_{\sigma(x+\alpha F_x(\hat{x},\hat{z}), z+\alpha F_z(\hat{x},\hat{z}))}$ in the definition of the Lie derivative (4.12). Since $I_{\sigma(x,z)}$ is constant with respect to α, it is straightforward to see that

$$\begin{aligned}
\mathcal{L}_F V_2(x, z) &= \frac{1}{2}\nabla(f^{QR}(x, z)^T I_p f^{QR}(x, z))^T F(\hat{x}, \hat{z}) \\
&\quad + \frac{1}{2}\nabla((Ax - b)^T (Ax - b))^T F(\hat{x}, \hat{z}).
\end{aligned}$$

Thus,

$$\begin{aligned}\mathcal{L}_F V_2(x,z) &= f^{QR}(x,z)^T I_p(D_x f^{QR}(x,z)F_x(\hat{x},\hat{z}) + D_z f^{QR}(x,z)F_z(\hat{x},\hat{z}))\\ &\quad + (Ax-b)^T A F_x(\hat{x},\hat{z}),\\ &= -f^{QR}(x,z)^T I_p(A^TA+I)I_{\hat{p}} f^{QR}(\hat{x},\hat{z}) - f^{QR}(x,z)^T I_p A^T(A\hat{x}-b)\\ &\quad + (Ax-b)^T A I_{\hat{p}} f^{QR}(\hat{x},\hat{z}). \end{aligned} \tag{4.17}$$

Due to the assumption that $\hat{p} \subseteq p$, we can write $I_p = I_{\hat{p}} + I_{p\backslash\hat{p}}$. Also, $f^{QR}(x,z)$ can be written equivalently in terms of the errors e_x, e_z as

$$f^{QR}(x,z) = f^{QR}(\hat{x},\hat{z}) - A^T e_z - e_x - A^TAe_x.$$

Likewise, $Ax - b = A\hat{x} - b + Ae_x$. Substituting these quantities into (4.17),

$$\begin{aligned}\mathcal{L}_F V_2(x,z) &= -(f^{QR}(\hat{x},\hat{z}) - A^Te_z - e_x - A^TAe_x)^T I_{\hat{p}}(A^TA+I)I_{\hat{p}} f^{QR}(\hat{x},\hat{z})\\ &\quad - f^{QR}(x,z)^T I_{p\backslash\hat{p}}(A^TA+I)I_{\hat{p}} f^{QR}(\hat{x},\hat{z})\\ &\quad - (f^{QR}(\hat{x},\hat{z}) - A^Te_z - e_x - A^TAe_x)^T I_{\hat{p}} A^T(A\hat{x}-b)\\ &\quad - f^{QR}(x,z)^T I_{p\backslash\hat{p}} A^T(A\hat{x}-b)\\ &\quad + (A\hat{x}-b+Ae_x)^T A I_{\hat{p}} f^{QR}(\hat{x},\hat{z}). \end{aligned} \tag{4.18}$$

We now derive upper bounds for a few terms in (4.18). For example,

$$\begin{aligned} e_z^T A I_{\hat{p}} A^T A I_{\hat{p}} f^{QR}(\hat{x},\hat{z}) &\le \frac{1}{2\kappa} e_z^T e_z + \frac{\kappa}{2} f^{QR}(\hat{x},\hat{z})^T I_{\hat{p}} A^T A I_{\hat{p}} A^T A I_{\hat{p}} A^T A I_{\hat{p}} f^{QR}(\hat{x},\hat{z}),\\ &\le \frac{1}{2\kappa} e_z^T e_z + \frac{\kappa}{2} f^{QR}(\hat{x},\hat{z})^T I_{\hat{p}} f^{QR}(\hat{x},\hat{z}), \end{aligned}$$

where we have used Lemma 2.1.1 and Theorem 2.1.2 along with the facts that $\rho(A^TA) = \rho(AA^T) \le 1$ and $\rho(I_{\hat{p}}) \le 1$. Likewise,

$$\begin{aligned} e_x^T I_{\hat{p}} A^T A I_{\hat{p}} f^{QR}(\hat{x},\hat{z}) &\le \frac{1}{2\kappa} e_x^T e_x + \frac{\kappa}{2} f^{QR}(\hat{x},\hat{z})^T I_{\hat{p}} A^T A I_{\hat{p}} A^T A I_{\hat{p}} f^{QR}(\hat{x},\hat{z}),\\ &\le \frac{1}{2\kappa} e_x^T e_x + \frac{\kappa}{2} f^{QR}(\hat{x},\hat{z})^T I_{\hat{p}} f^{QR}(\hat{x},\hat{z}), \end{aligned}$$

and

$$\begin{aligned} f^{QR}(x,z)^T I_{p\backslash\hat{p}} A^T(A\hat{x}-b) &\le \frac{1}{2\kappa} f^{QR}(x,z)^T I_{p\backslash\hat{p}} f^{QR}(x,z)\\ &\quad + \frac{\kappa}{2}(A\hat{x}-b)^T AA^T(A\hat{x}-b),\\ &\le \frac{1}{2\kappa} f^{QR}(x,z)^T I_{p\backslash\hat{p}} f^{QR}(x,z) + \frac{\kappa}{2}(A\hat{x}-b)^T(A\hat{x}-b). \end{aligned}$$

Repeatedly bounding every term in (4.18) in an analogous way (which we omit for the sake of space and presentation) and adding the bound (4.16), we attain the following inequality

$$\mathcal{L}_F V(x,z) \leq -(1-5\kappa) f^{QR}(\hat{x},\hat{z})^T I_{\hat{p}} f^{QR}(\hat{x},\hat{z}) - \frac{1}{2}(1-5\kappa)(A\hat{x}-b)^T(A\hat{x}-b) + \frac{3}{2\kappa} f^{QR}(x,z)^T I_{p\backslash\hat{p}} f^{QR}(x,z) + \frac{4}{\kappa} e_x^T e_x + \frac{2}{\kappa} e_z^T e_z.$$

Equation (4.14) follows by choosing $\kappa = \frac{1}{10}$, completing the proof. □

The reason why we have only considered the case $\sigma(\hat{x},\hat{z}) \subseteq \sigma(x,z)$ (and not the more general case of $\sigma(\hat{x},\hat{z}) \neq \sigma(x,z)$) when deriving the bound (4.14) in Proposition 4.3.2 is the following: our distributed trigger design later (specifically, the trigger sets $\mathcal{T}_i^0$ introduced in Section 4.4) ensures that $\sigma(\hat{x},\hat{z}) \subseteq \sigma(x,z)$ always. For this reason, we need not know how V evolves in the more general case.

4.3.2 Centralized trigger set design and convergence analysis

Here, we use our knowledge of the evolution of the function V, cf. Proposition 4.3.2, to design the centralized trigger set $\mathcal{T}^c$. Our approach is to incrementally design subsets of $\mathcal{T}^c$ and then combine them at the end to define $\mathcal{T}^c$. The main observation that we base our design on is the following: The first two terms in the right-hand-side of (4.14) are negative and thus desirable and the rest are positive. However, following a state broadcast, the positive terms become zero. This motivates our first trigger set that should belong to $\mathcal{T}^c$,

$$\mathcal{T}^{c,e} := \{(x,z,\hat{x},\hat{z}) \in (\mathbb{R}^n_{\geq 0} \times \mathbb{R}^m)^2 : \ A\hat{x}-b \neq 0 \text{ or } I_{\sigma(\hat{x},\hat{z})} f^{QR}(\hat{x},\hat{z}) \neq 0, \text{ and } \frac{1}{8}(A\hat{x}-b)^T(A\hat{x}-b) + \frac{1}{4} f^{QR}(\hat{x},\hat{z})^T I_{\sigma(\hat{x},\hat{z})} f^{QR}(\hat{x},\hat{z}) \leq 20 e_z^T e_z + 40 e_x^T e_x\}. \tag{4.19}$$

The numbers $\frac{1}{8}$ and $\frac{1}{4}$ in the inequalities that define $\mathcal{T}^{c,e}$ are design choices that we have made to ease the presentation. Any other choice in $(0,1)$ is also possible, with the appropriate modifications in the ensuing exposition. Note that, when both $A\hat{x} - b = I_{\sigma(\hat{x},\hat{z})} f^{QR}(\hat{x}, \hat{z}) = 0$, no state broadcasts are required since the system is at a (desired) equilibrium.

Likewise, after a state broadcast, $\sigma(x, z) = \sigma(\hat{x}, \hat{z})$ and $I_{\sigma(x,z)\setminus\sigma(\hat{x},\hat{z})} = 0$ which means that the last term in (4.14) is also zero. For this reason, define

$$\mathcal{T}^{c,\sigma} := \{(x, z, \hat{x}, \hat{z}) \in (\mathbb{R}^n_{\geq 0} \times \mathbb{R}^m)^2 : \sigma(x, z) \neq \sigma(\hat{x}, \hat{z})\}, \tag{4.20}$$

which prescribes a state broadcast when the mode σ changes.

We require one final trigger for the following reason. While the set $\mathbb{R}^n_{\geq 0} \times \mathbb{R}^m$ is invariant under the continuous-time dynamics (4.9), this does not hold any more in the event-triggered case because agents use outdated state information. To preserve the invariance of this set, we define

$$\mathcal{T}^{c,0} := \{(x, z, \hat{x}, \hat{z}) \in (\mathbb{R}^n_{\geq 0} \times \mathbb{R}^m)^2 : \exists i \in \{1, \dots, n\} \text{ s.t. } \hat{x}_i > 0,\ x_i = 0\}. \tag{4.21}$$

If this trigger is activated by some agent i's state becoming zero, then it is easy to see from the definition of the dynamics (4.11a) that $\dot{x}_i \geq 0$ after the state broadcast and thus x_i remains non-negative. Finally, the overall centralized trigger set is

$$\mathcal{T}^c := \mathcal{T}^{c,e} \cup \mathcal{T}^{c,\sigma} \cup \mathcal{T}^{c,0}. \tag{4.22}$$

The following result characterizes the convergence properties of (4.11) under the centralized event-triggered communication scheme specified by (4.22).

Theorem 4.3.3. (Convergence of the centralized event-triggered design). *If ψ is a persistently flowing solution of* (4.11) *with $\mathcal{T}^c$ defined as in* (4.22)*, then there exists a point $(x_*, z') \in \mathcal{X} \times \mathbb{R}^m$ such that,*

$$\psi(t, j) \to (x_*, z', x_*, z') \quad \text{as} \quad t + j \longrightarrow \infty, \quad (t, j) \in \text{dom}(\psi).$$

Proof. Let $(t, j) \mapsto \psi(t, j) = (x(t, j), z(t, j), \hat{x}(t, j), \hat{z}(t, j))$. We begin the proof by showing that V is non-increasing along ψ. To this end, it suffices to prove that (a) $\mathcal{L}_F V(x(t, j), z(t, j)) \leq 0$ when ψ is flowing and $\mathcal{L}_F V(x(t, j), z(t, j))$ exists, (b) $V(x(t, j), z(t, j)) \leq \lim_{\tau \to t^-} V(x(t, j), z(t, j))$ when $\psi(t, j)$ is flowing but $\mathcal{L}_F V(x(t, j), z(t, j))$ does not exist, and (c) $V(x(t, j + 1), z(t, j + 1)) \leq V(x(t, j), z(t, j))$ when ψ is jumping.

We begin with (a) and consider $(t,j) \in \mathrm{dom}(\psi)$ for which ψ is flowing. Then, the interval $I^j := \{t : (t,j) \in \mathrm{dom}(\psi)\}$ has non-empty interior and $t \in \mathrm{int}(I^j)$. This means that $\psi(t,j) \notin \mathcal{T}^c$ and, in particular, $\sigma(x(t,j),z(t,j)) = \sigma(\hat{x}(t,j),\hat{z}(t,j))$ by construction of $\mathcal{T}^{c,\sigma}$. Also, let $X \times Z$ be a compact set such that $\psi(t,j) \in X \times Z \times X \times Z$. Therefore, the conditions of Proposition 4.3.2 are satisfied and $\mathcal{L}_F V(x(\tau,j),z(\tau,j))$ exists for all $\tau \in \mathrm{int}(I^j)$. Using (4.14), it holds that

$$\begin{aligned}\mathcal{L}_F V(x(t,j),z(t,j)) \leq & -\frac{1}{8}(A\hat{x}(t,j)-b)^T(A\hat{x}(t,j)-b) \\ & -\frac{1}{4}f^{QR}(\hat{x}(t,j),\hat{z}(t,j))^T I_{\sigma(\hat{x}(t,j),\hat{z}(t,j))} f^{QR}(\hat{x}(t,j),\hat{z}(t,j)),\end{aligned}$$

where we have used (i) the fact that, since $\sigma(x(t,j),z(t,j)) = \sigma(\hat{x}(t,j),\hat{z}(t,j))$, the last quantity in (4.14) is zero and (ii) the bound on $20e_x^Te_x + 40e_z^Te_z$ in the definition of $\mathcal{T}^{c,e}$. Clearly, in this case, $\mathcal{L}_F V(x(t,j),z(t,j)) \leq 0$ when $\psi(t,j) \in X \times Z \times X \times Z$.

Next, consider (b). Since V_1 is smooth, $\mathcal{L}_F V_1(x(t,j),z(t,j))$ exists, however, when $V_2(x(t,j),z(t,j))$ is discontinuous, $\mathcal{L}_F V(x(t,j),z(t,j))$ does not. This happen at any $(t,j) \in \mathrm{dom}(\psi)$ for which (i) I^j (as defined previously) has non-empty interior and (ii) $\sigma(x(t,j),z(t,j)) \neq \lim_{\tau\to t^-}\sigma(x(\tau,j),z(\tau,j))$ (cf. Proposition 4.3.2). Note that condition (i) ensures that the limit in condition (ii) is well-defined. For purposes of presentation, define the sets

$$\begin{aligned}\mathcal{S}_+ &:= \sigma(x(t,j),z(t,j)) \setminus \lim_{\tau\to t^-}\sigma(x(\tau,j),z(\tau,j)), \\ \mathcal{S}_- &:= \lim_{\tau\to t^-}\sigma(x(\tau,j),z(\tau,j)) \setminus \sigma(x(t,j),z(t,j)).\end{aligned}$$

Note that one of $\mathcal{S}_+, \mathcal{S}_-$ may be empty. We can write

$$\begin{aligned}V_2(x(t,j),z(t,j)) = & \lim_{\tau\to\tau^-} V_2(x(\tau,j),z(\tau,j)) \\ & + \frac{1}{2}\sum_{i\in\mathcal{S}_+} f_i^{QR}(x(t,j),z(t,j))^2 - \frac{1}{2}\sum_{i\in\mathcal{S}_-} f_i^{QR}(x(t,j),z(t,j))^2.\end{aligned}$$

For each $i \in \mathcal{S}_+$, it must be that $f_i^{QR}(x(t,j),z(t,j)) = 0$ since $f_i^{QR}(\hat{x}(t,j),\hat{z}(t,j)) < 0$ and f^{QR}, x, z are continuous. Moreover, the last term in the right-hand-side of the above expression is non-positive. Thus, $V_2(x(t,j),z(t,j)) \leq \lim_{\tau\to t^-} V_2(x(\tau,j),z(\tau,j))$.

Next, when ψ is jumping, as is case (c), $V(x(t,j),z(t,j)) = V(x(t,j+1),z(t,j+1)$ because $(x(t,j+1),z(t,j+1)) = (x(t,j),z(t,j))$ according to (4.11c).

To summarize, $V(x(t,j),z(t,j))$ is non-increasing when $\psi(t,j) \in X \times Z \times X \times Z$. Without loss of generality, we choose $X \times Z = V^{-1}(\leq c)$, where $c = V(x(0,0),z(0,0))$. $X \times Z$ is compact because the sublevel sets of V_1 are compact, and $X \times Z \times X \times Z$ is invariant so as not to contradict $V(x(t,j),z(t,j))$ being non-increasing on $X \times Z \times X \times Z$. Thus, $V(x(t,j),z(t,j))$ is non-increasing at all times and ψ is bounded.

Now we establish the convergence property of (4.11). First note that, in this preliminary design, ψ being persistently flowing implies also that the Lie derivative of V along F exists for τ_P time on those intervals of persistent flow. This is because ψ flowing implies that σ is constant and thus the Lie derivative of V along F exists (cf. (4.13)). There are two possible characterizations of persistently flowing ψ as given in Chapter 2. Consider (PFi). By the boundedness of ψ just established, it must be that for all $t \geq t_J$

$$
\begin{aligned}
0 &= \dot{x}(t,j) = I_{\sigma(\hat{x}(t_J,J),\hat{z}(t_J,J))} f^{QR}(\hat{x}(t_J,J)),\hat{z}(t_J,J)), \\
0 &= \dot{z}(t,j) = A\hat{x}(t_J,J) - b.
\end{aligned}
$$

By Lemma 4.2.3 this means that $(\hat{x}(t_J,J),\hat{z}(t_J,J))$ is a saddle-point of $L^{\mathcal{K}}_{QR}$ (without loss of generality, we assume that $\mathcal{K} \geq K_*(X \times Z)$). Applying Lemma 4.2.2 reveals that $\hat{x}(t_J,J) \in \mathcal{X}$. Since $\hat{x}(t,j)$ is a sampled version of $x(t_J,J)$ it is clear that $x(t_J,J) \in \mathcal{X}$ as well, and since their dynamics are stationary in finite time, they converge to a point, completing the proof.

Consider then (PFii), the second characterization of persistently flowing. We have established that $\{V(x(t_{j_k},j_k),z(t_{j_k},j_k))\}_{k=0}^{\infty}$ is non-increasing. Since it is also bounded from below by 0, by the monotone convergence theorem there exists a $V_* \in [0,c]$ such that $\lim_{k\to\infty} V(x(t_{j_k},j_k),z(t_{j_k},j_k)) = V_*$. Thus

$$
V(x(t_{j_k},j_k),z(t_{j_k},j_k)) - V(x(t_{j_{k+1}},j_{k+1}),z(t_{j_{k+1}},j_{k+1})) \to 0.
$$

Let $\delta > 0$ and consider $\kappa \in \mathbb{N}$ such that

$$
V(x(t_{j_k},j_k),z(t_{j_k},j_k)) - V(x(t_{j_{k+1}},j_{k+1}),z(t_{j_{k+1}},j_{k+1})) < \delta,
$$

for all $k \geq \kappa$. By the bound established on $\mathcal{L}_F V(x(t, j_k), z(t, j_k))$, which exists for all $(t, j_k) \in ([t_{j_k}, t_{j_k} + \tau_P), j_k)$, it holds that,

$$
\begin{aligned}
&V(x(t_{j_{k+1}}, j_{k+1}), z(t_{j_{k+1}}, j_{k+1})) \\
&\quad \leq V(x(t_{j_k}, j_k), z(t_{j_k}, j_k)) - \frac{1}{8}(A\hat{x}(t_{j_k}, j_k) - b)^T (A\hat{x}(t_{j_k}, j_k) - b)\tau_P \\
&\quad\quad - \frac{1}{4} f^{QR}(\hat{x}(t_{j_k}, j_k), \hat{z}(t_{j_k}, j_k))^T I_{\sigma(\hat{x}(t_{j_k}, j_k), \hat{z}(t_{j_k}, j_k))} f^{QR}(\hat{x}(t_{j_k}, j_k), \hat{z}(t_{j_k}, j_k))\tau_P.
\end{aligned}
$$

Therefore, $V(x(t_{j_k}, j_k), z(t_{j_k}, j_k)) - V(x(t_{j_{k+1}}, j_{k+1}), z(t_{j_{k+1}}, j_{k+1})) < \delta$ for all $k \geq \kappa$ implies that

$$
\begin{aligned}
f^{QR}(\hat{x}(t_{j_k}, j_k), \hat{z}(t_{j_k}, j_k))^T I_{\sigma(\hat{x}(t_{j_k}, j_k), \hat{z}(t_{j_k}, j_k))} f^{QR}(\hat{x}(t_{j_k}, j_k), \hat{z}(t_{j_k}, j_k)) &\leq 4\delta\tau_P, \\
(A\hat{x}(t_{j_k}, j_k) - b)^T (A\hat{x}(t_{j_k}, j_k) - b) &\leq 8\delta\tau_P,
\end{aligned}
$$

for all $k \geq \kappa$. Since τ_P is a uniform constant and $\delta > 0$ can be taken arbitrarily small, we deduce

$$
I_{\sigma(\hat{x}(t_{j_k}, j_k), \hat{z}(t_{j_k}, j_k))} f^{QR}(\hat{x}(t_{j_k}, j_k), \hat{z}(t_{j_k}, j_k)) \to 0 \quad \text{and} \quad A\hat{x}(t_{j_k}, j_k) - b \to 0,
$$

as $k \to \infty$. By Lemma 4.2.3, this means that $(\hat{x}(t_{j_k}, j_k), \hat{z}(t_{j_k}, j_k))$ converges to the set of saddle-points of $L^{\mathcal{K}}_{QR}$. The same argument holds for $x(t_{j_k}, j_k)$ since $\hat{x}(t_{j_k}, j_k)$ is a sampled version of that state.

Finally, we establish the convergence to a point. By the Bolzano-Weierstrass Theorem, there exists a subsequence $\{j_{k_\ell}\}$ such that $(x(t_{j_{k_\ell}}, j_{k_\ell}), z(t_{j_{k_\ell}}, j_{k_\ell}))$ converges to a saddle-point $(\bar{x}', \bar{z}')$ of $L^{\mathcal{K}}_{QR}$. Fix $\delta' > 0$ and let ℓ_* be such that

$$
\|(x(t_{j_{k_\ell}}, j_{k_\ell}), z(t_{j_{k_\ell}}, j_{k_\ell})) - (\bar{x}', \bar{z}')\| < \delta',
$$

for all $\ell \geq \ell_*$. Consider the function $W = W_1 + V_2$ where

$$
W_1(x, z) = \frac{1}{2}(x - \bar{x}')^T (x - \bar{x}') + \frac{1}{2}(z - \bar{z}')^T (z - \bar{z}').
$$

Let $c' = W(x(t_{j_{k_{\ell_*}}}, j_{k_{\ell_*}}), z(t_{j_{k_{\ell_*}}}, j_{k_{\ell_*}}))$ and $X' \times Z' = W^{-1}(\leq c')$. Repeating the previous analysis, but for W instead of V, we deduce that $X' \times Z' \times X' \times Z'$ is invariant. Consequently, $\|(x(t, j), z(t, j)) - (\bar{x}', \bar{z}')\| < \delta'$ for all $(t, j) \in \mathrm{dom}(\psi)$ such that $t + j \geq t_{j_{k_{\ell_*}}} + j_{k_{\ell_*}}$. Since $\delta' > 0$ is arbitrary, it holds that

$$
\psi(t, j) \to (\bar{x}', \bar{z}', \bar{x}', \bar{z}') \quad \text{as} \quad t + j \longrightarrow \infty, \quad (t, j) \in \mathrm{dom}(\psi).
$$

Since $(\bar{x}', \bar{z}')$ is a saddle-point of $L_{QR}^{\mathcal{K}}$ and, without loss of generality $\mathcal{K} \geq K_*(X \times Z)$, applying Lemma 4.2.2 reveals that $\bar{x}' \in \mathcal{X}$, which completes the proof. □

Remark 4.3.4. (Motivation for quadratic regularization of linear program). Here we revisit the claim made in Section 4.2 that using the saddle-point dynamics (3.8) derived for the original linear program (2.8) would not be amenable to an event-triggered implementation. If we were to follow the same design methodology for such dynamics, we would find that the bound on the Lie derivative of V would resemble (4.14), but without the non-positive term $-f^{QR}(\hat{x}, \hat{z})^T I_{\sigma(\hat{x},\hat{z})} f^{QR}(\hat{x}, \hat{z})$. Following the same methodology to identify the trigger set, one would then use the trigger

$$\frac{1}{8}(A\hat{x} - b)^T(A\hat{x} - b) \leq 20 e_z^T e_z + 40 e_x^T e_x,$$

to define $\mathcal{T}^{c,e}$ and ensure that the function V does not increase. However, this trigger may easily result in continuous-time communication: consider a scenario where $A\hat{x} - b = 0$, but the state x is still evolving. Then the trigger would require continuous-time broadcasting of x to ensure that e_x remains zero. •

4.4 Algorithm design with distributed event-triggered communication

In this section, we provide a distributed solution to Problem 4.1.1, e.g., a coordination algorithm to solve linear programs requiring only communication at discrete instants of time triggered by criteria that agents can evaluate with local information. Our strategy to accomplish this is to investigate to what extent the centralized triggers identified in Section 4.3.2 can be implemented in a distributed way. In turn, making these triggers distributed poses the additional challenge of dealing with the asynchronism in the state broadcasts across different agents, which raises the possibility of non-persistency in the solutions. We deal with both issues in our forthcoming discussion and establish the convergence of our distributed design.

4.4.1 Distributed trigger set design

Here, we design distributed triggers that individual agents can evaluate with the local information available to them to guarantee the monotonically decreasing evolution of the candidate Lyapunov function V. Our design methodology builds on the centralized trigger sets $\mathcal{T}^{c,e}$, $\mathcal{T}^{c,\sigma}$, and $\mathcal{T}^{c,0}$ of Section 4.3.2. As a technical detail, the distributed algorithm that results from this section has an extended state which, for ease of notation, we denote by

$$\begin{aligned}&\xi = (x, z, s, q, r, \hat{x}, \hat{z}),\\ &\subseteq \Xi := \mathbb{R}^n_{\geq 0} \times \mathbb{R}^m \times \mathbb{R}^n_{\geq 0} \times \{0,1\}^{(n+m)\times n} \times \mathbb{R}^{n+m} \times \mathbb{R}^n_{\geq 0} \times \mathbb{R}^m.\end{aligned}$$

The meaning and dynamics of states s, q, and r will be revealed as they become necessary in our development.

We start by showing how the inequality that defines whether the network state belongs to the set $\mathcal{T}^{c,e}$ in (4.19) can be distributed across the group of agents. Given $\mu_1, \dots, \mu_{n+m} > 0$, consider the following trigger set for each agent,

$$\mathcal{T}^e_i := \begin{cases} \{\xi \in \Xi : f^{QR}_i(\hat{x}, \hat{z}) \neq 0 \text{ and } (e_x)^2_i \geq \mu_i f^{QR}_i(\hat{x}, \hat{z})^2\}, & \text{if } i \leq n, \\ \{\xi \in \Xi : a^T_{i-n}\hat{x} - b_{i-n} \neq 0 \text{ and } (e_z)^2_{i-n} \geq \mu_i (a^T_{i-n}\hat{x} - b_{i-n})^2\}, & \text{if } i \geq n+1. \end{cases}$$

If each $\mu_i \leq \frac{1}{160}$ and $(x, z, \hat{x}, \hat{z})$ is such that the inequalities defining each $\mathcal{T}^e_i$ do not hold, then it is clear that $(x, z, \hat{x}, \hat{z}) \notin \mathcal{T}^{c,e}$. To ensure convergence of the resulting algorithm, we later characterize the specific ranges for the design parameters $\{\mu_i\}_{i=1}^{n+m}$.

Next, we show how the inclusion of the network state in the triggered set $\mathcal{T}^{c,0}$ defined in (4.21) can be easily evaluated by individual agents with partial information. In fact, for each $i \in \{1, \dots, n\}$, define the set

$$\mathcal{T}^0_i := \{\xi \in \Xi : \hat{x}_i > 0 \text{ but } x_i = 0\}.$$

Clearly, $(x, z, \hat{x}, \hat{z}) \in \mathcal{T}^{c,0}$ if and only if there is $i \in \{1, \dots, n\}$ such that $\xi \in \mathcal{T}^0_i$.

The triggered set $\mathcal{T}^{c,\sigma}$ defined in (4.20) presents a greater challenge from a distributed computation viewpoint. The problem is that, in the absence of fully up-to-date information from its neighbors, an agent will fail to detect the mode switches that characterize the definition of this set. The specific scenario we refer to is the following: assume agent $i \in \{1,\dots,n\}$ has $x_i = 0$ and the information available to it confirms that its state should remain constant, i.e., with $f_i^{QR}(\hat{x},\hat{z}) < 0$. If the condition $f_i^{QR}(x,z) \geq 0$ becomes true as the network state evolves, this fact is undetectable by i with its outdated information. In such a case, $i \notin \sigma(\hat{x},\hat{z})$ but $i \in \sigma(x,z)$, meaning that the equality $\sigma(x,z) = \sigma(\hat{x},\hat{z})$ defining the trigger set $\mathcal{T}^{c,\sigma}$ would not be not enforced. To deal with this issue, we first need to understand the effect that a mismatch in the modes has on the evolution of the candidate Lyapunov function V. We address this in the following result.

Proposition 4.4.1. (Bound on evolution of candidate Lyapunov function under mode mismatch). *Suppose that $(\hat{x},\hat{z}) \in \mathbb{R}^n_{\geq 0} \times \mathbb{R}^m$ is such that $i \notin \sigma(\hat{x},\hat{z})$ for some $i \in \{1,\dots,n\}$ and let $t \mapsto (x(t),z(t))$ be the solution to*

$$(\dot{x},\dot{z}) = F(\hat{x},\hat{z}),$$

starting from $(\hat{x},\hat{z})$. Let $T > 0$ be the minimum time such that $i \in \sigma(x(T),z(T))$. Then, for any $\nu > 0$, and all t such that $t - T < \frac{\nu}{2\sqrt{2}}$, the following holds,

$$f_i^{QR}(x(t),z(t))^2 \leq \nu^2 f^{QR}(\hat{x},\hat{z})^T I_{\sigma(\hat{x},\hat{z})\cap\mathcal{N}_i^x} f^{QR}(\hat{x},\hat{z}) + \nu^2 (A\hat{x}-b)^T I_{\mathcal{N}_i^z}(A\hat{x}-b),$$

where $\mathcal{N}_i^x := \mathcal{N}_i \cap \{1,\dots,n\}$ and $\mathcal{N}_i^z := \mathcal{N}_i \cap \{n+1,\dots,n+m\}$.

Proof. We use the shorthand notation $\hat{p} = \sigma(\hat{x},\hat{z})$ and $p(t) = \sigma(x(t),z(t))$. Since $i \notin \hat{p}$, it must be that $\hat{x}_i = 0$ and $f_i^{QR}(\hat{x},\hat{z}) < 0$. Moreover, if $i \in p(T)$, it must be, by continuity of $t \mapsto (x(t),z(t))$ and $(x,z) \mapsto f^{QR}(x,z)$, that $f_i^{QR}(x(T),z(T)) = 0$. Let us compute the Taylor expansion of $t \mapsto f_i^{QR}(x(t),z(t))$ using $t = T$ as the initial point. For technical reasons, we actually consider the equivalent mapping $t \mapsto I_{\{i\}} f^{QR}(x(t),z(t))$ instead,

$$\begin{aligned}
I_{\{i\}} f^{QR}(x(t),z(t)) &= I_{\{i\}} f^{QR}(x(T),z(T)) + D_x I_{\{i\}} f^{QR}(x(T),z(T))^T F_x(\hat{x},\hat{z})(t-T) \\
&\quad + D_z I_{\{i\}} f^{QR}(x(T),z(T))^T F_z(\hat{x},\hat{z})(t-T), \\
&= I_{\{i\}}(A^T A + I) I_{\hat{p}} f^{QR}(\hat{x},\hat{z})(t-T) + I_{\{i\}} A^T (A\hat{x}-b)(t-T),
\end{aligned}$$

where the equality holds because the higher order terms are zero. Thus,

$$
\begin{aligned}
f^{QR}(x(t), z(t))^T I_{\{i\}} f^{QR}(x(t), z(t)) \\
&= f^{QR}(\hat{x}, \hat{z})^T I_{\hat{p}} (A^T A + I) I_{\{i\}} (A^T A + I) I_{\hat{p}} f^{QR}(\hat{x}, \hat{z})(t - T)^2 \\
&\quad + 2 f^{QR}(\hat{x}, \hat{z})^T I_{\hat{p}} (A^T A + I) I_{\{i\}} A^T (A\hat{x} - b)(t - T)^2 \\
&\quad + (A\hat{x} - b)^T A I_{\{i\}} A^T (A\hat{x} - b)(t - T)^2.
\end{aligned}
$$

Using Lemma 2.1.1 with $\kappa = \frac{1}{2}$ and exploiting the consistency between the matrix A and the neighbors of i, we obtain

$$
\begin{aligned}
& f^{QR}(x(t), z(t))^T I_{\{i\}} f^{QR}(x(t), z(t)) \\
&\quad \leq 2 f^{QR}(\hat{x}, \hat{z})^T I_{\hat{p}} (A^T A + I) I_{\{i\}} (A^T A + I) I_{\hat{p}} f^{QR}(\hat{x}, \hat{z})(t - T)^2 \\
&\qquad + 2 (A\hat{x} - b)^T A I_{\{i\}} A^T (A\hat{x} - b)(t - T)^2, \\
&\quad \leq 2 f^{QR}(\hat{x}, \hat{z})^T I_{\hat{p} \cap \mathcal{N}_i^x} (A^T A + I)^2 I_{\hat{p} \cap \mathcal{N}_i^x} f^{QR}(\hat{x}, \hat{z})(t - T)^2 \\
&\qquad + 2 (A\hat{x} - b)^T I_{\mathcal{N}_i^z} A A^T I_{\mathcal{N}_i^z} (A\hat{x} - b)(t - T)^2, \\
&\quad \leq 8 f^{QR}(\hat{x}, \hat{z})^T I_{\hat{p} \cap \mathcal{N}_i^x} f^{QR}(\hat{x}, \hat{z})(t - T)^2 + 2 (A\hat{x} - b)^T I_{\mathcal{N}_i^z} (A\hat{x} - b)(t - T)^2, \\
&\quad \leq 8 (t - T)^2 (f^{QR}(\hat{x}, \hat{z})^T I_{\hat{p} \cap \mathcal{N}_i^x} f^{QR}(\hat{x}, \hat{z}) + (A\hat{x} - b)^T I_{\mathcal{N}_i^z} (A\hat{x} - b)).
\end{aligned}
$$

Using the bound $t - T \leq \frac{\nu}{2\sqrt{2}}$ in the statement of the result completes the proof. □

The importance of Proposition 4.4.1 comes from the following observation: given the upper bound on the evolution of the candidate Lyapunov function V obtained in Proposition 4.3.2, one can appropriately choose the value of ν so that the negative terms in (4.14) can compensate for the presence of the last term due to a mode mismatch of finite time length. This observation motivates the introduction of the following trigger sets, which cause neighbors to send synchronized broadcasts periodically to an agent if its state remains at zero. First, if an agent i's state is zero and it has not received a synchronized broadcast from its neighbors for τ_i time (here, $\tau_i > 0$ is a design parameter), it triggers a broadcast to notify its neighbors that it requires new states. This behavior is captured by the trigger set

$$
\mathcal{T}_i^{\text{request}} := \begin{cases} \{\xi \in \Xi : x_i = 0 \text{ and } s_i \geq \tau_i\}, & \text{if } i \leq n, \\ \emptyset, & \text{if } i \geq n+1. \end{cases}
$$

where we use the state s_i to denote the time since i has last sent a broadcast. On the receiving end, if i receives a broadcast request from a neighbor j, then it should also broadcast immediately,

$$\mathcal{T}_i^{\text{send}} := \{\xi \in \Xi : \exists j \in \mathcal{N}_i^x \text{ s.t. } q_{i,j} = 1\},$$

where $q_{i,j} \in \{0, 1\}$ is a state with $q_{i,j} = 1$ indicating that j has requested a broadcast from i.

Our last component of the distributed trigger design addresses the problem posed by the asynchronism in state broadcasts. In fact, given that agents determine autonomously when to communicate with their neighbors, this may cause non-persistence in the resulting network evolution. As an example, consider a scenario where successive state broadcasts by one agent cause another neighboring agent to generate a state broadcast of its own after increasingly smaller time intervals, and vice versa. To address this problem, we provide a final component to the design of the distributed trigger set as follows,

$$\mathcal{T}_i^{\text{synch}} := \{\xi \in \Xi : 0 \leq r_i \leq r_i^{\min}\},$$

where r_i represents the time elapsed between when agent i received a state broadcast from a neighbor and i's last broadcast. We use $r_i = -1$ to indicate that i has not received a broadcast from a neighbor since its own last state broadcast. The threshold $r_i^{\min} > 0$ is a design parameter (smaller values result in less frequent updates). Intuitively, this trigger means that if an agent broadcasts its state and in turn receives a state broadcast from a neighbor faster than some tolerated rate, the agent broadcasts its state immediately again. The effect of this trigger is that, if broadcasts start occurring too frequently in the network, neighboring agents' broadcasts synchronize. This emergent behavior is described in more depth in the proof of Theorem 4.4.3 later.

Finally, the overall distributed trigger set for each $i \in \{1, \ldots, n+m\}$ is,

$$\mathcal{T}_i := \mathcal{T}_i^e \cup \mathcal{T}_i^0 \cup \mathcal{T}_i^{\text{request}} \cup \mathcal{T}_i^{\text{send}} \cup \mathcal{T}_i^{\text{synch}}. \tag{4.23}$$

4.4.2 Distributed algorithm and convergence analysis

We now state the distributed algorithm and its convergence properties which are the main contributions of this chapter.

Algorithm 4.4.2. (Distributed linear programming with event-triggered communication). *For each agent $i \in \{1, \dots, n+m\}$, if $\xi \notin \mathcal{T}_i$ then*

$$\dot{x}_i = \begin{cases} f_i^{QR}(\hat{x}, \hat{z}), & \text{if } \hat{x}_i > 0, \\ \max\{0, f_i^{QR}(\hat{x}, \hat{z})\}, & \text{if } \hat{x}_i = 0, \end{cases} \qquad \text{if } i \leq n \tag{4.24a}$$

$$\dot{z}_{i-n} = a_{i-n}^T \hat{x} - b_{i-n}, \qquad \text{if } i \geq n+1 \tag{4.24b}$$

$$\dot{s}_i = \begin{cases} 1, & \text{if } s_i < \tau_i, \\ 0, & \text{if } s_i \geq \tau_i, \end{cases} \qquad \text{for all } i \tag{4.24c}$$

and, if $\xi \in \mathcal{T}_i$, then

$$\hat{x}_i^+ = x_i \qquad \text{if } i \leq n \tag{4.24d}$$

$$\hat{z}_{i-n}^+ = z_{i-n}, \qquad \text{if } i \geq n+1 \tag{4.24e}$$

$$(s_i^+, r_i^+, r_j^+) = (0, -1, s_j), \qquad \text{for all } i \text{ and all } j \in \mathcal{N}_i \tag{4.24f}$$

$$q_{j,i}^+ = 1, \qquad \text{if } \xi \in \mathcal{T}_i^{\text{request}} \text{ and for all } j \in \mathcal{N}_i \tag{4.24g}$$

$$q_{i,j}^+ = 0, \qquad \text{if } \xi \in \mathcal{T}_i^{\text{send}} \text{ and for all } j \in \mathcal{N}_i^x \tag{4.24h}$$

The entire network state is given by $\xi \in \Xi$. However, the local state of an individual agent $i \in \{1, \dots, n\}$ consists of $x_i, \hat{x}_i, s_i, r_i$, and $\cup_{j \in \mathcal{N}_i^x}\{q_{i,j}\}$. Likewise, the local state of agent $i \in \{n+1, \dots, n+m\}$ consists of $z_{i-n}, \hat{z}_{i-n}, s_i, r_i$, and $\cup_{j \in \mathcal{N}_i^x}\{q_{i,j}\}$. These latter agents may be implemented as virtual agents. Then, recalling the assumptions on local information outlined in Section 3.2.2, it is straightforward to see that the coordination algorithm (4.24) can be implemented by the agents in a distributed way. We are now ready to state our main convergence result.

Theorem 4.4.3. (Distributed triggers - convergence and persistently flowing solutions). *For each $i \in \{1, \dots, n+m\}$, let $0 < \mu_i \leq \frac{1}{160}$ and*

$$0 < r_i^{\min} \leq \tau_i < \frac{1}{\sqrt{960|\mathcal{N}_i| \max_{j \in \mathcal{N}_i} |\mathcal{N}_j|}}.$$

Let ψ be a solution of (4.24)*, with each set $\mathcal{T}_i$ defined by* (4.23)*. Then,*

(i) if ψ is persistently flowing, there exists a point $(x_*, z) \in \mathcal{X} \times \mathbb{R}^m$ such that,

$$(x(t,j), z(t,j)) \to (x_*, z') \quad as \quad t + j \longrightarrow \infty, \quad (t,j) \in \mathrm{dom}(\psi),$$

(ii) if there exists $\delta_P > 0$ such that, for any time $(t', j') \in \mathrm{dom}(\psi)$ where $\psi(t', j') \in \mathcal{T}_i^0$ for some $i \in \{1, \dots, n\}$, it holds that $\psi(t,j) \notin \mathcal{T}_i^0$ for all $(t,j) \in ((t', t' + \delta_P] \times \mathbb{N}) \cap \mathrm{dom}(\psi)$, the solution ψ is persistently flowing.

Proof. The proof of the convergence result in (i) follows closely the argument we employed to establish Theorem 4.3.3. One key difference is that the intervals on which ψ flows do not necessarily correspond to the intervals on which $\mathcal{L}_F V$ exists. This is because the value of σ may change even though ψ still flows. However, since the dynamics $(\dot{x}, \dot{z}) = F(\hat{x}, \hat{z})$ is constant on periods of flow it is easy to see that there can be at most n agents added to σ in any given period of flow. This means that, if ψ is persistently flowing according to the characterization (PFii), the Lie derivative $\mathcal{L}_F V$ exists persistently often for periods of length τ_P/n (since σ must be constant for an interval of length at least τ_P/n persistently often). Thus, let us consider a time (t,j) such that $(t,j) \in (t_j, t_j + \tau_p/n) \times \{j\} \subset \mathrm{dom}(\psi)$ and $\mathcal{L}_F V(x(t,j), z(t,j))$ exists. Note that, if ψ is persistently flowing according to the characterization (PFi), we may take $\tau_P = \infty$ and the following analysis holds. To ease notation, denote $p(t,j) = \sigma(x(t,j), z(t,j))$ and $\hat{p}(t,j) = \sigma(\hat{x}(t,j), \hat{z}(t,j))$. Then, following the exposition in the proof of Theorem 4.3.3, one can see that, due to trigger sets $\mathcal{T}_i^e$ and $\mathcal{T}_i^0$ and the conditions on μ_i,

$$\begin{aligned}
&\mathcal{L}_F V(x(t,j), z(t,j)) \\
&\quad \le -\frac{1}{4} f^{QR}(\hat{x}(t,j), \hat{z}(t,j))^T I_{\hat{p}(t,j)} f^{QR}(\hat{x}(t,j), \hat{z}(t,j)) \\
&\qquad - \frac{1}{8}(A\hat{x}(t,j) - b)^T (A\hat{x}(t,j) - b) \\
&\qquad + 15 f^{QR}(x(t,j), z(t,j)) I_{p(t,j)\setminus\hat{p}(t,j)} f^{QR}(x(t,j), z(t,j)).
\end{aligned} \tag{4.25}$$

We focus on the last term, which is the only positive one. If $i \in p(t,j) \setminus \hat{p}(t,j)$, then it must be that $\hat{x}_i = 0$ and thus i is receiving state broadcasts from its neighbors every τ_i seconds by design of $\mathcal{T}_i^{\text{request}}$ and $\{\mathcal{T}_j^{\text{send}}\}_{j\in\mathcal{N}_i}$. Therefore, the maximum amount of time that any i remains in $p(t,j)\setminus\hat{p}(t,j)$ is τ_i seconds. Since each $\tau_i < \frac{1}{\sqrt{960|\mathcal{N}_i|\max_{j\in\mathcal{N}_i}|\mathcal{N}_j|}}$, we apply Proposition 4.4.1 using $\nu < \frac{2\sqrt{2}}{\sqrt{960|\mathcal{N}_i|\max_{j\in\mathcal{N}_i}|\mathcal{N}_j|}}$ to obtain

$$
\begin{aligned}
&f_i^{QR}(x(t,j),z(t,j))^2 \\
&\quad< \frac{1}{120|\mathcal{N}_i|\max_{j\in\mathcal{N}_i}|\mathcal{N}_j|}\big(f^{QR}(\hat{x}(t,j),\hat{z}(t,j))^T I_{\hat{p}(t,j)\cap\mathcal{N}_i^x} f^{QR}(\hat{x}(t,j),\hat{z}(t,j)) \\
&\qquad\qquad + (A\hat{x}(t,j)-b)^T I_{\mathcal{N}_i^z}(A\hat{x}(t,j)-b)\big).
\end{aligned}
$$

From the above bound it is clear that

$$
\begin{aligned}
&f^{QR}(x(t,j),z(t,j)) I_{p(t,j)\setminus\hat{p}(t,j)} f^{QR}(x(t,j),z(t,j)) \\
&< \tfrac{1}{120}\sum_{i\in p(t,j)\setminus\hat{p}(t,j)} \tfrac{1}{|\mathcal{N}_i|\max_{j\in\mathcal{N}_i}|\mathcal{N}_j|}\big(f^{QR}(\hat{x}(t,j),\hat{z}(t,j))^T I_{\hat{p}(t,j)\cap\mathcal{N}_i^x} f^{QR}(\hat{x}(t,j),\hat{z}(t,j)) \\
&\qquad\qquad + (A\hat{x}(t,j)-b)^T I_{\mathcal{N}_i^z}(A\hat{x}(t,j)-b)\big), \\
&< \tfrac{1}{120}\sum_{i\in\{1,\dots,n\}} \tfrac{1}{|\mathcal{N}_i|}\sum_{k\in\mathcal{N}_j}\tfrac{1}{|\mathcal{N}_j|}\big(f^{QR}(\hat{x}(t,j),\hat{z}(t,j))^T I_{\hat{p}(t,j)} f^{QR}(\hat{x}(t,j),\hat{z}(t,j)) \\
&\qquad\qquad + (A\hat{x}(t,j)-b)^T(A\hat{x}(t,j)-b)\big), \\
&< \tfrac{1}{120}\big(f^{QR}(\hat{x}(t,j),\hat{z}(t,j))^T I_{\hat{p}(t,j)} f^{QR}(\hat{x}(t,j),\hat{z}(t,j)) + (A\hat{x}(t,j)-b)^T(A\hat{x}(t,j)-b)\big),
\end{aligned}
$$

which, when combined with (4.25), reveals that there exists some $\varepsilon > 0$ such that

$$
\begin{aligned}
&\mathcal{L}_F V(x(t,j),z(t,j)) \\
&\le -\varepsilon\big(f^{QR}(\hat{x}(t,j),\hat{z}(t,j))^T I_{\hat{p}(t,j)} f^{QR}(\hat{x}(t,j),\hat{z}(t,j)) + (A\hat{x}(t,j)-b)^T(A\hat{x}(t,j)-b)\big).
\end{aligned}
$$

The remainder of the convergence proof now follows in the same way as the proof of Theorem 4.3.3.

Next, we prove (ii) by contradiction. Suppose that the conditions in (ii) are satisfied but ψ is not persistently flowing. Then, for any $\varepsilon > 0$, there exists T_ε such that for every $(t,j)\in\text{dom}(\psi)$ with $t+j\ge T_\varepsilon$, the time between state broadcasts is less than ε. Choose

$$\varepsilon < \min\left\{\frac{1}{n+1}\min_i r_i^{\min}, \min_i \tau_i, \frac{1}{n+1}\delta_P, \min_i\sqrt{\mu_i}\right\}.$$

Then, we can show that all the state broadcasts in the network are synchronized from T_ε time forward due to the trigger sets $\mathcal{T}_i^{\text{synch}}$: by our choice of ε, there are at least $(n+1)\varepsilon$ broadcasts every $\min_i r_i^{\min}$ seconds. This means that at least one agent has broadcast twice in the last $\min_i r_i^{\min}$ seconds. Accordingly, all the neighbors of that agent synchronously broadcast their state at the same time due to the trigger sets $\mathcal{T}_i^{\text{synch}}$. Propagating this logic to the second-hop neighbors and so on, one can see that the entire network is synchronously broadcasting its state and this will be true for all $t+j \geq T_\varepsilon$. Let us then explore the possible causes of the next broadcast. Clearly, the next broadcast will not be due to any $\mathcal{T}_i^{\text{synch}}$, since agent broadcasts are synchronized already. Likewise, it will not be due to any $\mathcal{T}_i^{\text{request}}$ or $\mathcal{T}_i^{\text{send}}$ since, by construction, $\min_i \tau_i > \varepsilon$ time must have elapsed before $\mathcal{T}_i^{\text{request}}$ is enabled by any agent. By assumption, only n broadcasts due to the $\mathcal{T}_i^0$ can occur in δ_P seconds. Without loss of generality, we can assume that the next broadcast due to one of $\mathcal{T}_i^0$ does not occur for another $\frac{\delta_P}{n+1} > \varepsilon$ time. This leaves the $\mathcal{T}_i^e$ trigger sets. Let us look at the evolution of $(e_x)_i$ for any given $i \in \{1,\dots,n\}$. Since i has not received a broadcast from its neighbors, the evolution of $(e_x)_i$ is

$$(e_x)_i(t,j) = f_i^{QR}(\hat{x},\hat{z})(t-t_j).$$

Therefore, for $\mathcal{T}_i^e$ to have been enabled, $\min_i \sqrt{\mu_i} > \varepsilon$ time must have elapsed (the same conclusion holds for $i \in \{n+1,\dots,n+m\}$). This means that the next broadcast is not triggered in ε time, contradicting the definition of ε, and this completes the proof. □

As shown in the proof of Theorem 4.4.3, the triggers defined by $\mathcal{T}_i^e$, $\mathcal{T}_i^{\text{request}}$, $\mathcal{T}_i^{\text{send}}$, and $\mathcal{T}_i^{\text{synch}}$ do not cause non-persistency in the solutions of (4.24). If we had used (2.8) in our derivation instead of (4.2), the resulting design would not have enjoyed this attribute, cf. Remark 4.3.4. In our experience, the hypothesis in Theorem 4.4.3(ii) is always satisfied with $\delta_P = \infty$, which suggests that all solutions of (4.24) are persistently flowing.

Remark 4.4.4. (Robustness to perturbations). We briefly comment here on the robustness properties of the coordination algorithm (4.24) against perturbations. These may arise in the execution as a result of communication noise, measurement error, modeling uncertainties, or disturbances in the dynamics, among other reasons. A key advantage of modeling the execution of the coordination algorithm in the hybrid systems framework introduced in Chapter 2 is that there exist a suite of established robustness characterizations for such systems. In particular, it is fairly straightforward to verify that (4.24) is a 'well-posed' hybrid system, as defined in [43], and as a consequence of this fact, the convergence properties stated in Theorem 4.4.3(i) remain valid if the hybrid system (4.24) is subjected to sufficiently small perturbations (see e.g., [43, Theorem 7.21]). Moreover, in Chapter 3, we have shown that the continuous-time dynamics (4.9) (upon which our distributed algorithm with event-triggered communication is built) is integral-input-to-state stable, and thus robust to disturbances of finite energy. We believe that the coordination algorithm (4.24) inherits this desirable property, although we do not characterize this explicitly here for reasons of space. Nevertheless, Section 4.5 below illustrates the algorithm performance under perturbation in simulation. •

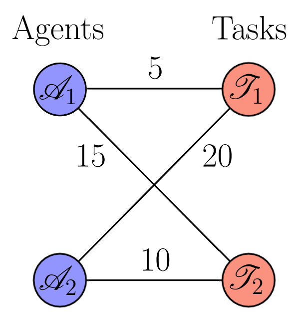


Figure 4.1: Assignment graph with agents $\mathscr{A}_1$ and $\mathscr{A}_2$ in blue, tasks $\mathscr{T}_1$ and $\mathscr{T}_2$ in red, and the benefit of a potential assignment as edge weights.

4.5 Simulations

Here we illustrate the execution of the coordination algorithm (4.24) with event-triggered communication in a multi-agent assignment example. The multi-agent assignment problem we consider is a resource allocation problem where N

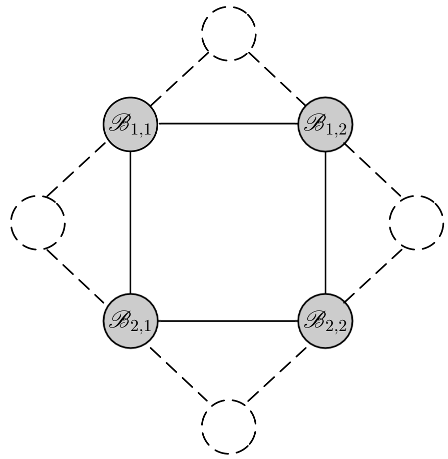


Figure 4.2: Connectivity among brokers. Broker $\mathscr{B}_{i,j}$ is responsible for determining the potential assignment of task $\mathscr{T}_j$ to agent $\mathscr{A}_i$ (see Figure 4.1). The dashed nodes represent the virtual brokers whose states correspond to the components of the Lagrange multipliers z in (4.24), see Section 3.2.2.

tasks are to be assigned to N agents. Each potential assignment of a task to an agent has an associated benefit and the global objective of the network is to maximize the sum of the benefits in the final assignment. The assignment of an agent to a task is managed by a broker and the set of all brokers use the strategy (4.24) to find the optimal assignment. Presumably, a broker is only concerned with the assignments of the agent and task that it manages and not the assignments of the entire network. Additionally, there may exist privacy concerns that limit the amount of information that a network makes available to any individual broker. These are a couple of reasons why a distributed algorithm is well-suited to solve this problem.

We consider an assignment problem with 2 agents (denoted by $\mathscr{A}_1$ and $\mathscr{A}_2$) and 2 tasks (denoted by $\mathscr{T}_1$ and $\mathscr{T}_2$) as shown in Figure 4.1. The assignment problem is to be solved by a set of 4 brokers as shown in Figure 4.2. In general, the number of brokers is the number of edges in the assignment graph. Broker $\mathscr{B}_{i,j}$ is responsible for determining the potential assignment of task $\mathscr{T}_j$ to agent $\mathscr{A}_i$ and has state $x_{i,j} \in \{0, 1\}$. Here, $x_{i,j} = 1$ means that task $\mathscr{T}_j$ is assigned to agent $\mathscr{A}_i$ (with associated benefit $c_{i,j} \in \mathbb{R}_{\geq 0}$) and $x_{i,j} = 0$ means that they are not assigned to each other. We formulate the multi-agent assignment problem as the following optimization problem,

$$\max \quad c_{1,1}x_{1,1} + c_{1,2}x_{1,2} + c_{2,1}x_{2,1} + c_{2,2}x_{2,2} \tag{4.26a}$$

$$\text{s.t.} \quad x_{1,1} + x_{1,2} = 1, \tag{4.26b}$$

$$x_{2,1} + x_{2,2} = 1, \tag{4.26c}$$

$$x_{1,1} + x_{2,1} = 1, \tag{4.26d}$$

$$x_{1,2} + x_{2,2} = 1, \tag{4.26e}$$

$$x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2} \in \{0, 1\}. \tag{4.26f}$$

Constraints (4.26b)-(4.26c) (resp. (4.26d)-(4.26e)) ensure that each agent (resp.

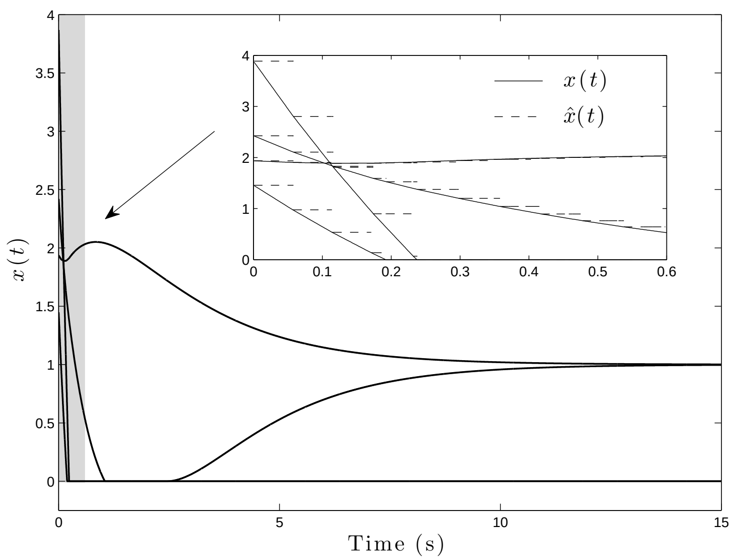


Figure 4.3: State trajectories of the brokers implementing (4.24) to solve the multi-agent assignment problem (4.27). An inlay displays the transient response in detail. The brokers' state is $x = (x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2})$ and the inlay also shows the evolution of the broadcast states, $\hat{x} = (\hat{x}_{1,1}, \hat{x}_{1,2}, \hat{x}_{2,1}, \hat{x}_{2,2})$ in dashed lines. The aggregate of the brokers' states converge to the unique solution $\mathcal{X} = \{(0, 1, 1, 0)\}$.

task) is assigned to one and only one task (resp. agent). Note that the connectivity between brokers shown in Figure 4.2 is consistent with the requirements for a distributed implementation as specified by the constraint equations of (4.26). It is known, see e.g., [93], that the relaxation $x_{i,j} \geq 0$ of the constraints (4.26f) gives rise to a linear program with an optimal solution that satisfies $x_{i,j} \in \{0, 1\}$. Thus, for our purposes, we solve instead the following linear program

$$\min \quad -5x_{1,1} - 15x_{1,2} - 20x_{2,1} - 10x_{2,2} \tag{4.27a}$$

$$\text{s.t.} \quad x_{1,1} + x_{1,2} = 1, \tag{4.27b}$$

$$x_{2,1} + x_{2,2} = 1, \tag{4.27c}$$

$$x_{1,1} + x_{2,1} = 1, \tag{4.27d}$$

$$x_{1,2} + x_{2,2} = 1, \tag{4.27e}$$

$$x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2} \geq 0, \tag{4.27f}$$

where we have also converted the maximization into a minimization by considering the negative of the objective function and substituted the values of the benefits given in Figure 4.1. Clearly, the linear program (4.27) is in standard form. Its solution set is $\mathcal{X} = \{x^*\}$, with $x^* = (x^*_{1,1}, x^*_{1,2}, x^*_{2,1}, x^*_{2,2}) = (0, 1, 1, 0)$, corresponding to the optimal assignment consisting of the pairings $(\mathscr{A}_1, \mathscr{T}_2)$ and $(\mathscr{A}_2, \mathscr{T}_1)$.

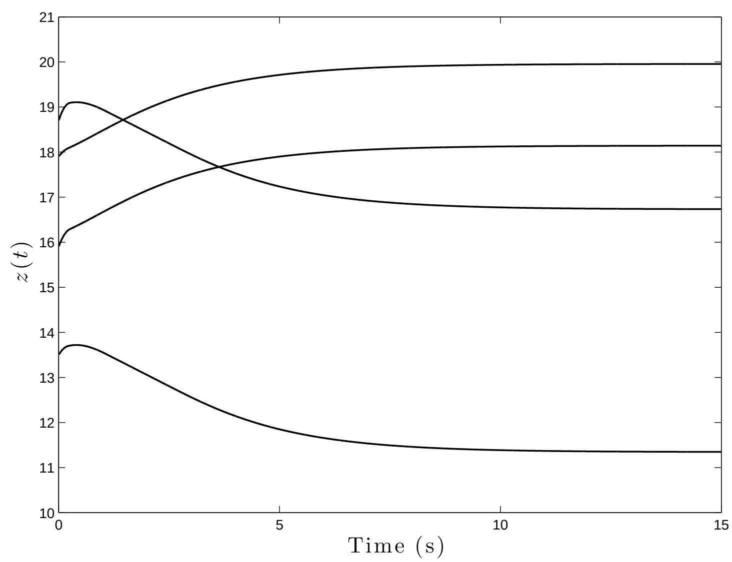


Figure 4.4: Evolution of the virtual brokers' states.

Figure 4.3 show the result of brokers executing the distributed coordination algorithm (4.24) with event-triggered communication. The trajectories of the virtual brokers is displayed in Figure 4.4, the evolution of the Lyapunov function

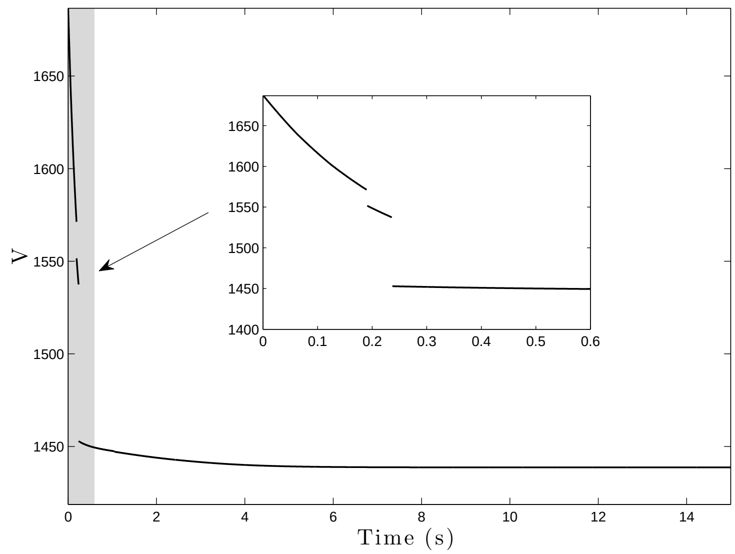


Figure 4.5: Evolution of the Lyapunov function V, which is discontinuous but decreasing.

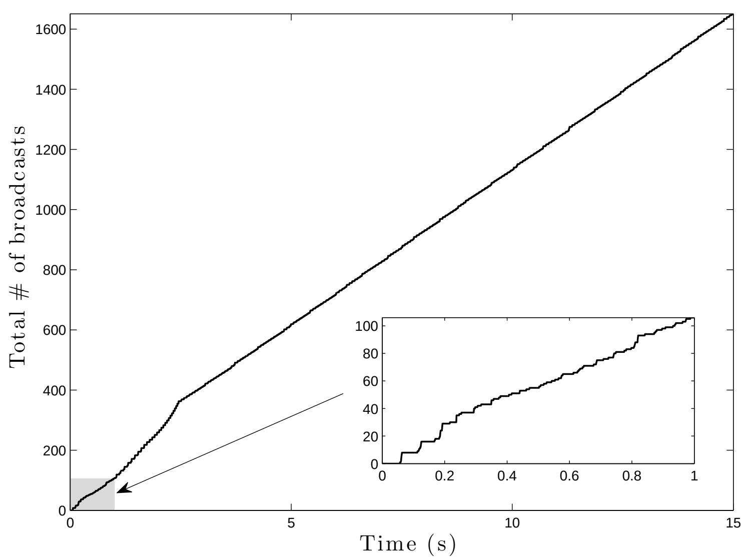


Figure 4.6: The cumulative number of broadcasts appears roughly linear and the execution is clearly persistently flowing.

is displayed in Figure 4.5, and the resulting cumulative number of broadcasts in displayed in Figure 4.6.

Figures 4.7 and 4.8 illustrate the algorithm performance in the presence

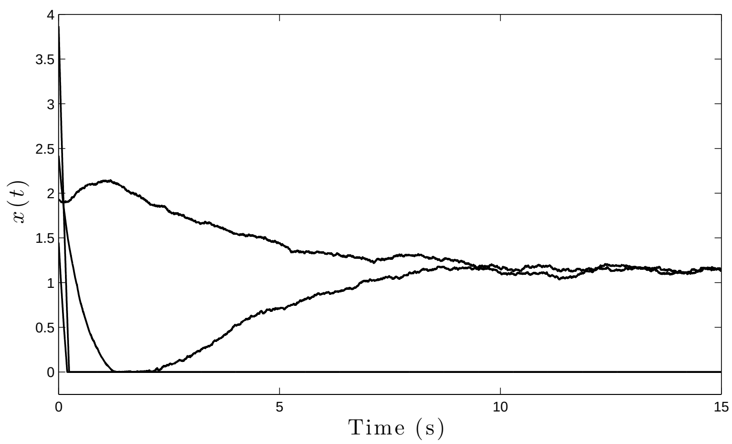


Figure 4.7: Trajectories of the noisy brokers' states as they implement (4.24) to solve the multi-agent assignment problem (4.27). The disturbances are additive noise in the communication channels.

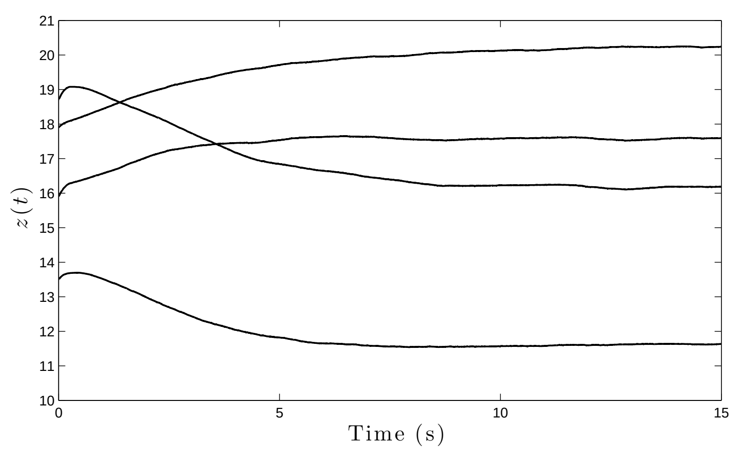


Figure 4.8: Trajectories of the noisy virtual brokers' states as they implement (4.24) to solve the multi-agent assignment problem (4.27). The disturbances are additive noise in the communication channels.

of additive white noise on the state broadcasts. In this simulation, broadcasts of information are corrupted by noise which is normally distributed with zero mean and standard deviation 1. The convergence in this case shows the algorithm's robustness to sufficiently small disturbances, as pointed out in Remark 4.4.4.

This concludes our study of distributed event-triggered linear programming.

Chapter 4, in part, is a reprint of the material [85] as it appears in 'Distributed event-triggered optimization for linear programming' by D. Richert and J. Cortés which was submitted to the 2014 IEEE Conference on Decision and Control as well as [86] as it appears in 'Distributed linear programming with event-triggered communication' by D. Richert and J. Cortés which was submitted to the SIAM Journal of Control and Optimization. The dissertation author was the primary investigator and author of these papers.

Chapter 5

Distributed bargaining in dyadic-exchange networks

In this chapter we consider resource-constrained networks where collaboration between subsystems gives rise to a more efficient use of these resources. We assume that agents are autonomous and rational and thus bargain with each other over who to collaborate with and how to split a transferable utility between them.

We begin with a problem statement, outlining the notion of bargaining outcomes in dyadic-exchange networks. Three outcomes that we focus on are stable, balanced, and Nash. Our design methodology proceeds linearly, first considering stable, then balanced, and finally Nash outcomes. We are able to show how the robust distributed linear programming algorithm of Chapter 3 can be used to find stable outcomes. Then a novel distributed balancing dynamics is proposed. The proof of convergence for this dynamics requires us to first establish boundedness of the states, and then use the LaSalle Invariance Principle to show asymptotic convergence. Distributed Nash bargaining dynamics are then proposed based on a merging of the stable and balancing dynamics. Drawing on the results of Chapter 3 and [43, Theorem 7.21], the proposed dynamics is shown be to robust to small disturbances.

As an interesting application, we conclude this chapter by applying the Nash bargaining dynamics to a wireless communication scenario. We consider multiple devices that send data to a base station according to a time division multiple access (TDMA) protocol. Devices may share their transmission time slots in order to gain an improved channel capacity. Simulation results show how a Nash outcome is achieved, yielding fair capacity improvements for each matched device and a network-wide capacity improvement of around 16%.

5.1 Problem statement

The main objective of this chapter is the design of provably correct distributed dynamics that solve the network bargaining game. This section provides a formal description of the problem. We begin by presenting the model for the group of agents and then recall various important notions of outcome for the network bargaining game.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ be an undirected weighted graph where $\mathcal{V} = \{1, \dots, n\}$ is a set of vertices, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of edges, and $W \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0}$ is a vector of edge weights indexed by edges in $\mathcal{G}$. In an exchange network, vertices correspond to agents (or players) and edges connect agents who have the ability to negotiate with each other. The set of agents that i can negotiate with are its neighbors and is denoted by $\mathcal{N}(i) := \{j : (i,j) \in \mathcal{E}\}$. Edge weights represent a transferable utility that agents may, should they come to an agreement, divide between them. Here, we assume that the network is a *dyadic-exchange* network, meaning that agents can pair with at most one other agent. Agents are selfish and seek to maximize the amount they receive. However if two agents i and j cannot come to an agreement, they forfeit the entire amount $w_{i,j}$. We consider bargaining outcomes of the following form.

Definition 5.1.1. (Outcomes). A *matching* $M \subseteq \mathcal{E}$ is a subset of edges without common vertices. An *outcome* is a pair (M, α), where $M \subseteq \mathcal{E}$ is a matching and $\alpha \in \mathbb{R}^n$ is an allocation to each agent such that $\alpha_i + \alpha_j = w_{i,j}$ if $(i,j) \in M$ and $\alpha_k = 0$ if agent k is not part of any edge in M. •

In any given outcome (M, α), an agent may decide to unilaterally deviate by matching with another neighbor. As an example, suppose that $(i,j) \in M$ and agent k is a neighbor of i. If $\alpha_i + \alpha_k < w_{i,k}$, then there is an incentive for i to deviate because it could receive an increased allocation of $\hat{\alpha}_i = w_{i,k} - \alpha_k > \alpha_i$.

Such a deviation is unilateral because k's allocation stays constant. Conversely, if $\alpha_i + \alpha_k \geq w_{i,k}$, then i does not have an incentive to deviate by matching with k. This discussion motivates the notion of a *stable outcome*, in which no agent benefits from a unilateral deviation.

Definition 5.1.2. (Stable outcome). An outcome (M, α^s) is stable if $\alpha^s \geq 0$ and

$$\alpha_i^s + \alpha_j^s \geq w_{i,j}, \quad \forall (i,j) \in \mathcal{E}. \qquad \bullet$$

Given an arbitrary matching M, it is not always possible to find allocations α^s such that (M, α^s) is a stable outcome. Thus, finding stable outcomes requires one to find an appropriate matching as well, making the problem combinatorial in the number of possible matchings.

Stable outcomes are not necessarily fair between matched agents, and this motivates the notion of *balanced* outcomes. As an example, again assume that the outcome (M, α^b) is given and that $(i,j) \in M$. The *best allocation* that i could expect to receive by matching with a neighbor other than j is

$$\mathtt{ba}_{i\backslash j}(\alpha^b) = \max_{k \in \mathcal{N}(i)\backslash j} \{w_{i,k} - \alpha_k^b\}_+.$$

Moreover, the set (possibly empty) of *best neighbors* with whom i could receive this allocation is

$$\mathtt{bn}_{i\backslash j}(\alpha^b) = \operatorname{argmax}_{k \in \mathcal{N}(i)\backslash j} \{w_{i,k} - \alpha_k^b\}_+.$$

Then, if agent i were to unilaterally deviate from the outcome and match instead with $k \in \mathtt{bn}_{i\backslash j}$, the resulting benefit of this deviation would be

$$\mathtt{ba}_{i\backslash j}(\alpha^b) - \alpha_i^b.$$

When the benefit of a deviation is the same for both i and j, we call the outcome balanced.

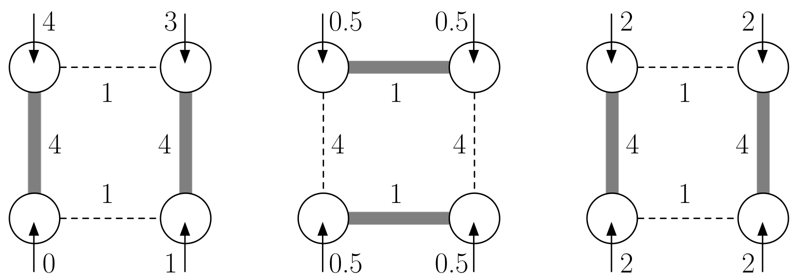
Definition 5.1.3. (Balanced outcome). An outcome (M, α^b) is balanced if for all $(i,j) \in M$,

$$\mathtt{ba}_{i\backslash j}(\alpha^b) - \alpha_i^b = \mathtt{ba}_{j\backslash i}(\alpha^b) - \alpha_j^b. \qquad \bullet$$

From its definition, it is easy to see that the main challenge in finding balanced outcomes is the fact that the allocations must satisfy a system of nonlinear (in fact, piecewise linear) equations, coupled between agents. Of course, outcomes that are both stable and balanced are desirable and what we seek in this chapter. Such outcomes are called *Nash*.

Definition 5.1.4. (Nash outcome). An outcome (M, α^N) is Nash if it is stable and balanced. •

Figure 5.1 shows an example of each outcome, highlighting their various attributes.



(a) Stable (b) Balanced (c) Nash

Figure 5.1: Stable, balanced, and Nash outcomes. For each outcome, matched agents are connected with thicker grey edges, dotted edges connect agents who decided not to match, and allocations are indicated by arrows. In the stable outcome, the 0 allocation is unfair to that node since its partner receives the whole edge weight. In the balanced outcome, agents can receive higher allocations by deviating from their matches. Nash outcomes do not exhibit either of these shortcomings.

The problem we aim to solve is to develop distributed dynamics that converge to each of the class of outcomes defined above: stable, balanced, and Nash. We refer to a dynamics as 1-*hop distributed*, or simply *distributed, over* $\mathcal{G}$ if its implementation requires each agent $i \in \{1, \dots, n\}$ only knowing (i) the states of 1-hop neighboring agents and (ii) the utilities $w_{i,j}$ for each $j \in \mathcal{N}(i)$. Likewise, we refer to a dynamics as 2-*hop distributed over* $\mathcal{G}$ if its implementation requires each agent $i \in \{1, \dots, n\}$ only knowing (i) the states of 1- and 2-hop neighboring agents and (ii) the utilities $w_{i,j}$ and $w_{j,k}$ for each $j \in \mathcal{N}(i)$ and $k \in \mathcal{N}(j)$. As agents' allocations evolve in the dynamics that follow, the quantity $w_{i,j} - \alpha_i(t)$ has the interpretation of "i's *offer* to j at time t", thus motivating the terminology *bargaining* in exchange networks.

5.2 Distributed dynamics to find stable outcomes

In this section, we propose a distributed dynamics to find stable outcomes in network bargaining. Our strategy to achieve this builds on a reformulation of the problem of finding a stable outcome in terms of finding the solutions to a linear program.

5.2.1 Stable outcomes as solutions of linear program

Here we relate the existence of stable outcomes to the solutions of a linear programming relaxation for the maximum weight matching problem. This reformulation allows us later to synthesize a distributed dynamics to find stable outcomes. Our discussion here follows [54], but because that reference does not present formal proofs of the results we need, we include them here for completeness.

We begin by recalling the formulation of the *maximum weight matching problem* on $\mathcal{G}$. Essentially, this corresponds to a matching in which the sum of the edge weights in the matching is maximal. Formally, for every $(i,j) \in \mathcal{E}$ we use variables $m_{i,j} \in \{0,1\}$ to indicate whether (i,j) is in the maximum weight matching (i.e., $m_{i,j} = 1$) or not ($m_{i,j} = 0$). Then, the solutions of the following integer program can be used to deduce a maximum weight matching,

$$\begin{aligned} \max \quad & \textstyle\sum_{(i,j)\in\mathcal{E}} w_{i,j} m_{i,j} && (5.1a) \\ \text{s.t.} \quad & \textstyle\sum_{j\in\mathcal{N}(i)} m_{i,j} \le 1, \quad \forall i \in \mathcal{V}, && (5.1b) \\ & m_{i,j} \in \{0,1\}, \quad \forall (i,j) \in \mathcal{E}. && (5.1c) \end{aligned}$$

The constraints (5.1b) ensure that each agent is matched to at most one other agent. If $m^* \in \{0,1\}^{|\mathcal{E}|}$ is a solution (indexed by edges in $\mathcal{G}$) to the above optimization problem, then a maximum weight matching is well-defined by the relationship $(i,j) \in M \Leftrightarrow m^*_{i,j} = 1$. Since (5.1) is combinatorial in the number of edges in the graph due to constraint (5.1c), we are interested in studying its linear programming relaxation,

$$\begin{aligned} \max \quad & \textstyle\sum_{(i,j)\in\mathcal{E}} w_{i,j} m_{i,j} \\ \text{s.t.} \quad & \textstyle\sum_{j\in\mathcal{N}(i)} m_{i,j} \leq 1, \quad \forall i \in \mathcal{V}, \\ & m_{i,j} \geq 0, \quad \forall (i,j) \in \mathcal{E}, \end{aligned} \tag{5.2}$$

and its associated dual

$$\begin{aligned} \min \quad & \textstyle\sum_{i\in\mathcal{V}} \alpha^s_i \\ \text{s.t.} \quad & \alpha^s_i + \alpha^s_j \geq w_{i,j}, \quad \forall (i,j) \in \mathcal{E}, \\ & \alpha^s_i \geq 0, \quad \forall i \in \mathcal{V}. \end{aligned} \tag{5.3}$$

Arguing with the KKT conditions for the relaxation (5.2), the following result states that when a stable outcome (M, α^s) exists, the matching M is a maximum weight matching on $\mathcal{G}$.

Lemma 5.2.1. (Maximum weight matchings and stable outcomes [54]). *Suppose that a stable outcome (M, α^s) exists for a given graph $\mathcal{G}$. Then M is a maximum weight matching.*

Proof. Our proof method is to encode the matching M using the indicator variables $m \in \{0,1\}^{|\mathcal{E}|}$ and then show that m is a solution of the maximum weight matching problem (5.1). To begin, for all $(i,j) \in \mathcal{E}$, let $m_{i,j} = 1$ if $(i,j) \in M$ and zero otherwise. Use m to denote the vector of $m_{i,j}$, indexed by edges in $\mathcal{G}$. Then m is feasible for the relaxation (5.2). By definition of outcome, cf. Definition 5.1.1, it holds that $m_{i,j}(\alpha^s_i + \alpha^s_j - w_{i,j}) = 0$ for all $(i,j) \in \mathcal{E}$ and $\alpha^s_i(1 - \sum_{j\in\mathcal{N}(i)} m_{i,j}) = 0$ for all $i \in \mathcal{V}$. In other words, complementary slackness is satisfied. Also, note that α^s is feasible for the dual (5.3). This means that (m, α^s) satisfy the KKT conditions for (5.2) and so m is a solution of (5.2). Since m is integral, it is also a solution of (5.1) implying that M is a maximum weight matching. This completes the proof. □

Building on this result, we show next that the existence of stable outcomes is directly related to the existence of integral solutions of the linear programming relaxation (5.2).

Lemma 5.2.2. (Existence of stable outcomes [54]). *A stable outcome exists for the graph $\mathcal{G}$ if and only if* (5.2) *admits an integral solution. Moreover, if $\mathcal{G}$ admits a stable outcome and $m^* \in \{0,1\}^{|\mathcal{E}|}$ is a solution to* (5.2)*, then $(M, \alpha^{s,*})$ is a stable outcome where the matching M is well-defined by the implication*

$$(i,j) \in M \Leftrightarrow m^*_{i,j} = 1, \tag{5.4}$$

and $\alpha^{s,*}$ is a solution to (5.3).

Proof. The proof of Lemma 5.2.1 revealed that, if a stable outcome exists, (5.2) admits an integral solution. Let us prove the other direction. By assumption, (5.2) yields an integral solution $m^* \in \{0,1\}^{|\mathcal{E}|}$ and let $\alpha^{s,*}$ be a solution to the dual (5.3). We induce the following matching: $(i,j) \in M$ if $m^*_{i,j} = 1$ and $(i,j) \notin M$ otherwise. By complementary slackness, $m^*_{i,j}(\alpha_i^{s,*} + \alpha_j^{s,*} - w_{i,j}) = 0$ for all $(i,j) \in \mathcal{E}$ and $\alpha_i^{s,*}(1 - \sum_{j \in \mathcal{N}(i)} m^*_{i,j}) = 0$ for all $i \in \mathcal{V}$. Then, it must be that $m^*_{i,j} = 1$ implies that $\alpha_i^{s,*} + \alpha_j^{s,*} = w_{i,j}$ and $\alpha_i^{s,*} = 0$ if i is not part of any matching. Thus, $(M, \alpha^{s,*})$ is a valid outcome. Next, $\alpha^{s,*}$ must be feasible for (5.3), which reveals that it is a stable allocation. Therefore, $(M, \alpha^{s,*})$ is a stable outcome. This completes the proof. □

5.2.2 Stable outcomes via distributed linear programming

Since we are interested in finding stable outcomes, from here on we make the standing assumption that one exists and that the maximum weight matching is unique. Besides its technical implications, requiring uniqueness has a practical motivation and is a standard assumption in exchange network bargaining. For example, if an agent has two equally good alternatives, it is unclear with whom it will choose to match with. It turns out that the set of graphs for which a unique maximum weight matching exists is open and dense in the set of graphs that admit a stable outcome, further justifying the assumption of uniqueness of the maximum weight matching.

Given the result in Lemma 5.2.2 above, finding a stable outcome is a matter of solving the relaxed maximum weight matching problem, where the matching is induced from the solution of (5.2) and the allocation is a solution to (5.3). Our next step is to put (5.3) in standard form by introducing slack variables $s_{i,j}$ for each $(i, j) \in \mathcal{E}$,

$$\begin{aligned} \min \quad & \textstyle\sum_{i \in \mathcal{V}} \alpha_i^s \\ \text{s.t.} \quad & \alpha_i^s + \alpha_j^s - s_{i,j} = w_{i,j}, \quad \forall (i,j) \in \mathcal{E}, \\ & \alpha_i^s \geq 0 \quad \forall i \in \mathcal{V}, \\ & s_{i,j} \geq 0 \quad \forall (i,j) \in \mathcal{E}. \end{aligned}$$

We use the notation s to represent the vector of slacks indexed by edges in $\mathcal{G}$. In the dynamics that follow, the variables s and m will be states. Thus, as a convention, we assume that each $s_{i,j}$ and $m_{i,j}$ are states of agent $\min\{i, j\}$. This means that the state of agent $i \in \mathcal{V}$ is

$$(\alpha_i^s, \{s_{i,j}\}_{j \in \mathcal{N}^+(i)}, \{m_{i,j}\}_{j \in \mathcal{N}^+(i)}) \in \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0}^{|\mathcal{N}^+(i)|} \times \mathbb{R}^{|\mathcal{N}^+(i)|},$$

where, for convenience, we denote by $\mathcal{N}^+(i) := \{j \in \mathcal{N}(i) : i < j\}$ the set of neighbors of i whose identity is greater than i.

Next, using the dynamics (3.8) of Section 3.2.1 to solve the linear program above results in the following dynamics for agent $i \in \{1, \ldots, n\}$,

$$\dot{\alpha}_i^s = \begin{cases} f_i^\alpha(\alpha^s, s, m), & \alpha_i^s > 0, \\ \max\{0, f_i^\alpha(\alpha^s, s, m)\}, & \alpha_i^s = 0, \end{cases} \tag{5.6a}$$

and, for each $j \in \mathcal{N}^+(i)$

$$\dot{s}_{i,j} = \begin{cases} f_{i,j}^s(\alpha^s, s, m), & s_{i,j} > 0, \\ \max\{0, f_{i,j}^s(\alpha^s, s, m)\}, & s_{i,j} = 0, \end{cases} \tag{5.6b}$$

$$\dot{m}_{i,j} = \alpha_i^s + \alpha_j^s - s_{i,j} - w_{i,j}, \tag{5.6c}$$

where

$$f_i^\alpha(\alpha^s, s, m) := -1 - \sum_{j \in \mathcal{N}(i)} \Big[m_{i,j} + \alpha_i^s + \alpha_j^s - s_{i,j} - w_{i,j} \Big],$$

and

$$f^s_{i,j}(\alpha^s, s, m) := -m_{i,j} - \alpha^s_i - \alpha^s_j + s_{i,j} + w_{i,j},$$

are derived from (3.7). The next result reveals how this dynamics can be used as a distributed algorithm to find stable outcomes.

Proposition 5.2.3. (Convergence to stable outcomes). *Given a graph $\mathcal{G}$, let $t \to (\alpha^s(t), s(t), m(t))$ be a trajectory of* (5.6) *starting from an initial point in $\mathbb{R}^n_{\geq 0} \times \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \times \mathbb{R}^{|\mathcal{E}|}$. Then the following limit exists*

$$\lim_{t\to\infty} (\alpha^s(t), s(t), m(t)) = (\alpha^*, s^*, m^*),$$

where $(\alpha^{s,*}, s^*)$ (resp. m^*) is a solution to (5.3) *(resp.* (5.2)*). Moreover, if a stable outcome exists, a maximum weight matching M is well-defined by the implication*

$$(i, j) \in M \quad \Leftrightarrow \quad m^*_{i,j} = 1,$$

and $(M, \alpha^{s,*})$ is a stable outcome. Finally, the dynamics (5.6) *is distributed over $\mathcal{G}$.*

The proof of the above results follows directly from Corollary 3.2.5, Lemma 5.2.2, and the assumptions made on the information available to each agent.

5.3 Distributed dynamics to find balanced outcomes

In this section, we introduce distributed dynamics that converge to balanced outcomes. Our starting point is the availability of a matching M to the network, i.e., each agent already knows if it is matched and who its partner is. Hence, the dynamics focuses on negotiating the allocations to find a balanced one. We drop this assumption later when considering Nash outcomes.

Our algorithm design is based on the observation that the condition $\alpha^b_i + \alpha^b_j = w_{i,j}$ for $(i, j) \in M$ that defines an allowable allocation for an outcome and the balance condition in Definition 5.1.3 for two matched agents can be equivalently stated as

$$0=\alpha_i^b-\frac{1}{2}(w_{i,j}+\mathtt{ba}_{i\setminus j}(\alpha^b)-\mathtt{ba}_{j\setminus i}(\alpha^b)) =: e_i^b(\alpha^b), \tag{5.7a}$$

$$0=\alpha_j^b-\frac{1}{2}(w_{i,j}-\mathtt{ba}_{i\setminus j}(\alpha^b)+\mathtt{ba}_{j\setminus i}(\alpha^b)) =: e_j^b(\alpha^b). \tag{5.7b}$$

We refer to $e_i^b, e_j^b : \mathbb{R}^n \to \mathbb{R}$ as the errors with respect to satisfying the balance condition of i and j, respectively. For an unmatched agent k, we define $e_k^b = \alpha_k^b$. We refer to $e^b(\alpha^b) \in \mathbb{R}^n$ as the vector of *balancing errors* for a given allocation. Based on the observation above, we propose the following distributed dynamics whereby agents adjust their allocations proportionally to their balancing errors,

$$\dot{\alpha}^b = -e^b(\alpha^b). \tag{5.8}$$

An important fact to note is that the equilibria of the above dynamics are, by construction, allocations in a balanced outcome. Also, note that (5.8) is continuous and requires agents to know 2-hop information, because for its pair of matched agents $(i, j) \in M$, agent i updates its own allocation (and hence its offer to j) based on $\mathtt{ba}_{j\setminus i}$.

The following result establishes the boundedness of the balancing errors under (5.8) and is useful later in establishing the asymptotic convergence of this dynamics to an allocation in a balanced outcome with matching M.

Proposition 5.3.1. (Balancing errors are bounded). *Given a matching M, let $t \mapsto \alpha^b(t)$ be a trajectory of (5.8) starting from any point in $\mathbb{R}^n$. Then*

$$t \mapsto V(e^b(\alpha^b(t))) := \tfrac{1}{2}\max_{i\in\mathcal{V}}(e_i^b(\alpha^b(t)))^2,$$

is non-increasing. Thus, $t \mapsto e^b(\alpha^b(t))$ lies in a bounded set.

Proof. Our proof strategy is to compute, for each $i \in \mathcal{V}$, the Lie derivative of e_i^b along the trajectories of (5.8). Based on these Lie derivatives, we introduce a new dynamics whose trajectories contain $t \mapsto e^b(\alpha^b(t))$ and establish the result reasoning with it.

Since e_i^b is locally Lipschitz, it is differentiable almost everywhere. Let $\Omega_i \subseteq \mathbb{R}^n$ be the set, of measure zero, of allocations for which e_i^b is not differentiable.

If i is matched, say $(i,j) \in M$, then Ω_i is precisely the set of allocations where at least one of the next best neighbor sets $\mathtt{bn}_{i\setminus j}(\alpha^b)$ or $\mathtt{bn}_{j\setminus i}(\alpha^b)$ have more than one element. If i is unmatched, then $\Omega_i = \emptyset$. Then, whenever $\alpha^b \in \mathbb{R}^n \setminus \Omega_i$, it is easy to see that for every $i \in \mathcal{V}$,

$$\mathcal{L}_{-e^b} e_i^b(\alpha^b) = \begin{cases} -e_i^b(\alpha^b), & \text{if } i \text{ is unmatched or} \\ & \text{if } \mathtt{bn}_{i\setminus j}(\alpha^b) = \emptyset \text{ and } \mathtt{bn}_{j\setminus i}(\alpha^b) = \emptyset, \\ -e_i^b(\alpha^b) + \frac{1}{2}e_\tau^b(\alpha^b), & \text{if } \mathtt{bn}_{i\setminus j}(\alpha^b) = \tau \text{ and } \mathtt{bn}_{j\setminus i}(\alpha^b) = \emptyset, \\ -e_i^b(\alpha^b) - \frac{1}{2}e_\kappa^b(\alpha^b), & \text{if } \mathtt{bn}_{i\setminus j}(\alpha^b) = \emptyset \text{ and } \mathtt{bn}_{j\setminus i}(\alpha^b) = \kappa, \\ -e_i^b(\alpha^b) + \frac{1}{2}e_\tau^b(\alpha^b) - \frac{1}{2}e_\kappa^b(\alpha^b), & \text{if } \mathtt{bn}_{i\setminus j}(\alpha^b) = \tau \text{ and } \mathtt{bn}_{j\setminus i}(\alpha^b) = \kappa. \end{cases}$$

This observation motivates our study of the dynamics

$$\dot{\xi}_i = \begin{cases} -\xi_i, & \text{if } i \text{ is unmatched or} \\ & \text{if } \mathtt{bn}_{i\setminus j}(\omega) = \emptyset \text{ and } \mathtt{bn}_{j\setminus i}(\omega) = \emptyset, \\ -\xi_i + \frac{1}{2}\xi_\tau, & \text{if } \mathtt{bn}_{i\setminus j}(\omega) = \tau \text{ and } \mathtt{bn}_{j\setminus i}(\omega) = \emptyset, \\ -\xi_i - \frac{1}{2}\xi_\kappa, & \text{if } \mathtt{bn}_{i\setminus j}(\omega) = \emptyset \text{ and } \mathtt{bn}_{j\setminus i}(\omega) = \kappa, \\ -\xi_i + \frac{1}{2}\xi_\tau - \frac{1}{2}\xi_\kappa, & \text{if } \mathtt{bn}_{i\setminus j}(\omega) = \tau \text{ and } \mathtt{bn}_{j\setminus i}(\omega) = \kappa, \end{cases} \tag{5.9a}$$

$$\dot{\omega}_i = -\xi_i, \tag{5.9b}$$

for every $i \in \mathcal{V}$, defined on $\mathbb{R}^n \times (\mathbb{R}^n \setminus \Omega)$, where $\Omega := \cup_{i\in\mathcal{V}}\Omega_i$. For convenience, we use the shorthand notation $F = (F^1, F^2) : \mathbb{R}^n \times (\mathbb{R}^n \setminus \Omega) \to \mathbb{R}^n \times \mathbb{R}^n$ to refer to (5.9). Note that F is piecewise continuous (because F^1 is piecewise continuous, while F^2 is continuous). Therefore, we understand its trajectories in the sense of Filippov. Using (2.3), we compute the Filippov set-valued map, defined on $\mathbb{R}^n \times \mathbb{R}^n$, for any matched i and $(\xi,\omega) \in \mathbb{R}^n \times \mathbb{R}^n$, as

$$\begin{aligned} \mathcal{F}[F_i^1](\xi,\omega) = \Big\{ -\xi_i - \tfrac{1}{2}\sum_{\tau\in\mathtt{bn}_{i\setminus j}(\omega)}\lambda_\tau^i\xi_\tau + \tfrac{1}{2}\sum_{\kappa\in\mathtt{bn}_{j\setminus i}(\omega)}\mu_\kappa^i\xi_\kappa : \\ \lambda^i \in \mathbb{R}^n_{\geq 0} \text{ is s.t. } \sum_{\tau\in\mathtt{bn}_{i\setminus j}(\omega)}\lambda_\tau^i = 1 \text{ if } \mathtt{bn}_{i\setminus j}(\omega) \neq \emptyset \text{ and} \\ \mu^i \in \mathbb{R}^n_{\geq 0} \text{ is s.t. } \sum_{\kappa\in\mathtt{bn}_{j\setminus i}(\omega)}\mu_\kappa^i = 1 \text{ if } \mathtt{bn}_{j\setminus i}(\omega) \neq \emptyset \Big\}. \end{aligned}$$

Here, we make the convention that the empty sum is zero. If i is unmatched, then $\mathcal{F}[F_i^1](\xi,\omega) = \{-\xi_i\}$. Furthermore, $\mathcal{F}[F_i^2] = \{-\xi_i\}$ for all $i \in \mathcal{V}$. Based on the discussion so far, we know that $t \mapsto (e^b(\alpha^b(t)), \alpha^b(t))$ is a Filippov trajectory of (5.9) with initial condition $(e^b(\alpha^b(0)), \alpha^b(0)) \in \mathbb{R}^n \times \mathbb{R}^n$. Thus, to prove the result, it is sufficient to establish the monotonicity of

$$V(\xi) = \max_{i\in\mathcal{V}} \frac{1}{2}\xi_i^2,$$

along (5.9). For notational purposes, we denote

$$\Omega(\xi) := \text{argmax}_{i\in\mathcal{V}} \frac{1}{2}\xi_i^2.$$

The generalized gradient of V is

$$\partial V(\xi) = \Big\{ \sum_{i\in\Omega(\xi)} \eta_i h_i \xi_i : \eta \in \mathbb{R}^n_{\geq 0} \text{ s.t. } \sum_{i\in\Omega(\xi)} \eta_i = 1 \Big\},$$

where $h_i \in \mathbb{R}^n$ is the unit vector with 1 in its i^{th} component and 0 elsewhere. Then, the set-valued Lie derivative of V along $\mathcal{F}[F]$ is given next

$$\begin{aligned}
\mathcal{L}_{\mathcal{F}[F]}V(\xi) &= \{a \in \mathbb{R} : \text{there exists } v \in \mathcal{F}[F](\xi,\omega) \text{ s.t. } a = \zeta^T v \text{ for all } \zeta \in \partial V(\xi)\}, \\
&= \Big\{ a \in \mathbb{R} : \text{for each } i \in \mathcal{V} \text{ there exists } \lambda^i \in \mathbb{R}^n_{\geq 0} \text{ with} \sum_{\tau\in\mathtt{bn}_{i\setminus j}(\omega)} \lambda^i_\tau = 1 \text{ if} \\
&\quad \mathtt{bn}_{i\setminus j}(\omega) \neq \emptyset \text{ and } \mu^i \in \mathbb{R}^n_{\geq 0} \text{ with } \sum_{\kappa\in\mathtt{bn}_{j\setminus i}(\omega)} \mu^i_\kappa = 1 \text{ if } \mathtt{bn}_{j\setminus i}(\omega) \neq \emptyset \text{ s.t.} \\
&\quad a = \Big(\sum_{i\in\mathcal{V}} h_i \Big[-\xi_i - \tfrac{1}{2} \sum_{\tau\in\mathtt{bn}_{i\setminus j}(\omega)} \lambda^i_\tau \xi_\tau + \tfrac{1}{2} \sum_{\kappa\in\mathtt{bn}_{j\setminus i}(\omega)} \mu^i_\kappa \xi_\kappa \Big] \Big)^T \Big(\sum_{i\in\Omega(\xi)} \eta_i h_i \xi_i \Big) \text{ for all} \\
&\quad \eta \in \mathbb{R}^n_{\geq 0} \text{ with} \sum_{i\in\Omega(\xi)} \eta_i = 1 \Big\}, \\
&= \Big\{ a \in \mathbb{R} : \text{for each } i \in \Omega(\xi) \text{ there exists } \lambda^i \in \mathbb{R}^n_{\geq 0} \text{ with} \sum_{\tau\in\mathtt{bn}_{i\setminus j}(\omega)} \lambda^i_\tau = 1 \text{ if} \\
&\quad \mathtt{bn}_{i\setminus j}(\omega) \neq \emptyset \text{ and } \mu^i \in \mathbb{R}^n_{\geq 0} \text{ with } \sum_{\kappa\in\mathtt{bn}_{j\setminus i}(\omega)} \mu^i_\kappa = 1 \text{ if } \mathtt{bn}_{j\setminus i}(\omega) \neq \emptyset \text{ s.t.} \\
&\quad a = \sum_{i\in\Omega(\xi)} \eta_i \Big(-\xi_i^2 - \tfrac{1}{2} \sum_{\tau\in\mathtt{bn}_{i\setminus j}(\omega)} \lambda^i_\tau \xi_i \xi_\tau + \tfrac{1}{2} \sum_{\kappa\in\mathtt{bn}_{j\setminus i}(\omega)} \mu^i_\kappa \xi_i \xi_\kappa \Big) \text{ for all} \\
&\quad \eta \in \mathbb{R}^n_{\geq 0} \text{ with} \sum_{i\in\Omega(\xi)} \eta_i = 1 \Big\}.
\end{aligned} \tag{5.10}$$

To upper bound the element in $\mathcal{L}_{\mathcal{F}[F]}V(\xi)$, note that

$$-\tfrac{1}{2}\sum_{\tau\in\texttt{bn}_{i\setminus j}(\omega)}\lambda^i_\tau\xi_i\xi_\tau \leq \tfrac{1}{4}\sum_{\tau\in\texttt{bn}_{i\setminus j}(\omega)}\lambda^i_\tau(\xi_i^2+\xi_\tau^2),$$

where we have used the inequality $ab \leq \frac{1}{2}a^2 + \frac{1}{2}b^2$ for $a, b \in \mathbb{R}$. For $\sum_{\tau\in\texttt{bn}_{i\setminus j}(\omega)}\lambda^i_\tau = 1$ and $i \in \Omega(\xi)$ (that is $\xi_i^2 \geq \xi_\tau^2$ for all $\tau \in \texttt{bn}_{i\setminus j}(\omega)$), we can further refine the bound as,

$$\tfrac{1}{2}\sum_{\tau\in\texttt{bn}_{i\setminus j}(\omega)}\lambda^i_\tau\xi_i\xi_\tau \leq \tfrac{1}{2}\xi_i^2,$$

The analogous bound

$$\tfrac{1}{2}\sum_{\kappa\in\texttt{bn}_{j\setminus i}(\omega)}\mu^i_\kappa\xi_i\xi_\kappa \leq \tfrac{1}{2}\xi_i^2,$$

can be derived similarly if $\sum_{\kappa\in\texttt{bn}_{j\setminus i}(\omega)}\mu^i_\kappa = 1$ and $i \in \Omega(\xi)$. Using these bounds in the Lie derivative (5.10) and noting that $\sum_{i\in\Omega(\xi)}\eta_i = 1$, it is straightforward to see that for any element $a \in \mathcal{L}_{\mathcal{F}[F]}V(\xi)$ it holds that $a \leq 0$. It follows that $t \mapsto V(\xi(t))$ and thus $t \mapsto V(e^b(\alpha^b(t)))$ is non-increasing and $t \mapsto e^b(\alpha^b(t))$ lies in the bounded set $V^{-1}(e^b(\alpha^b(0)))$, which completes the proof. □

The next result establishes the local stability of the balanced allocations associated with a given matching and plays a key role later in establishing the global asymptotic pointwise convergence of the dynamics (5.8).

Proposition 5.3.2. (Local stability of each balanced allocation). *Given a matching $M \subseteq \mathcal{E}$, let $\mathcal{B}_M = \{\alpha^{b,*} \in \mathbb{R}^n \mid (M, \alpha^{b,*})$ is a balanced outcome$\}$. Then every allocation in $\mathcal{B}_M$ is locally stable under the dynamics* (5.8).

The proof of the above result makes use of the upper-semicontinuity of the next-best-neighbor sets.

Lemma 5.3.3. (Upper-semicontinuity of the next-best-neighbor sets map). *Let $\alpha^{b,*} \in \mathbb{R}^n$. Then there exists $\varepsilon > 0$ such that, for all $(i, j) \in \mathcal{E}$ and all $\|\alpha^b - \alpha^{b,*}\| < \varepsilon$, the following inclusion holds*

$$\texttt{bn}_{i\setminus j}(\alpha^b) \subseteq \texttt{bn}_{i\setminus j}(\alpha^{b,*}).$$

Proof. Note that, since the number of edges is finite, it is enough to prove that such ε exists for each edge $(i,j) \in \mathcal{E}$ (because then one takes the minimum over all of them). Therefore, let $(i,j) \in \mathcal{E}$ and, arguing by contradiction, assume that for every $\varepsilon > 0$, there exists α^b with $\|\alpha^b - \alpha^{b,*}\| < \varepsilon$ such that $\mathtt{bn}_{i\setminus j}(\alpha^b) \not\subseteq \mathtt{bn}_{i\setminus j}(\alpha^{b,*})$. Equivalently, suppose that $\{\alpha^{b,k}\}_{k=1}^{\infty}$ is a sequence converging to $\alpha^{b,*}$ such that, for every k, there exists a $\tau^k \in \mathtt{bn}_{i\setminus j}(\alpha^{b,k}) \setminus \mathtt{bn}_{i\setminus j}(\alpha^{b,*})$. By definition of the next-best-neighbor set, it must be that

$$w_{i,\tau^k} - \alpha^{b,k}_{\tau^k} \geq w_{i,\tau} - \alpha^{b,k}_{\tau},$$

for all $\tau \in \mathcal{N}(i) \setminus j$. Since $\mathcal{N}(i) \setminus j$ has a finite number of elements, there must be some $\hat{\tau} \in \mathcal{N}(i)$ such that $\tau^k = \hat{\tau}$ infinitely often. Therefore, let $\{k_\ell\}_{\ell=1}^{\infty}$ be a subsequence such that $\tau^{k_\ell} = \hat{\tau}$ for all ℓ. Then

$$w_{i,\hat{\tau}} - \alpha^{b,k_\ell}_{\hat{\tau}} \geq w_{i,\tau} - \alpha^{b,k_\ell}_{\tau},$$

for all $\tau \in \mathcal{N}(i) \setminus j$. Taking now the limit as $\ell \to \infty$,

$$w_{i,\hat{\tau}} - \alpha^{b,*}_{\hat{\tau}} \geq w_{i,\tau} - \alpha^{b,*}_{\tau},$$

for all $\tau \in \mathcal{N}(i) \setminus j$, which contradicts $\hat{\tau} \notin \mathtt{bn}_{i\setminus j}(\alpha^{b,*})$. $\square$

We may now prove the local stabliity of balanced allocations.

Proof of Proposition 5.3.2. Take an arbitrary balanced allocation $\alpha^{b,*} \in \mathcal{B}_M$ and consider the change of coordinates $\tilde{\alpha}^b = \alpha^b - \alpha^{b,*}$. Then

$$\dot{\tilde{\alpha}}^b = -e^b(\tilde{\alpha}^b + \alpha^{b,*}).$$

For brevity, denote this dynamics $\tilde{F} : \mathbb{R}^n \to \mathbb{R}^n$. We compute the Lie derivative of

$$V(\tilde{\alpha}^b) = \frac{1}{2} \max_{i \in \mathcal{V}} (\tilde{\alpha}^b_i)^2,$$

along $\tilde{F}$. The derivation is very similar the one used in the proof of Proposition 5.3.1,

$$\mathcal{L}_{\tilde{F}} V(\tilde{\alpha}^b) = \left\{ a \in \mathbb{R} : a = -\sum_{i \in \mathbb{M}(\tilde{\alpha}^b)} \lambda_i \tilde{\alpha}^b_i e^b_i(\tilde{\alpha}^b + \tilde{\alpha}^{b,*}),\ \text{for all } \lambda \in \mathbb{R}^n_{\geq 0} \text{ s.t.} \sum_{i \in \mathbb{M}(\tilde{\alpha}^b)} \lambda_i = 1 \right\},$$

where

$$\mathbb{M}(\tilde{\alpha}^b) := \frac{1}{2}\operatorname{argmax}_{i\in\mathcal{V}}(\tilde{\alpha}_i^b)^2.$$

Consider one of the specific summands $-\tilde{\alpha}_i^b e_i^b(\tilde{\alpha}^b + \alpha^{b,*})$ for some $i \in \mathbb{M}(\tilde{\alpha}^b)$. For $(i,j) \in M$, take $\tau \in \texttt{bn}_{i\backslash j}(\tilde{\alpha}^b + \alpha^{b,*})$ and $\kappa \in \texttt{bn}_{j\backslash i}(\tilde{\alpha}^b + \alpha^{b,*})$ so that we can write,

$$-\tilde{\alpha}_i^b e_i^b(\tilde{\alpha}^b + \tilde{\alpha}^{b,*}) = -\tilde{\alpha}_i^b(\tilde{\alpha}_i^b + \alpha_i^{b,*} - \tfrac{1}{2}(w_{i,j} + w_{i,\tau} - \tilde{\alpha}_\tau^b - \alpha_\tau^{b,*} - w_{j,\kappa} + \tilde{\alpha}_\kappa^b + \alpha_\kappa^{b,*})).$$

Now, according to Lemma 5.3.3, there exists $\varepsilon > 0$ such that, for all $(k,l) \in \mathcal{E}$, we have

$$\texttt{bn}_{k\backslash l}(\tilde{\alpha}^b + \alpha^{b,*}) = \texttt{bn}_{k\backslash l}(\alpha^b) \subseteq \texttt{bn}_{k\backslash l}(\alpha^{b,*}),$$

for all α^b such that $\|\tilde{\alpha}^b\| = \|\alpha^b - \alpha^{b,*}\| < \varepsilon$. Therefore, for such allocations, we have $\tau \in \texttt{bn}_{i\backslash j}(\alpha^{b,*})$ and $\kappa \in \texttt{bn}_{j\backslash i}(\alpha^{b,*})$, and hence

$$\begin{aligned}
-\tilde{\alpha}_i^b e_i^b(\tilde{\alpha}^b + \tilde{\alpha}^{b,*}) &= -\tilde{\alpha}_i^b(\tilde{\alpha}_i^b + \tfrac{1}{2}\tilde{\alpha}_\tau^b - \tfrac{1}{2}\tilde{\alpha}_\kappa^b + e_i^b(\alpha^{b,*})),\\
&= -(\tilde{\alpha}_i^b)^2 - \tfrac{1}{2}\tilde{\alpha}_i^b\tilde{\alpha}_\tau^b + \tfrac{1}{2}\tilde{\alpha}_i^b\tilde{\alpha}_\kappa^b,\\
&\leq -(\tilde{\alpha}_i^b)^2 + \tfrac{1}{4}(\tilde{\alpha}_i^b)^2 + \tfrac{1}{4}(\tilde{\alpha}_\tau^b)^2 + \tfrac{1}{4}(\tilde{\alpha}_i^b)^2 + \tfrac{1}{4}(\tilde{\alpha}_\kappa^b)^2 \leq 0,
\end{aligned}$$

where we have used the fact that $\alpha^{b,*} \in \mathcal{B}_M$ in the second equality, the inequality $ab \leq \frac{1}{2}a^2 + \frac{1}{2}b^2$ for $a, b \in \mathbb{R}$ in the first inequality and the fact that $i \in \mathbb{M}(\tilde{\alpha}^b)$ in the last inequality. Thus $a \leq 0$ for each $a \in \mathcal{L}_{\tilde{F}}\tilde{V}(\tilde{\alpha}^b)$ when $\|\tilde{\alpha}^b\| \leq \varepsilon$, which means that $\tilde{\alpha}^b = 0$ is locally stable. In the original coordinates, $\alpha^b = \alpha^{b,*}$ is locally stable. Since $\alpha^{b,*}$ is arbitrary, we deduce that every allocation in $\mathcal{B}_M$ is locally stable. □

The boundedness of the balancing errors together with the local stability of the balanced allocations under the dynamics allow us to employ the LaSalle Invariance Principle, cf. Theorem 2.3.1 in the proof of the next result and establish the pointwise convergence of the dynamics to an allocation in a balanced outcome with matching M.

Proposition 5.3.4. (Convergence to a balanced outcome). *Given a matching M, let $t \to \alpha^b(t)$ be a trajectory of* (5.6) *starting from an initial point in $\mathbb{R}^n$. Then $t \mapsto (M, \alpha^b(t))$ converges to a balanced outcome. Moreover, the dynamics* (5.8) *is distributed with respect to 2-hop neighborhoods over $\mathcal{G}$.*

Proof. Note that, for each pair of matched agents $(i,j) \in M$, the sum $\dot{\alpha}_i^b + \dot{\alpha}_j^b = w_{i,j} - (\alpha_i^b + \alpha_j^b)$, implying that $\alpha_i^b(t) + \alpha_j^b(t) \to w_{i,j}$ exponentially fast. For each unmatched agent k, one has that $\dot{\alpha}_k^b = -\alpha_k^b$, implying that $\alpha_i^b(t) \to 0$ exponentially fast. Therefore, it follows that $t \mapsto (M, \alpha^b(t))$ converges to the set of (valid) outcomes. It remains to further show that it converges to the set of balanced outcomes. Following the approach employed in the proof of Proposition 5.3.1, we argue with the trajectories of (5.9), which we showed contain the trajectory $t \mapsto e^b(\alpha^b(t))$. For matched agents $(i,j) \in M$,

$$\dot{\xi}_i + \dot{\xi}_j = -(\xi_i + \xi_j), \tag{5.11}$$

under the dynamics (5.9). Interestingly, this dynamics is independent of ω. Thus, using the Lyapunov function

$$\widetilde{V}(\xi) = \tfrac{1}{2} \sum_{(i,j)\in M} (\xi_i + \xi_j)^2 + \tfrac{1}{2} \sum_{\substack{\{i\in\mathcal{V}: \\ i \text{ is unmatched}\}}} (\xi_i)^2,$$

it is trivial to see that

$$\mathcal{L}_{\mathcal{F}[F]}\widetilde{V}(\xi) = - \sum_{(i,j)\in M} (\xi_i + \xi_j)^2 - \sum_{\substack{\{i\in\mathcal{V}: \\ i \text{ is unmatched}\}}} (\xi_i)^2,$$

which, again, is independent of ω. By the boundedness of $t \mapsto \xi(t)$ established in Proposition 5.3.1, and using $\widetilde{V}$, we are now able to apply the LaSalle Invariance Principle, cf. Theorem 2.3.1, which asserts that the trajectory $t \mapsto \xi(t)$ converges to the largest weakly positively invariant set $\mathcal{M}$ contained in

$$\begin{aligned} L :&= \{\xi \in \mathbb{R}^n : \mathcal{L}_{\mathcal{F}[F]}\widetilde{V}(\xi) = 0\}, \\ &= \{\xi \in \mathbb{R}^n : \xi_i = -\xi_j, \ \forall (i,j) \in M, \text{ and } \xi_i = 0 \text{ if } i \text{ is unmatched}\}. \end{aligned}$$

Incidentally, this set is closed already which is why we omit the closure operator. We next show, using the fact that $t \mapsto V(\xi(t))$ is non-increasing (cf. Proposition 5.3.1) and the weak invariance of $\mathcal{M}$, that in fact $\mathcal{M} = \{0\}$. Take a point $\xi \in \mathcal{M} \subseteq L$ and take an $i \in \Omega(\xi)$. If i is unmatched, then $\xi_i = 0$ already and the proof would be complete. So, assume $(i,j) \in M$ for some $j \in \mathcal{V}$. Then, $\xi_j = -\xi_i$

and it also holds that $\dot{\xi}_i = -\dot{\xi}_j$ (see e.g., (5.11)). In fact, it must be that $\dot{\xi}_i = \dot{\xi}_j = 0$, otherwise one of ξ_i or ξ_j would be increasing, which would contradict $t \mapsto V(\xi(t))$ being non-increasing. If $\texttt{bn}_{i\setminus j}(\omega) = \texttt{bn}_{j\setminus i}(\omega) = \emptyset$ then $0 = \dot{\xi}_i = -\xi_i = \xi_j$, which would complete the proof. Suppose then that $\tau = \texttt{bn}_{i\setminus j}(\omega)$ and $\texttt{bn}_{j\setminus i}(\omega) = \emptyset$. Then $0 = \dot{\xi}_i = -\xi_i + \frac{1}{2}\xi_\tau$, which contradicts $i \in \Omega(\xi)$ (unless of course $\xi_i = 0$, which would complete the proof). A similar argument holds if $\texttt{bn}_{i\setminus j}(\omega) = \emptyset$ and $\texttt{bn}_{j\setminus i}(\omega) = \kappa$. The final case is if $\texttt{bn}_{i\setminus j}(\omega) = \tau$ and $\texttt{bn}_{j\setminus i}(\omega) = \kappa$. In this case, $0 = \dot{\xi}_i = -\xi_i + \frac{1}{2}\xi_\tau - \frac{1}{2}\xi_\kappa$. So as not to contradict $i \in \Omega(\xi)$, it must be that $\xi_i = -\xi_\tau = \xi_\kappa$, which means that $\tau, \kappa \in \Omega(\xi)$ as well. Therefore, using the same argument we used for i, it must be that $0 = \dot{\xi}_\tau = \dot{\xi}_\kappa$. Assume without loss of generality that ξ_τ is strictly negative (if it were zero the proof would be complete and if it were positive we could argue instead with ξ_κ). This means that ξ_τ grows larger at a constant rate since $\dot{\omega}_\tau = -\xi_\tau$. At some time, it would happen that $\omega_\tau > w_{i,\tau}$, which would make $\texttt{bn}_{i\setminus j}(\omega) = \emptyset$. This corresponds to a case we previously considered where we showed that, so as not to contradict the monotonicity of $t \mapsto V(\xi(t))$ it must be that $\xi_i = 0$. In summary, $\mathcal{M} = \{0\} \subset \mathbb{R}^n$, so $\xi(t) \to 0$. By construction of the dynamics (5.9) it follows that $e^b(\alpha^b(t)) \to 0$ which means, by construction of e^b, that $(M, \alpha^b(t))$ converges to the set of balanced outcomes. This, along with the local stability of each balanced allocation (cf. Proposition 5.3.2) is sufficient to ensure pointwise convergence to a balanced outcome [49, Proposition 2.2]. Finally, it is clear from (5.8) that the dynamics is distributed with respect to 2-hop neighborhoods, which completes the proof. □

5.4 Distributed dynamics to find Nash outcomes

In this section, we combine the previous developments to propose distributed dynamics that converge to Nash outcomes. The design of this dynamics is inspired by the following result from [10] revealing that balanced outcomes associated with maximum weight matchings are stable.

Proposition 5.4.1. (Balanced implies stable). *Let M be a maximum weight matching on $\mathcal{G}$ and suppose that $\mathcal{G}$ admits a stable outcome. Then, a balanced* *outcome of the form* (M, α^b) *is also stable, and thus Nash.*

In a nutshell, our proposed dynamics combine the fact that (i) the distributed dynamics (5.6) of Section 5.2 allow agents to determine a maximum weight matching and (ii) given such a maximum weight matching, the distributed dynamics (5.8) of Section 5.3 converge to balanced outcomes. The combination of these facts with Proposition 5.4.1 yields the desired convergence to Nash outcomes.

When putting the two dynamics together, however, one should note that the convergence of (5.6) is asymptotic, and hence agents implement (5.8) before the final stable matching is realized. To do this, we have agents guess with whom (if any) they will be matched in the final Nash outcome. An agent i guesses that it will match with $j \in \mathcal{N}(i)$ if the current value of the matching state $m_{i,j}(t)$ coming from the dynamics (5.6) is closest to 1 as compared to all other neighbors in $\mathcal{N}(i) \setminus j$. As we show later, this guess becomes correct in finite time. Formally, agent i predicts its *partner* by computing

$$\mathcal{P}_i(m) = \{j \in \mathcal{N}(i) : |m_{i,j} - 1| < |m_{i,k} - 1|, \forall k \in \mathcal{N}(i) \setminus j\}.$$

Clearly, $\mathcal{P}_i(m)$ is at most a singleton and can be computed by i using local information. If $\mathcal{P}_i(m) = \{j\}$, we use the slight abuse of notation and write $\mathcal{P}_i(m) = j$.

With the above discussion in mind, we next propose the following distributed strategy: each agent $i \in \mathcal{V}$ implements its corresponding dynamics in (5.6) to find a stable outcome but only begins balancing its allocation if, for some $j \in \mathcal{N}(i)$, agents i and j identify each other as partners. Formally, this dynamics is represented by, for each $i \in \mathcal{V}$,

$$\dot{\alpha}_i^s = \begin{cases} f_i^\alpha(\alpha^s, s, m), & \alpha_i^s > 0, \\ \max\{0, f_i^\alpha(\alpha^s, s, m)\}, & \alpha_i^s = 0, \end{cases} \tag{5.12a}$$

$$\dot{\alpha}_i^b = \begin{cases} -e_i^b(\alpha^b), & \text{if for some } j \in \mathcal{N}(i), \mathcal{P}_i(m) = j \text{ and } \mathcal{P}_j(m) = i, \\ -\alpha_i^b, & \text{otherwise}, \end{cases} \tag{5.12b}$$

and, for each $j \in \mathcal{N}^+(i)$,

$$\dot{s}_{i,j} = \begin{cases} f^s_{i,j}(\alpha^s, s, m), & s_{i,j} > 0, \\ \max\{0, f^s_{i,j}(\alpha^s, s, m)\}, & s_{i,j} = 0, \end{cases} \tag{5.12c}$$

$$\dot{m}_{i,j} = \alpha^s_i + \alpha^s_j - s_{i,j} - w_{i,j}. \tag{5.12d}$$

The state of agent $i \in \mathcal{V}$ is then

$$(\alpha^s_i, \alpha^b_i, \{s_{i,j}\}_{j \in \mathcal{N}^+(i)}, \{m_{i,j}\}_{j \in \mathcal{N}^+(i)}) \in \mathbb{R}_{\geq 0} \times \mathbb{R} \times \mathbb{R}^{|\mathcal{N}^+(i)|}_{\geq 0} \times \mathbb{R}^{|\mathcal{N}^+(i)|}.$$

For convenience, we denote the dynamics (5.12) by

$$F^{\text{Nash}} : \mathbb{R}^n_{\geq 0} \times \mathbb{R}^n \times \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \times \mathbb{R}^{|\mathcal{E}|} \to \mathbb{R}^n_{\geq 0} \times \mathbb{R}^n \times \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \times \mathbb{R}^{|\mathcal{E}|}.$$

The dynamics (5.12) can be viewed as a cascade system, with the states m feeding into the balancing dynamics (5.12b). The next result establishes the asymptotic convergence of this cascade system.

Theorem 5.4.2. (Asymptotic convergence to Nash outcomes). *Let $t \to (\alpha^s(t), \alpha^b(t), s(t), m(t))$ be a trajectory of (5.12) starting from an initial point in $\mathbb{R}^n_{\geq 0} \times \mathbb{R}^n \times \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \times \mathbb{R}^{|\mathcal{E}|}$. Then, if there exists a stable outcome, for some $T > 0$ the maximum weight matching M is well-defined by the implication*

$$(i, j) \in M \quad \Leftrightarrow \quad \mathcal{P}_i(m(t)) = j \text{ and } \mathcal{P}_j(m(t)) = i.$$

for all $t \geq T$. Furthermore, $t \mapsto (M, \alpha^b(t))$ converges to a Nash outcome. Moreover, (5.12) is distributed with respect to 2-hop neighborhoods over $\mathcal{G}$.

Proof. Let $m^* \in \mathbb{R}^{|\mathcal{E}|}$ be the unique integral solution of (5.2). The asymptotic convergence properties of (5.6), cf. Proposition 5.2.3, guarantee that, for every $\varepsilon > 0$, there exists $T > 0$ such that, for all $t \geq T$,

$$\varepsilon > \begin{cases} |m_{i,j}(t) - 1|, & \text{if } m^*_{i,j} = 1, \\ |m_{i,j}(t)|, & \text{if } m^*_{i,j} = 0. \end{cases}$$

Thus, taking $\varepsilon < \frac{1}{2}$, it is straightforward to see that the matching induced by the implication

$$(i, j) \in M \quad \Leftrightarrow \quad \mathcal{P}_i(m(t)) = j \text{ and } \mathcal{P}_j(m(t)) = i,$$

is well-defined, a maximum weight matching, and constant for all $t \geq T$. Then, considering only $t \geq T$ and applying Propositions 5.3.4 and 5.4.1, we deduce that $t \mapsto (M, \alpha^b(t))$ converges to a Nash outcome. The fact that (5.12) is distributed with respect to 2-hop neighborhoods follows from its definition, which completes the proof. □

Finally, we comment on the robustness properties of the Nash bargaining dynamics (5.12) against perturbations such as communication noise, measurement error, modeling uncertainties, or disturbances. A central motivation for using the linear programming dynamics (3.8), and continuous-time dynamics in general, is that there exist various established robustness characterizations for them. In particular, using previously established results from Chapter 3, it holds that (5.12) is a 'well-posed' dynamics, as defined in [43]. As a straightforward consequence of [43, Theorem 7.21], the Nash bargaining dynamics is robust to small perturbations, as we state next.

Corollary 5.4.3. (Robustness to small perturbations). *Given a graph $\mathcal{G}$, assume there exists a stable outcome and let $t \to (\alpha^s(t), \alpha^b(t), s(t), m(t))$ be a trajectory, starting from an initial point in $\mathbb{R}^n_{\geq 0} \times \mathbb{R}^n \times \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \times \mathbb{R}^{|\mathcal{E}|}$, of the perturbed dynamics*

$$(\dot{\alpha}^s, \dot{\alpha}^b, \dot{s}, \dot{m}) = F^{\text{Nash}}(\alpha^s + d_1, \alpha^b + d_2, s + d_3, m + d_4) + d_5,$$

where $d_1, d_2 : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^n$, $d_3, d_4 : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^{|\mathcal{E}|}$, and $d_5 : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^{2n+2|\mathcal{E}|}$ are disturbances. Then, for every $\varepsilon > 0$, there exist $\delta, T > 0$ such that, for $\max_i \|d_i\|_\infty < \delta$, the maximum weight matching M is well-defined by the implication

$$(i,j) \in M \quad \Leftrightarrow \quad \mathcal{P}_i(m(t)) = j \text{ and } \mathcal{P}_j(m(t)) = i,$$

for all $t \geq T$, and $t \mapsto (M, \alpha^b(t))$ converges to an ε-neighborhood of the set of Nash outcomes of $\mathcal{G}$.

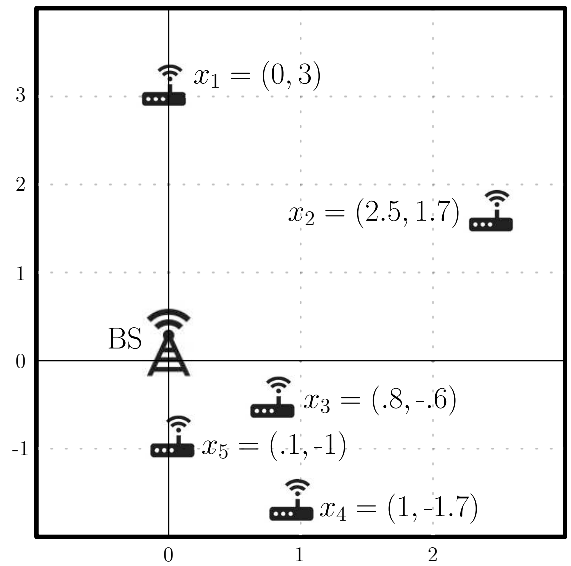


Figure 5.2: Spatial distribution of devices $\{1, \ldots, 5\}$ and the base station (BS).

device, i	1	2	3	4	5
% access, ρ_i	20%	28%	11%	21%	20%

one transmission period, T →

Figure 5.3: TDMA transmission time allocations for each device.

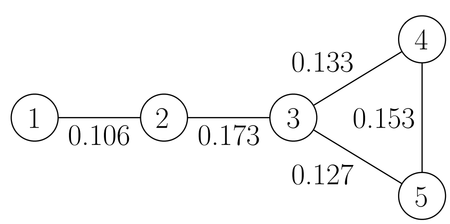


Figure 5.4: Bargaining graph resulting from the position and TDMA transmission time allocations for each device. Here, we have taken $P_{\max} = 3$.

5.5 Application to multi-user wireless communication

In this section, we provide some simulation results of our proposed Nash bargaining dynamics as applied to a multi-user wireless communication scenario.

The scenario we describe here is a simplified version of the one found in [90], and we direct the reader to that reference for a more detailed discussion on the model. We assume that there are $n = 5$ single antenna devices distributed spatially in an environment that send data to a fixed base station. We denote the position of device $i \in \{1, \ldots, 5\}$ as $x_i \in \mathbb{R}^2$ and we assume without loss of generality that the base station is located at the origin. Figure 5.2 illustrates the position of the devices. An individual device's transmission is managed using a time division multiple access (TDMA) protocol. That is, each device i is assigned a certain percentage ρ_i of a transmission period of length T in which it is allowed to transmit as specified in Figure 5.3. We use a commonly used model for the capacity $c_i > 0$ of the communication channel from device i to the base station, which is a function of their relative distance,

$$c_i = \log(1 + |x_i|^{-1}).$$

In the above, we have taken various physical parameters (such as transmit power constraints, path loss constants, and others) to be 1 for the sake of presentation. Since i only transmits for ρ_i percent of each transmission period, the effective capacity of the channel from device i to base station is $\rho_i c_i$. It is well-known in wireless communication [105] that multiple antenna devices can improve the channel capacity. Thus, devices i and j may decide to share their data and transmit a multiplexed data signal in both i and j's allocated time slots. In essence, i and j would behave as a single virtual 2-antenna device. The resulting channel capacity is given by

$$c_{i,j} = \log(1 + |x_i|^{-1} + |x_j|^{-1}),$$

which is greater than both c_i and c_j. However, there is a cost to agent i and j cooperating in this way because their data must be transmitted to each other. We assume that the device-to-device transmissions do not interfere with the device-to-base station transmissions. The power needed to transmit between i and j is given by

$$P_{i,j} = |x_i - x_j|.$$

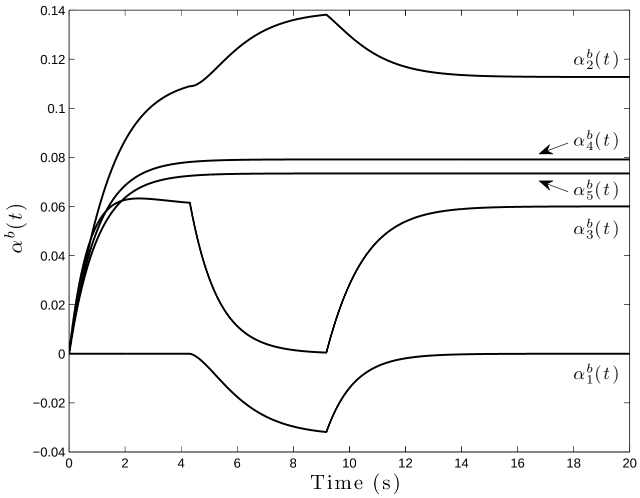


Figure 5.5: Evolution of each device's allocation in dynamics (5.12). At various times (i.e., $t \approx 4$ and 9), certain devices change who they identify as partners in the matching which explains the kinks in the trajectories at those times. This occurs because of the evolution of the matching states in Figure 5.6 and devices cannot correctly deduce the stable matching until $t \approx 9$.

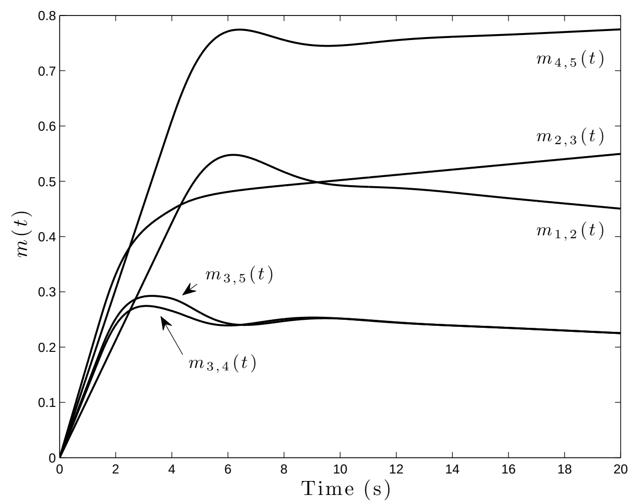


Figure 5.6: Evolution of neighboring device's matching states in dynamics (5.12). The final convergence of the matching states to $\{0,1\}^{|\mathcal{E}|}$ (which we do not show for the sake of presentation) takes much longer than devices need to accurately identify a Nash outcome.

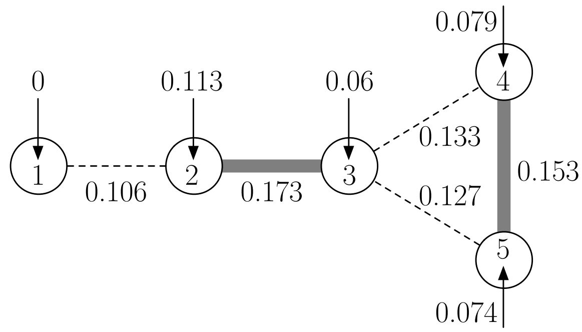


Figure 5.7: Nash outcome that is distributedly computed by devices. Device matchings are shown by thicker grey edges and allocations to each device are indicated with arrows

Table 5.1: Improvements in capacity due to collaboration

Device	Effective channel capacity without collaboration, c_i	Increase in effective channel capacity in Nash outcome, α_i^b	% improvement
1	0.288	0	0
2	0.288	0.113	39.2
3	0.693	0.06	8.7
4	0.693	0.079	11.4
5	0.406	0.074	18.2
{1,...,5}	0.441	0.070	15.8

If this power is larger than some $P_{\max} > 0$, then i and j will not share their data. We can model this scenario via a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$, where $\mathcal{V} = \{1, \ldots, 5\}$ are the devices, edges correspond to whether or not i and j are willing, based on the power requirements, to share their data

$$(i, j) \in \mathcal{E} \Leftrightarrow P_{i,j} \leq P_{\max},$$

and the edge weights represent the increase in effective channel capacity should devices cooperate,

$$w_{i,j} = (\rho_i + \rho_j)c_{i,j} - \rho_i c_i - \rho_j c_j, \quad \forall (i, j) \in \mathcal{E}.$$

Figure 5.4 shows this graph, using the data for the scenario we consider. It is interesting to note that, besides channel capacity and power constraints, one could incorporate other factors into the edge weight definition. For example, if privacy is a concern in the network, then devices may be less likely to share their data with untrustworthy devices which can be modeled by a smaller edge weight. A matching

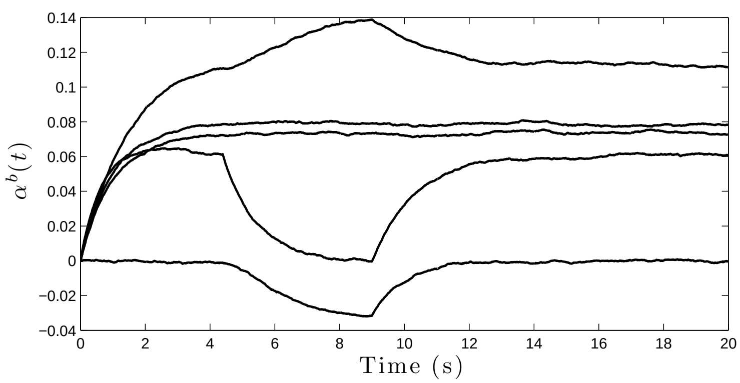


Figure 5.8: Trajectories of the noisy Nash dynamics. The noise is normally distributed with zero mean and standard deviation 0.01. Stable matchings are still correctly deduced and devices' allocations converge to a neighborhood of the allocations in the Nash outcome.

M in the context of this setting corresponds to disjoint pairs of devices that decide to share their data and transmission time slots in order to achieve a higher effective channel capacity. An allocation corresponds to how the resulting improved bit rate is divided between matched devices. For example, if i is allocated an amount of α_i^b, then i and j will transmit their data such that i's data reaches the base station at a rate of $c_i + \alpha_i^b$. The percent improvement in bit rate for i is then given by α_i^b/c_i. Devices use the dynamics (5.12) to find, in a distributed way, a Nash outcome for this problem. Figures 5.5 and 5.6 reveal the resulting state trajectories and Figure 5.7 displays the final Nash outcome. The percent improvements resulting from collaboration for each device are collected in Table 5.1. The last row in this table show that the network-wide improvement is 15.8%. Before bargaining, devices 1 and 2 have the lowest individual channel capacities and would thus greatly benefit from collaboration. However, due to power constraints, device 1 can only match with device 2, who in turn prefers to match with device 3. This explains why, in the end, device 1 is left unmatched. Figure 5.8 illustrates how convergence is still achieved when noise is present in the devices' dynamics, as forecasted by

Corollary 5.4.3.

This concludes our study of distributed bargaining in dyadic-exchange networks.

Chapter 5, in part, is a reprint of the material [80] as it appears in 'Distributed linear programming and bargaining in exchange networks' by D. Richert and J. Cortés in the proceedings of the 2013 American Control Conference as well as the material [84] as it appears in 'Distributed bargaining in dyadic-exchange networks' by D. Richert and J. Cortés which was submitted to the IEEE Transactions on Control of Network Systems. The dissertation author was the primary investigator and author of this paper and unpublished material.

Chapter 6

Cooperation inducing mechanisms in UAV formation pairs

Digging deeper into the bargaining problem in exchange networks, this chapter explores the logistical issues surrounding how agents may effectively realize the payoff that they were promised in a bargaining outcome. As an illustrative example, we consider the problem of optimally allocating the leader task between pairs of selfish unmanned aerial vehicles (UAVs) flying in formation. The UAV that follows the other achieves a fuel benefit due to a reduction in aerodynamic drag. We assume (e.g., using the Nash bargaining dynamics of Chapter 5) that a network of UAVs have autonomously agreed upon who to fly with in formation as well as a division of the combined fuel benefit of the formation. However, the non-cooperative nature of the agents makes it necessary to arbitrate leader-allocation mechanisms that induce collaboration. The aim of this chapter is the design of such mechanisms and algorithms that individual agents can use to compute them.

Upon modeling the UAV formation scenario, we find that such mechanisms only exist if UAVs are willing to forgo some cost gain, modeled by a parameter $\varepsilon \geq 0$, before breaking a formation. Restricting our search to only those leader allocations that induce cooperation, the result is an optimization problem with two components. On the one hand, given a fixed number of leader switches, the problem is to determine the optimal leader allocation and, on the other, the problem is to find the optimal number of leader switches. Both problems turn out to be nonconvex.

Nonconvex problems (see e.g. [40] and references therein) are widely considered the most challenging in the field of optimization. In particular, a unified theory on how to solve them does not exist. Nevertheless, a common approach is to convert the nonconvex problem into a convex one, for which efficient solution methods do exist [17, 23]. Such conversions can be performed by way of relaxations of the constraints or restrictions of the feasible set [1]. For the first problem of finding optimal leader allocations given a fixed number of leader switches, we employ the latter approach. This is achieved by first considering the case when switching the lead has no cost, allowing us to find the optimal value of the program. To this end, we design the COST REALIZATION ALGORITHM to determine an optimal cooperation-inducing leader allocation. Considering the more general case, when switching the lead is costly, we restrict the feasible set of leader allocations to mimic those of the solution provided by the COST REALIZATION ALGORITHM. Remarkably, the restriction convexifies the feasible set of the original nonconvex problem while maintaining its optimal value.

Regarding the second problem, our analysis reveals a quasiconvexity-like property of the optimal value of the problem as a function of the number of switches. This property allows us to design the BINARY SEARCH ALGORITHM, which finds the optimal number of leader switches in logarithmic time.

Several simulations throughout the chapter illustrate our results.

6.1 Problem setup

This section describes the problem setup. After introducing the notions of formation, lead distance, and cost-to-target function, we present the optimization problem we seek to solve. Consider a pair of UAVs with unique identifiers (UIDs) i and j evolving in $X \subset \mathbb{R}^3$. Both i and j have synchronized clocks and can communicate with each other. A superscript i (resp. j) denotes a quantity associated with i (resp. j). Agent i has position $x^i(t) \in X$ at time $t \in \mathbb{R}_{\geq 0}$, a target location $\bar{x}^i \in X$, and the objective of flying from origin $x^i(0)$ to target location while consuming the least amount of fuel. The same is valid for agent j. For UAVs flying in close proximity, the inter-agent distance between them is negligible compared to the total distance they must travel to their target. Therefore we make the abstraction that i and j are point masses that may have concurrent position.

6.1.1 Formations and lead distances

To move from origin to destination efficiently, agents i and j might decide to travel in formation. Here, we formally introduce this notion and examine the associated costs. Without loss of generality (via an appropriate change of coordinate frame), suppose that i and j have rendezvoused at the origin, $x_r = \mathbf{0}$, at time $t = 0$ and are flying in the direction $u = (1, 0, 0)$. Agents i and j are *in formation* at a time t if

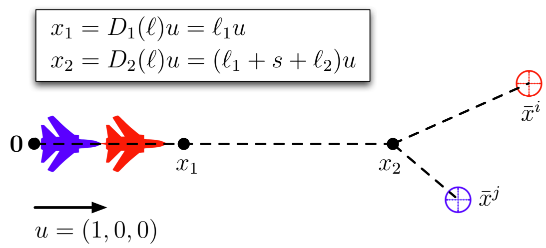
(i) $x^i(0) = x^j(0) = x_r$,

(ii) $[x^i(0), x^i(t)] = [x^j(0), x^j(t)] \in \mathrm{ray}(x_r, u)$,

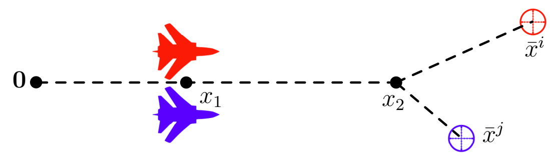
(iii) $d(x^i(\tau), x^j(\tau)) = 0$ for all $\tau \in [0, t]$,

where $\mathrm{ray}(x_r, u)$ is the ray originating at x_r in the direction of u and $d : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}_{\geq 0}$ is the Euclidean distance between two points. The execution of a formation is completely described by a *vector of lead distances* (VOLD) and the UID of the agent which leads first. Without loss of generality, let i lead the formation first. A VOLD $\ell \in \mathbb{R}^N_{\geq 0}$ is a finite-dimensional vector prescribing which UAV leads the formation when and for how long. For instance, i will initially lead the formation for distance ℓ_1 at which point i and j will switch the lead. Upon completion of the leader switch, j will lead the formation for distance ℓ_2. As such, for n odd (resp. even), ℓ_n is the n^{th} distance led by i (resp. j). We use N to denote the cardinality of a VOLD.

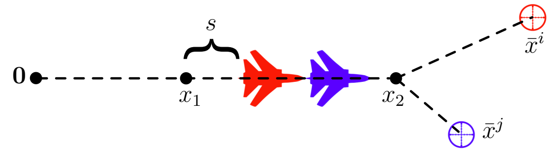
A *leader switch* is a maneuver which takes a distance s to complete, see Figure 6.1. During a leader switch, the fuel consumption per unit distance is $\Gamma > 1$, and hence, the fuel consumed by both UAVs is $s\Gamma$. We have scaled the quantity Γ relative to the fuel consumption per unit distance of leading the formation (which, by assumption, is 1). Conversely, flying in the wake of another UAV reduces the aerodynamic drag on the following UAV. Thus, the relative fuel cost per unit distance of a UAV following is $\gamma < 1$. Flying solo or leading the formation incur the same fuel consumption per unit distance. Upon completion (or breaking) of the formation, UAVs fly directly to their respective targets. For reasons of presentation, we assume that UAVs are identical in the sense that γ, Γ, and the cost per unit distance of flying solo are the same for all agents. However, the remaining analysis could easily be adapted for agents that are not identical.



(a) Red begins leading the formation



(b) A leader switch is initiated at x_1



(c) Blue leads after the switch

Figure 6.1: Example flight behavior of UAVs given a VOLD $\ell = (\ell_1, \ell_2)$. The dashed lines represent the proposed flight paths of the UAVs. During a switch, the UAVs deviate slightly from the formation heading and red (resp. blue) decreases (resp. increases) its speed. After maintaining the new speeds for distance s, the UAVs return to the original heading and speed of the formation. Both UAVs consume $s\Gamma$ amount of fuel in this maneuver. At x_2, the UAVs fly directly to their respective targets.

6.1.2 Cost-to-target functions

Here we define an agent's cost-to-target. To begin we introduce some auxiliary functions. Given a VOLD $\ell \in \mathbb{R}^N_{\geq 0}$, the distance of the nth switch from the origin is

$$D_n(\ell) = \begin{cases} 0, & n = 0, \\ \sum_{k=1}^{n} \ell_k + (n-1)s, & 1 \le n \le N. \end{cases}$$

The total distance of the formation prescribed by ℓ is $D_N(\ell)$. Likewise, given a VOLD ℓ, the number of leader switches that have been initiated when the UAVs have been in formation for distance $D \ge 0$ is

$$\#_{\mathrm{sw}}(\ell, D) = \max\{n \in \{0, 1, \ldots, N\} : D_n(\ell) \le D\},$$

and the distance from the last switch is $D_{\mathrm{LS}}(\ell, D) = D - D_{\#_{\mathrm{sw}}(\ell,D)}(\ell)$. Given a VOLD and a distance D, a UAV is able to compute the relative fuel consumed on its flight from x_r to its target if it were to break the formation at Du. We refer to it as the UAV's *cost-to-target* function. Formally, for agent i, we have $\mathtt{ct}^i : \mathbb{R}^N_{\ge 0} \times \mathbb{R}_{\ge 0} \to \mathbb{R}_{>0}$ given by

$$\mathtt{ct}^i(\ell, D) = \sum_{k \in \mathbb{N} \cap \mathbb{O}}^{\#_{\mathrm{sw}}(\ell,D)} \ell_k + s\Gamma(\#_{\mathrm{sw}}(\ell, D) - 1)_+ + \gamma \sum_{k \in \mathbb{N} \cap \mathbb{E}}^{\#_{\mathrm{sw}}(\ell,D)} \ell_k + R^i(\ell, D) + d(Du, \bar{x}^i).$$

The first term is the fuel consumed by leading, the second term is the fuel consumed due to switching the lead, and the third term is the fuel consumed while following. The fourth term is a residual term accounting for the fuel consumed since the last switch. For $\#_{\mathrm{sw}}(\ell, D)$ odd

$$R^i(\ell, D) = \gamma(D_{\mathrm{LS}}(\ell, D) - s)_+ + \Gamma \min\{D_{\mathrm{LS}}(\ell, D), s\},$$

and for $\#_{\mathrm{sw}}(\ell, D)$ even

$$R^i(\ell, D) = (D_{\mathrm{LS}}(\ell, D) - s)_+ + \Gamma \min\{D_{\mathrm{LS}}(\ell, D), s\}.$$

Lastly, the $d(Du, \bar{x}^i)$ is the fuel that i would consume by breaking the formation at Du and flying to its target. Slight variations of a UAV's cost-to-target are the cost-to-target-at-the-kth-switch functions. For $k = 1, \ldots, N$, these are given by $\mathtt{ct}^i_k : \mathbb{R}^k_{\ge 0} \to \mathbb{R}_{>0}$ defined as

$$\mathtt{ct}^i_k(\ell_1, \ldots, \ell_k) = \sum_{n \in \mathbb{N} \cap \mathbb{O}}^{k} \ell_n + \gamma \sum_{n \in \mathbb{N} \cap \mathbb{E}}^{k} \ell_n + s\Gamma(k-1) + d\bigg(\Big(\sum_{n=1}^{k} \ell_n + s(k-1)\Big)u, \bar{x}^i\bigg).$$

By construction, $\mathtt{ct}^i(\ell, D_k(\ell)) \equiv \mathtt{ct}^i_k(\ell_1, \dots, \ell_k)$. Analogous $\mathtt{ct}^j$ and $\mathtt{ct}^j_k$ exist for j. In addition to an individual UAV's cost-to-target, it is possible to characterize the combined cost-to-targets of i and j at the end of their formation in terms of the formation breakaway location. To do so, consider any $c^i, c^j \in \mathbb{R}_{>0}$ and suppose there exists an $\ell \in \mathbb{R}^N_{\geq 0}$ such that $c^i = \mathtt{ct}^i_N(\ell)$ and $c^j = \mathtt{ct}^j_N(\ell)$ (i.e., the final cost-to-target at the end of the formation for i and j are c^i and c^j respectively). The UAVs' combined cost-to-targets at the end of the formation is

$$c^j + c^i = \mathtt{ct}^j_N(\ell) + \mathtt{ct}^i_N(\ell).$$

Under a change of variables $L = \sum_{k=1}^N \ell_k$ this becomes

$$\begin{aligned} c^j + c^i = \mathtt{ct}^{i+j}_N(L) := (1+\gamma)L + 2(N-1)s\Gamma \\ + d\big((L + s(N-1))u, \bar{x}^j\big) + d\big((L + s(N-1))u, \bar{x}^i\big). \end{aligned}$$

We call $\mathtt{ct}^{i+j}_N : \mathbb{R}_{\geq 0} \to \mathbb{R}_{>0}$ the *combined cost-to-target* function of a formation with $N-1$ switches. Note that the formation breakaway distance is $L + s(N-1)$.

6.1.3 Problem statement

Upon arrival at the rendezvous location x_r, the agents need to determine a VOLD to dictate how to execute their formation. Suppose i declares an upper bound C^i on its final cost-to-target. Then j would propose a VOLD which solves the following two-stage optimization problem. First, among VOLDs with a fixed cardinality N, the minimum cost-to-target j can expect is

$$\min_{\ell \in \mathbb{R}^N_{\geq 0}} \mathtt{ct}^j_N(\ell) \tag{6.1a}$$

$$\text{s.t.} \quad \mathtt{ct}^i_N(\ell) \leq C^i, \tag{6.1b}$$

$$\mathtt{ct}^i_N(\ell) \leq \mathtt{ct}^i(\ell, D) + \varepsilon^i, \forall D \in [0, D_N(\ell)], \tag{6.1c}$$

$$\mathtt{ct}^j_N(\ell) \leq \mathtt{ct}^j(\ell, D) + \varepsilon^j, \forall D \in [0, D_N(\ell)]. \tag{6.1d}$$

The parameters $\varepsilon^i, \varepsilon^j \geq 0$, intrinsic to each UAV, model their degree of cooperation (see Remark 6.1.1). Constraint (6.1c) ensures that at no point in the formation will i's cost-to-target be ε^i less than its final cost-to-target. The assumption is that i would break the formation earlier if it could benefit (by more than ε^i) in doing so. An analogous reason for j motivates (6.1d). The optimal value of (6.1) is denoted $C^j(N)$. Next, among all VOLDs (of any number of leader switches), the minimum cost-to-target that j could expect is

$$\min_{N \in \mathbb{N}_{\geq 2}} \quad C^j(N). \tag{6.2}$$

If N^* is minimizes (6.2) and ℓ^* minimizes (6.1) for fixed N^*, then j would propose ℓ^* to i. For reasons of notation, let $\mathcal{F}(N)$ be the feasible set of (6.1). To relate this problem setup to the network bargaining of Chapter 5, the transferable utility between i and j would be

$$w_{i,j} = d(x_r, \bar{x}^i) + d(x_r, \bar{x}^j) - C^i - \mathtt{ct}^j_{N^*}(\ell^*),$$

and the mechanism to ensure that agents realize their allocation of $w_{i,j}$ is the vector of lead distances prescribed by ℓ^*.

In general, $\mathcal{F}(N)$ is nonconvex and (6.2) is combinatorial. We devote much of this chapter to transforming (6.1) into a convex problem (i.e., a convex objective function minimized over a convex set) and developing tools to efficiently solve (6.2).

From this point on, we assume that the formation heading is not in the same direction as either agents' target,

$$\bar{x}^i \notin \mathrm{ray}(x_r, u) \quad \text{and/or} \quad \bar{x}^j \notin \mathrm{ray}(x_r, u). \tag{6.3}$$

Without this assumption, inducing cooperation between i and j is trivial: if a UAV cannot "breakaway" from the formation, the other UAV will just follow in the formation until it no longer benefits from doing so.

Remark 6.1.1. (Selfish vs. fully cooperative UAVs). If $\varepsilon^i = \varepsilon^j = 0$, constraints (6.1c)-(6.1d) imply that i and j only abide by VOLDs for which their cost-to-target at any time in formation is never better than their cost-to-target at the end of the formation. We call such UAVs *selfish*. For selfish UAVs, the solution to problem (6.1) is trivially $\ell^* = \mathbf{0} \in \mathbb{R}^N$ for any N (agents never fly in formation). This is because neither UAV is willing to be the last to lead the formation compared to flying straight to its target. On the other hand, removing (6.1c)-(6.1d) (equivalently, setting $\varepsilon^i = \varepsilon^j = \infty$) implies that UAVs will abide by any VOLD $\ell \in \mathbb{R}^N_{\geq 0}$. We call such UAVs *fully cooperative*. However, these UAVs could potentially save fuel by breaking the formation earlier. This discussion motivates our problem formulation, which accounts for agents who are selfishly motivated yet willing to forfeit a small amount of fuel to ensure the formation occurs. Figure 6.2 shows the dependency of C^j_* on $\varepsilon^i, \varepsilon^j$. •

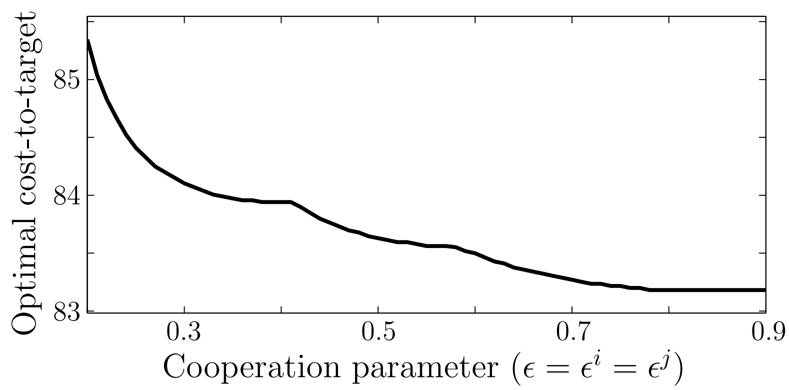


Figure 6.2: Optimal value of (6.2) with respect to $\varepsilon = \varepsilon^i = \varepsilon^j$. As ε increases, j's optimal cost-to-target, C^j_* decreases. The nonsmoothness at $\varepsilon \approx 0.41$ and $\varepsilon \approx 0.58$ is due to decreases in the optimal number of leader switches in the solution to (6.2). The simulation data are: $C^i = 82$, $s = 0.2$, $\Gamma = 1.7$, $\gamma = 0.5$, $\bar{x}^i = (100, 10)$, $\bar{x}^j = (90, -20)$.

6.2 Unveiling the structure of optimal VOLDs

This section describes properties of the cost-to-target functions and of the solutions to (6.1). Using them, we provide a more explicit description of the feasible set, allowing us to express (6.1) in standard form. Without loss of generality, many of the results only refer to i.

6.2.1 Properties of the cost-to-target functions

A cost-to-target function is continuous and piecewise differentiable with respect to distance D. In particular, $\texttt{ct}^i$ is not differentiable at distances where leader switches are initiated or completed. That is, $\partial_D \texttt{ct}^i$ exists at (ℓ, D) iff

$D_{\text{LS}}(\ell, D) \notin \{0, s\}$. The following reveals a useful convexity-like property of the cost-to-target functions.

Lemma 6.2.1. (Leading, following, and switching become more costly as the formation progresses). *Let ℓ be a VOLD and $D_1 < D_2$. Suppose that, under ℓ, UAV i is leading (or following, or switching) at both $D_1 u$ and $D_2 u$. Then $\partial_D \texttt{ct}^i(\ell, D_1) < \partial_D \texttt{ct}^i(\ell, D_2)$. Moreover, if i is leading or switching at $D_1 u$, then $\partial_D \texttt{ct}^i(\ell, D_1) > 0$.*

Proof. The derivative of $\texttt{ct}^i$ with respect to D is

$$\partial_D \texttt{ct}^i(\ell, D) = \partial_D d(Du, \bar{x}^i) + \begin{cases} \gamma, & \text{if } i \text{ follows at } Du, \\ 1, & \text{if } i \text{ leads at } Du, \\ \Gamma, & \text{otherwise.} \end{cases}$$

The function $D \mapsto d(Du, \bar{x}^i)$ is strictly convex under (6.3). Thus, $\partial_D d(Du, \bar{x}^i)$ is strictly increasing. Suppose that i is leading at both $D_1 u$ and $D_2 u$. Then

$$\begin{aligned} \partial_D \texttt{ct}^i(\ell, D_1) &= \partial_D d(D_1 u, \bar{x}^i) + 1, \\ &< \partial_D d(D_2 u, \bar{x}^i) + 1 = \partial_D \texttt{ct}^i(\ell, D_2). \end{aligned}$$

Similar analysis holds for when i is following (or switching) at both $D_1 u$ and $D_2 u$. To show that $\partial_D \texttt{ct}^i(\ell, D_1) > 0$ when i is leading at $D_1 u$, let $a > 0$ be sufficiently small such that i is also leading at $(D_1 - a)u$. Then

$$\texttt{ct}^i(\ell, D_1) = \texttt{ct}^i(\ell, D_1 - a) - d((D_1 - a)u, \bar{x}^i) + a + d(D_1 u, \bar{x}^i) > \texttt{ct}^i(\ell, D_1 - a),$$

where we have used the triangle inequality. Since a can be taken arbitrarily small, $\partial_D \texttt{ct}^i(\ell, D_1) > 0$ follows. If i is switching at $D_1 u$, the above argument with Γa instead of a, together with $\Gamma > 1$, yields the same conclusion. □

Roughly speaking, Lemma 6.2.1 states that it is more costly to lead (or follow or switch) in the formation as it progresses. The last statement in Lemma 6.2.1 simply states that leading or switching is always costly. This is to be distinguished from following which, as we show later, decreases the cost-to-target function in an optimal VOLD. Using a similar argument as in the proof of Lemma 6.2.1, the following states some properties of the cost-to-target-at-the-kth-switch functions.

Lemma 6.2.2. (Properties of the cost-to-target-at-the-kth-switch functions). *For* $\ell \in \mathbb{R}^N_{\geq 0}$

(P1) $\partial_{\ell_1} \mathtt{ct}^i_k(\ell_1, \ldots, \ell_k) > 0$, *for* $k \geq 1$,

(P2) $\partial_{\ell_2} \mathtt{ct}^j_k(\ell_1, \ldots, \ell_k) > 0$, *for* $k \geq 2$,

(P3) $\partial_{\ell_n} \mathtt{ct}^m_k(\ell_1, \ldots, \ell_k) = \partial_{\ell_{n+2}} \mathtt{ct}^m_k(\ell_1, \ldots, \ell_k)$, *for* $k \geq n+2$ *and* $m = i, j$,

(P4) $\partial_{\ell_n} \mathtt{ct}^m_k(\ell_1, \ldots, \ell_k) < \partial_{\ell_n} \mathtt{ct}^m_{k+2}(\ell_1, \ldots, \ell_k)$, *for* $k \geq n$ *and* $m = i, j$.

6.2.2 Properties of the optimal VOLDs

This section explores an important property of the breakaway distance prescribed by a solution to (6.1). For some $\ell \in \mathbb{R}_{\geq 0}$ let $c^j = \mathtt{ct}^j_N(\ell)$ and $c^i = \mathtt{ct}^i_N(\ell)$. Recalling the discussion on the combined cost-to-target function (cf. Section 6.1.2), the possible breakaway locations of the formation can be described by all L satisfying

$$c^j + c^i = \mathtt{ct}^{i+j}_N(L).$$

Since $\mathtt{ct}^{i+j}_N$ is strictly convex, there exist two solutions L_1, L_2 to this equation (note that L_1, L_2 may not be distinct). Letting $L^*_N = \operatorname{argmin}_L \mathtt{ct}^{i+j}_N(L)$, we assume without loss of generality that $L_1 \leq L^*_N \leq L_2$. The following result states that $L_1 + s(N-1)$, and not $L_2 + s(N-1)$, is the breakaway location for a solution to (6.1).

Proposition 6.2.3. (UAVs breakaway as soon as possible). *For* $N \in \mathbb{N}$*, let* ℓ^* *be a solution to* (6.1)*. Then,*

$$\ell^* \in \mathbb{L}_N := \left\{ \ell \in \mathbb{R}^N_{\geq 0} : \sum_{k=1}^{N} \ell_k \leq L^*_N \right\}.$$

Proof. Consider the case when N is even so that j leads the last segment. Proving the result by contradiction, suppose ℓ solves (6.1) but $\sum_{k=1}^N \ell_k = \hat{L} > L^*_N$. Since $\mathtt{ct}^{i+j}_N$ is strictly convex, we know $\partial_L \mathtt{ct}^{i+j}_N(\hat{L}) > 0$. That is

$$\gamma + \partial_L d((\hat{L} + s(N-1))u, \bar{x}^i) > -1 - \partial_L d((\hat{L} + s(N-1))u, \bar{x}^j). \tag{6.4}$$

In other words, $\partial_{\ell_N}\mathtt{ct}^i_N(\ell) > -\partial_{\ell_N}\mathtt{ct}^j_N(\ell)$. A rearrangement of (6.4) reveals that $\partial_{\ell_{N-1}}\mathtt{ct}^j_N(\ell) > -\partial_{\ell_{N-1}}\mathtt{ct}^i_N(\ell)$ also. Consider the alternative VOLD $\ell' \in \mathbb{R}^N_{\geq 0}$ where, for some $a, b \geq 0$ to be designed, $\ell'_{N-1} = \ell_{N-1} - a, \ell'_N = \ell_N - b$, and $\ell'_k = \ell_k$ otherwise. To reach a contradiction of ℓ being a solution of (6.1), we want to design a, b such that $\mathtt{ct}^i_N(\ell') \leq \mathtt{ct}^i_N(\ell)$, $\mathtt{ct}^j_N(\ell') < \mathtt{ct}^j_N(\ell)$ and ℓ' satisfies (6.1c)-(6.1d). First, let a, b be sufficiently small such that the following linear approximations are valid. We desire

$$a\partial_{\ell_{N-1}}\mathtt{ct}^j_N(\ell) + b\partial_{\ell_N}\mathtt{ct}^j_N(\ell) > 0, \tag{6.5a}$$

$$a\partial_{\ell_{N-1}}\mathtt{ct}^i_N(\ell) + b\partial_{\ell_N}\mathtt{ct}^i_N(\ell) \geq 0. \tag{6.5b}$$

There are four cases: (i) $\partial_{\ell_N}\mathtt{ct}^i_N(\ell), \partial_{\ell_{N-1}}\mathtt{ct}^j_N(\ell) > 0$ (ii) $\partial_{\ell_N}\mathtt{ct}^i_N(\ell) < 0 < \partial_{\ell_{N-1}}\mathtt{ct}^j_N(\ell)$ (iii) $\partial_{\ell_{N-1}}\mathtt{ct}^j_N(\ell) < 0 < \partial_{\ell_N}\mathtt{ct}^i_N(\ell)$ (iv) $\partial_{\ell_N}\mathtt{ct}^i_N(\ell), \partial_{\ell_{N-1}}\mathtt{ct}^j_N(\ell) < 0$. For the sake of space, consider only case (iv) which we claim is the most complex. Combining (6.5) yields

$$-a\frac{\partial_{\ell_{N-1}}\mathtt{ct}^i_N(\ell)}{\partial_{\ell_N}\mathtt{ct}^i_N(\ell)} < b < -a\frac{\partial_{\ell_{N-1}}\mathtt{ct}^j_N(\ell)}{\partial_{\ell_N}\mathtt{ct}^j_N(\ell)}.$$

Given $a > 0$, $\exists b > 0$ since $\partial_{\ell_{N-1}}\mathtt{ct}^j_N(\ell) > -\partial_{\ell_{N-1}}\mathtt{ct}^i_N(\ell)$ and $\partial_{\ell_N}\mathtt{ct}^i_N(\ell) > -\partial_{\ell_N}\mathtt{ct}^j_N(\ell)$. It remains to show that ℓ' satisfies (6.1c)-(6.1d). Consider first the interval $D \in [0, D_{N-1}(\ell')]$ where, by construction of ℓ', $\mathtt{ct}^i(\ell', D) = \mathtt{ct}^i(\ell, D)$. Since ℓ satisfies (6.1c)-(6.1d) on this interval

$$\mathtt{ct}^i(\ell', D) \geq \mathtt{ct}^i_N(\ell) - \varepsilon^i > \mathtt{ct}^i_N(\ell') - \varepsilon^i.$$

Similar analysis holds for j (i.e., cooperation is induced up until $D_{N-1}(\ell')u$). Beyond $D_{N-1}(\ell')u$, UAV j is switching and leading. So, $\forall D \in (D_{N-1}(\ell'), D_N(\ell')]$

$$\mathtt{ct}^j(\ell', D) > \mathtt{ct}^j_{N-1}(\ell') > \mathtt{ct}^j_N(\ell') - \varepsilon^j,$$

due to Lemma 6.2.1. Thus, j will cooperate under ℓ'. Still under case (iv), note that $\partial_{\ell_N}\mathtt{ct}^i_N(\ell) < 0 \Rightarrow \partial^-_D\mathtt{ct}^i(\ell, D_N(\ell)) < 0 \Rightarrow \partial^-_D\mathtt{ct}^i(\ell', D_N(\ell')) < 0$ which, due to Lemma 6.2.1, further implies that following always decreases i's cost-to-target function in the formation. Agent i only switches and follows beyond $D_{N-1}(\ell')u$. So, $\forall D \in (D_{N-1}(\ell'), D_{N-1}(\ell') + s]$

$$\mathtt{ct}^i(\ell', D) > \mathtt{ct}^i(\ell', D_{N-1}(\ell')) > \mathtt{ct}^i(\ell', D_N(\ell')) - \varepsilon^i,$$

and $\forall D \in [D_{N-1}(\ell') + s, D_N(\ell'))$

$$\mathtt{ct}^i(\ell', D) > \mathtt{ct}^i(\ell', D_N(\ell')) > \mathtt{ct}^i(\ell', D_N(\ell')) - \varepsilon^i.$$

Thus, i will also cooperate (i.e., ℓ' satisfies (6.1c)-(6.1d)). Similar arguments hold for cases (i)-(iii). In summary, ℓ' decreases (6.1a) while satisfying (6.1d) which contradicts ℓ solving (6.1). N odd is dealt with analogously. □

Proposition 6.2.3 gives an upper bound on the breakaway distance of an optimal VOLD. Thus, we can restrict the feasible set of (6.1) to $\ell \in \mathcal{F}(N) \cap \mathbb{L}_N$. Following this result, for $\ell \in \mathbb{L}_N$, one has the additional property that a UAV's cost-to-target strictly decreases while following.

Corollary 6.2.4. (Following is beneficial). *If $\ell \in \mathbb{L}_N$ has i following at $\hat{D}u$, $\partial_D \mathtt{ct}^i(\ell, \hat{D}) < -\partial_D \mathtt{ct}^j(\ell, \hat{D}) < 0$.*

Proof. Following the proof of Proposition 6.2.3, if $\ell \in \mathbb{L}_N$ then we arrive at (6.4) with the inequality reversed. However, the LHS of (6.4) is $\partial_D \mathtt{ct}^i(\ell, \hat{D})$ where $\hat{D} = \hat{L}$. The result follows from applying Lemma 6.2.1 □

Corollary 6.2.4 also allows us to identify additional properties of the cost-to-target-at-the-k^{th}-switch functions.

Lemma 6.2.5. (Properties of the cost-to-target-at-the-kth-switch functions - continued). *For $\ell \in \mathbb{L}_N$,*

(P5) $\partial_{\ell_1} \mathtt{ct}_k^j(\ell_1, \ldots, \ell_k) \leq -\partial_{\ell_1} \mathtt{ct}_k^i(\ell_1, \ldots, \ell_k) < 0$, $k \geq 2$,

(P6) $\partial_{\ell_2} \mathtt{ct}_k^i(\ell_1, \ldots, \ell_k) \leq -\partial_{\ell_2} \mathtt{ct}_k^j(\ell_1, \ldots, \ell_k) < 0$, $k \geq 2$.

The results thus far are now used to state a fact about the final cost-to-target for UAV i given a solution to (6.1).

Lemma 6.2.6. (i receives its bound on final cost-to-target). *If ℓ is a solution to* (6.1) *then* $C^i = \mathtt{ct}_N^i(\ell)$.

Proof. The proof is by contradiction, so let ℓ solve (6.1) and assume $C^i < \mathtt{ct}^i_N(\ell)$. Decrease ℓ_2 by some amount $a > 0$, thus increasing i's cost-to-target and decreasing j's cost-to-target (cf. Lemma 6.2.5(P6)). For a sufficiently small, (6.1b) is still satisfied and (6.1a) decreases. Also (6.1c)-(6.1d) are satisfied, as shown by repeated application of Lemma 6.2.2(P4). Thus, we have reached a contradiction (i.e., if $C^i < \mathtt{ct}^i_N(\ell)$ then ℓ does not solve (6.1)). □

6.2.3 Equivalent formulation

Here, we combine the results established above to reduce the feasibility set of (6.1) to only those VOLDs exhibiting properties of optimal VOLDs. In particular, Lemma 6.2.1 and Corollary 6.2.4 reveal that, for $\ell \in \mathbb{L}_N$, the local minima of $\mathtt{ct}^i$ and $\mathtt{ct}^j$ occur at the distances where an agent initiates a switch from following to leading. So, if cooperation is induced at those points, then cooperation is induced for the entire formation. Additionally, we can now fix i's final cost-to-target at C^i. To summarize, we reformulate (6.1) in terms of the cost-to-target-at-the-kth-switch functions as follows. For fixed $N \in \mathbb{N}$

$$\min_{\ell \in \mathbb{L}_N} \quad \mathtt{ct}^j_N(\ell) \tag{6.6a}$$

$$\text{s.t. } c^j = \mathtt{ct}^j_N(\ell), \quad C^i = \mathtt{ct}^i_N(\ell), \tag{6.6b}$$

$$c^j \leq \mathtt{ct}^j_k(\ell_1, \ldots, \ell_k) + \varepsilon^j, \ \ k \in \mathbb{O}_{[1,N-1]}, \tag{6.6c}$$

$$C^i \leq \mathtt{ct}^i_k(\ell_1, \ldots, \ell_k) + \varepsilon^i, \ \ k \in \mathbb{E}_{[2,N-1]}. \tag{6.6d}$$

Given the above discussion, the set of solutions to (6.6) is the set of solutions to (6.1). The equality constraints in (6.6) are not affine and substituting them into the inequality constraints yields nonconvex inequality constraints.

Remark 6.2.7. (Total lead distance functions). For $\ell \in \mathbb{L}_N$ satisfying (6.6b)-(6.6d), one can show that, without knowing the specific elements of ℓ, there exists a unique distance that i must lead the formation. This is

$$\mathtt{tl}^i_N(c^j) := \big[L + d((L + (N-1)s)u, \bar{x}^j) \\ + (N-1)s\Gamma - c^j\big]/(1-\gamma) \equiv \sum_{k \in \mathbb{N} \cap \mathbb{O}}^{N} \ell_k,$$

where L satisfies $c^j + C^i = \mathtt{ct}_N^{i+j}(L)$ ($\mathtt{tl}_N^i$ is well-defined since L is unique). Also, $\mathtt{tl}_N^j(c^j) := L - \mathtt{tl}_N^i(c^j)$. •

6.3 Optimal VOLDs under no-cost switching

This section solves problem (6.2) when switching the lead does not incur a cost to UAVs (i.e., $s = s\Gamma = 0$). We start by characterizing the optimal value C_*^j and then design the COST REALIZATION ALGORITHM to generate a VOLD that realizes the optimal fuel consumption of UAV j in the formation. Note that, under no-cost switching

$$\mathtt{ct}_{N'}^{i+j}(L) = \mathtt{ct}_{N''}^{i+j}(L), \quad \forall N', N'' \in \mathbb{N}.$$

This can be interpreted as follows. Given a fixed breakaway location (in this case, L because $s = 0$) and i's cost-to-target, the final cost-to-target for j is independent of the number of leader switches in the VOLD. Based on this observation, we are able to prove the following.

Theorem 6.3.1. (Optimal value under no-cost switching). *For $s = 0$, $\max\{\varepsilon^i, \varepsilon^j\} > 0$, and any $N \geq 2$, C_*^j is the optimal value of the convex problem*

$$\min_L \{\mathtt{ct}_N^{i+j}(L) - C^i\}. \tag{6.7}$$

Proof. The proof is constructive. For $s = 0$, let $c^j = \min_L \mathtt{ct}_N^{i+j}(L) - C^i$ for any $N \geq 2$ and for brevity, let $\ell_L^i = \mathtt{tl}_N^i(c^j)$ and $\ell_L^j = \mathtt{tl}_N^j(c^j)$ (cf. Remark 6.2.7). Begin with the VOLD $\ell = (\ell_L^i, \ell_L^j)$. If ℓ is not feasible then it must be that $\mathtt{ct}_1^j(\ell_L^i) < c^j - \varepsilon^j$. By assumption, $\mathtt{ct}_1^j(0) > c^j - \varepsilon^j$. Therefore, by the intermediate value theorem, there exists a $\ell_1 \in (0, \ell_L^i)$ such that $\mathtt{ct}_1^j(\ell_1) = c^j - \varepsilon^j$. With a slight abuse of notation, let $\ell = (\ell_1, \ell_L^j, \ell_L^i - \ell_1)$. Then, because the breakaway distance has been preserved, i and j still realize their costs of C^i and c^j, respectively. Again, if ℓ is not feasible, then it must be that $\mathtt{ct}_2^i(\ell_1, \ell_L^j) < C^i - \varepsilon^i$ and, by the intermediate value theorem, there exists a $\ell_2 \in (0, \ell_L^j)$ such that $\mathtt{ct}_2^i(\ell_1, \ell_2) = C^i - \varepsilon^i$. Then, update $\ell = (\ell_1, \ell_2, \ell_L^i - \ell_1, \ell_L^j - \ell_2)$. This process may be repeated as long as ℓ is not feasible. If it never happens that ℓ is feasible, this implies that $\ell_k \to 0$ (we view $\{\ell_k\}$ as a sequence which is bounded and monotonic). This further implies that there exists a $L := \sum_{k=1}^{\infty} \ell_k$ such that $c^j - \varepsilon^j + C^i - \varepsilon^i = \mathtt{ct}_N^{i+j}(L)$. However, if $\max\{\varepsilon^i, \varepsilon^j\} > 0$, this contradicts c^j being the optimal value of (6.7). Therefore, it must be that for some finite number of steps, ℓ becomes feasible under the proposed procedure. $\square$

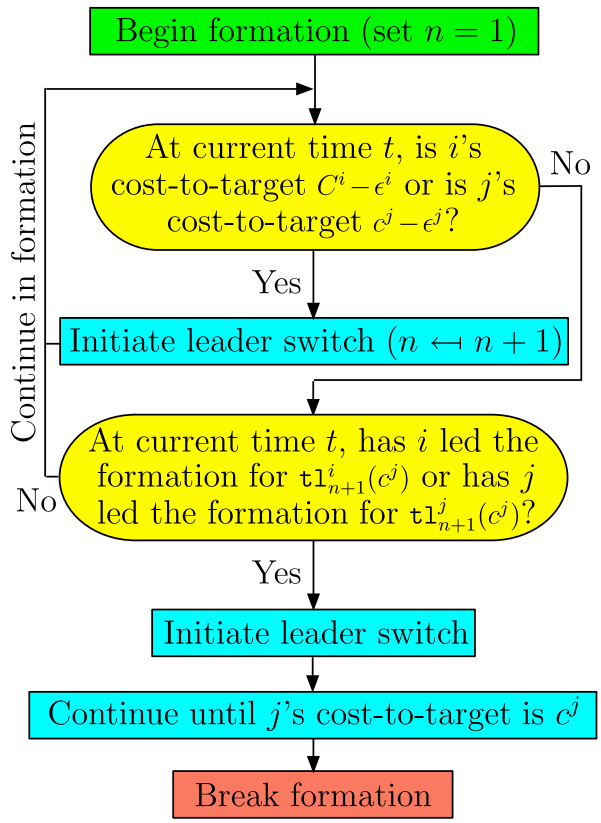


Figure 6.3: The COST REALIZATION ALGORITHM, with input c^j, over ε-cooperative agents i and j. Upon breaking the formation, UAVs fly directly to their respective targets. UAVs have knowledge of the following parameters when implementing the algorithm: C^i, s, Γ, γ, ε^i, ε^j, $\bar{x}^i$, $\bar{x}^j$.

The above result establishes that the optimal value of (6.2) under no-cost switching can be found as the optimal value of a simple convex problem. This result is useful for j as it is able to know a priori what final cost-to-target it can expect from a formation with i. However, i and j still do not know how to realize these cost-to-targets. The COST REALIZATION ALGORITHM provided in Figure 6.3

resolves this issue. Its design is inspired by the constructive proof of Theorem 6.3.1. Agents i and j implement this algorithm on-the-fly while in formation and only require knowledge of the optimal value to (6.2).

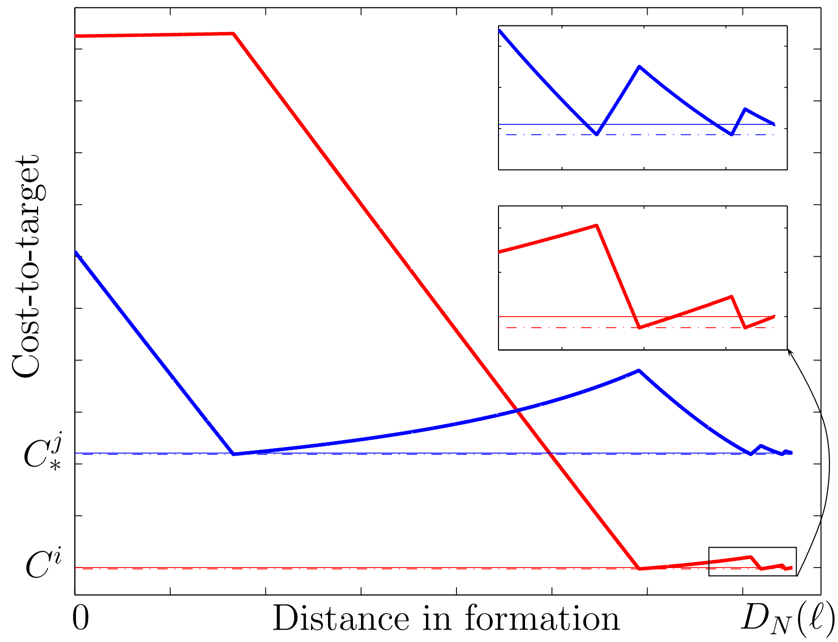
Corollary 6.3.2. (Inducing optimal solutions: no-cost switching). *For $s\Gamma = 0$ and input C_*^j, the* COST REALIZATION ALGORITHM *induces a VOLD that solves* (6.2)*.*

Figure 6.4 reports the cost-to-targets in a simulation of two UAVs flying from origin to target locations while implementing the COST REALIZATION ALGORITHM. A leader switch is indicated when an agents' cost-to-target transitions from increasing to decreasing (or vice versa). As one can see, the COST REALIZATION ALGORITHM schedules a leader switch whenever one of the agents' cost-to-target reaches ε below the projected final cost-to-target.

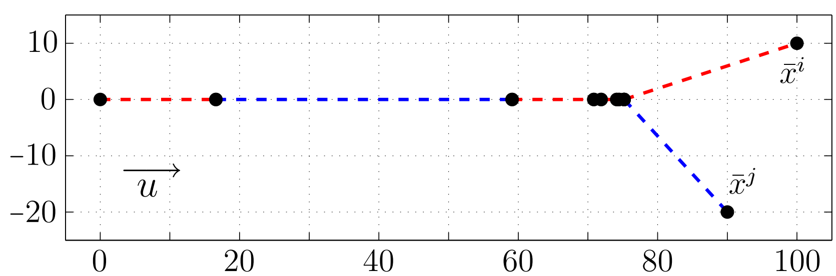
Remark 6.3.3. (Robustness of the COST REALIZATION ALGORITHM**).** Small measurement, modeling (i.e., unmodeled wind effects), and computational uncertainties result in small perturbations to an agent's final cost-to-target resulting from the COST REALIZATION ALGORITHM. Thus, for ε^i (resp. ε^j) sufficiently large, i (resp. j) is willing to remain in formation despite these perturbations to its expected final cost-to-target. In this sense, the parameters ε^i and ε^j ensure that the COST REALIZATION ALGORITHM is robust to small uncertainties. •

6.4 Optimal VOLDs under costly switching

This section solves problem (6.2) when switching is costly. The key difference with respect to Section 6.3 is that, under no-cost switching, whenever an inequality constraint in (6.6) becomes active, agents can initiate a leader switch to ensure cooperation is maintained without affecting their final cost-to-targets. However, under costly switching, the same logic does not hold because adding a leader switch increases the final cost-to-target of both agents.



(a) Cost-to-targets under no-cost switching



(b) Switch locations under no-cost switching.

Figure 6.4: Execution of the COST REALIZATION ALGORITHM under no-cost switching with input $C^j_* = 84$. (a) shows the cost-to-targets for i (red) and j (blue) resulting from the induced VOLD. Horizontal lines are final cost-to-targets and dash-dot lines are $\varepsilon^i = \varepsilon^j$ below that. (b) shows the actual flight paths: a red (resp. blue) dotted line is a segment on which i (resp. j) leads. Simulation data are $C^i = 80$,$\gamma = 0.5$, $\varepsilon^i = \varepsilon^j = 0.05$, $\bar{x}^i = (100, 10)$, $\bar{x}^j = (90, -20)$. The cost-benefits of the formation are $d(x_r, \bar{x}^i) - C^i = 20$ (20%) and $d(x_r, \bar{x}^j) - C^j_* = 8$ (8.7%) for i and j, resp.

6.4.1 Convex restriction

We start by restricting the feasible set of (6.6) to VOLDs that exhibit the same structure as in the no-cost case (i.e., equality constraints for $k = 1, \ldots, N-2$).

That is,

$$\min_{c^j} \quad c^j \tag{6.8a}$$

$$\text{s.t. } c^j = \mathtt{ct}_k^j(\ell_1, \ldots, \ell_k) + \varepsilon^j, \;\; k \in \mathbb{O}_{[1,N-2]}, \tag{6.8b}$$

$$C^i = \mathtt{ct}_k^i(\ell_1, \ldots, \ell_k) + \varepsilon^i, \;\; k \in \mathbb{E}_{[2,N-2]}, \tag{6.8c}$$

$$c^j = \mathtt{ct}_N^j(\ell_1, \ldots, \ell_N), \tag{6.8d}$$

$$C^i = \mathtt{ct}_N^i(\ell_1, \ldots, \ell_N), \tag{6.8e}$$

$$\ell \in \mathbb{L}_N, \tag{6.8f}$$

$$c^j \leq \mathtt{ct}_{N-1}^j(\ell_1, \ldots, \ell_{N-1}) + \varepsilon^j, \, N \in \mathbb{E}, \tag{6.8g}$$

$$C^i \leq \mathtt{ct}_{N-1}^i(\ell_1, \ldots, \ell_{N-1}) + \varepsilon^i, \; N \in \mathbb{O}. \tag{6.8h}$$

Given c^j, (6.8b)-(6.8e) define a unique ℓ. Thus, the variable of optimization is now c^j. Constraints (6.8g)-(6.8h) ensure that the entire VOLD induces cooperation. Denote the set of feasible c^j in the above problem by $\mathcal{F}_r(N)$ and the optimal value by $C_r^j(N)$. In general, for any given $N \in \mathbb{N}$ and optimal value $C^j(N)$, one has $C_r^j(N) \geq C^j(N)$. However, for some N, we have $C_r^j(N) = C_*^j$.

Theorem 6.4.1. (Restriction is exact). *Let $C_*^j < \infty$ be the optimal value of* (6.2). *Then there exists an $N \geq 2$ such that $C_r^j(N) = C_*^j$.*

Proof. Let N be a minimizer of (6.2) and suppose ℓ is a solution of (6.6) for fixed N. Our method is to build a new VOLD, ℓ' satisfying (6.8b)-(6.8e), from ℓ. Initially, set $\ell' = \ell$ and let $k_0 \in [1, N-2]$ be the smallest k such that one of the constraints (6.6c)-(6.6d) is not active when evaluated at ℓ'. Assume k_0 is odd, so $C_*^j - \varepsilon^j < \mathtt{ct}_{k_0}^j(\ell'_1, \ldots, \ell'_{k_0})$. Increase ℓ'_{k_0} and decrease ℓ'_{k_0+2} at the same rate until $C_*^j - \varepsilon^j = \mathtt{ct}_{k_0}^j(\ell'_1, \ldots, \ell'_{k_0})$ (this is possible due to (P5)). After performing this procedure, the k_0 constraint is active, the RHS of the $k_0 + 1$ constraint has increased in value (see (P1) and (P3)), and all other constraint functions have maintained their original value (thus, the final cost-to-go for i and j have also remained at C_*^j and C^i respectively). Thus, ℓ' still solves (6.6). Next, we focus on the $k_0 + 1$ constraint whose RHS has increased in value and thus $C^i - \varepsilon^i < \mathtt{ct}_{k_0+1}^i(\ell'_1, \ldots, \ell'_{k_0+1})$. Again, increase ℓ'_{k_0+1} and decrease ℓ'_{k_0+3} at the same rate until the $k_0 + 1$ constraint becomes active. We are able to repeat this procedure until the $N - 1$ constraint is reached. It is not possible to make this constraint active using the same procedure because there is no ℓ'_{N+1} component to decrease. Therefore, once the $N-1$ constraint is reached, ℓ' satisfies (6.8b)-(6.8e). One point of concern in the proposed procedure occurs if when decreasing (say) ℓ'_{k_0+2} it happens that $\ell'_{k_0+2} \leq 0$. However, in this event, one can show that N is not a minimizer of (6.2): a new VOLD with one less leader switch can be constructed that decreases the objective function and satisfies the constraints. □

Recall that, given input C^j_*, the COST REALIZATION ALGORITHM generates a VOLD satisfying (6.8b)-(6.8h) for some N. Thus, the following is a result of Theorem 6.4.1.

Corollary 6.4.2. (Constructing an optimal solution: costly switching). *Under costly switching, the* COST REALIZATION ALGORITHM *with input* C^j_* *induces a VOLD which solves* (6.2).

Corollary 6.4.2 generalizes Corollary 6.3.2. Next, we state an analogous result to Theorem 6.3.1, allowing us to find the optimal value of (6.2) under costly switching.

Theorem 6.4.3. (Restriction is convex). *The problem* (6.8) *is convex.*

Proof. It suffices to show that $\mathcal{F}_r(N)$ is convex. Suppose that N is even, begin with $c^j \in \operatorname{int}(\mathcal{F}_r(N))$, and let $a > 0$ and $\mathbf{b} \in \mathbb{R}^N$ be sufficiently small such that the following analysis holds. Let $\ell + \mathbf{b}$ satisfy (6.8b)-(6.8e) for $c^j + a$. Towards characterizing $\mathbf{b}$, notice that $c^j + a - \varepsilon^j = \mathtt{ct}^j_1(\ell_1 + b_1)$. Hence, $a = b_1 \partial_{\ell_1} \mathtt{ct}^j_1(\ell_1)$, implying $b_1 < 0$. Next, note that $C^i - \varepsilon^i = \mathtt{ct}^i_2(\ell_1 + b_1, \ell_2 + b_2)$. Therefore

$$0 = b_1 \partial_{\ell_1} \mathtt{ct}^i_2(\ell_1, \ell_2) + b_2 \partial_{\ell_2} \mathtt{ct}^i_2(\ell_1, \ell_2),$$

from which we see that $b_2 < 0$. Repeating this argument while invoking Lemmas 6.2.2 and 6.2.5, we see that $b_k < 0$ for $k = 1, \ldots, N-2$ and b_{N-1}, b_N need to satisfy

$$b_{N-1} \partial_{\ell_2} \mathtt{ct}^j_N(\ell) + b_N \partial_{\ell_1} \mathtt{ct}^j_N(\ell) > 0,$$
$$b_{N-1} \partial_{\ell_2} \mathtt{ct}^i_N(\ell) + b_N \partial_{\ell_1} \mathtt{ct}^i_N(\ell) > 0.$$

Evoking (P5)-(P6), we deduce $b_{N-1}, b_N < 0$ as well, and hence $\mathbf{b} < 0$. Next, we study how (6.8g) changes as we increase slightly c^j. In particular, $\mathbf{b} < 0$ satisfies the equation $c^j + a = \mathtt{ct}_N^j(\ell + \mathbf{b})$. Or, in other words

$$\begin{aligned} a &= \partial_{\ell_1}\mathtt{ct}_N^j(\ell) \sum_{k\in\mathbb{N}\cap\mathbb{O}}^{N-1} b_k + \partial_{\ell_2}\mathtt{ct}_N^j(\ell) \sum_{k\in\mathbb{N}\cap\mathbb{E}}^{N} b_k, \\ &< \partial_{\ell_1}\mathtt{ct}_{N-1}^j(\ell_1,\ldots,\ell_{N-1}) \sum_{k\in\mathbb{N}\cap\mathbb{O}}^{N-1} b_k + \partial_{\ell_2}\mathtt{ct}_{N-1}^j(\ell_1,\ldots,\ell_{N-1}) \sum_{k\in\mathbb{N}\cap\mathbb{E}}^{N-2} b_k, \end{aligned} \tag{6.9}$$

where (P4) has been used. The LHS (resp. RHS) of (6.9) represents the increase in the LHS (resp. RHS) of (6.8g). Thus, (6.8g) remains satisfied by increasing c^j. Therefore, by increasing c^j the only constraint one may violate is $\ell \in \mathbb{R}_{\geq 0}^N \supset \mathbb{L}_N$. However, increasing c^j more further decreases each ℓ_i. Thus, $\mathcal{F}_r(N)$ must be convex. □

By Theorem 6.4.3, given $N \in \mathbb{N}$, the optimal value of (6.8) can be efficiently found under costly switching. Note that the restriction to the feasible set does not limit the type of real-world scenarios that we can solve. Moreover, the solution of (6.8) maximizes the distance between switches. From an implementation point of view, this is a desirable and robust switching protocol because UAVs are not required to perform switching maneuvers arbitrarily fast. To find the optimal value of (6.2), we next study how to determine the optimal number of leader switches.

6.4.2 Optimal number of leader switches

Here, we identify a criterion that allows us to determine an optimal N and helps us search for it. The following result provides such a criterion via a quasiconvexity-like property of $C_r^j(N)$. Figure 6.5 illustrates Theorem 6.4.4.

Theorem 6.4.4. (Certificate for optimal number of switches). *For $N \in \mathbb{N}$, the following statements hold*

(i) if adding two switches increases (6.8)

$$C_r^j(N) < C_r^j(N+2),$$

then adding any more multiple of two switches also increases it

$$C_r^j(N+2k) \leq C_r^j(N+2(k+1)), \quad \forall k \in \mathbb{N}.$$

The inequality is strict iff (6.8) *is feasible for* $N+2k$.

(ii) *if removing two switches increases* (6.8)

$$C_r^j(N) < C_r^j(N-2),$$

then removing any more multiple of two switches also increases it

$$C_r^j(N-2k) \leq C_r^j(N-2(k+1)), \ \forall k \leq N/2-2.$$

The inequality is strict iff (6.8) *is feasible for* $N-2k$.

The proof of the above Theorem makes use of the following two technical results.

Lemma 6.4.5. (Property of the last two lead/follow distances). *Let* $\ell^N \in \mathbb{L}_N$ *and* $\ell^{N+2k} \in \mathbb{L}_{N+2k}$, *for* $k \in \mathbb{N}$, *such that the distance to the second last switch under* ℓ^N *is less than or equal to the distance to the second last switch under* ℓ^{N+2k}

$$\sum_{k=1}^{N-2} \ell_k^N \leq \sum_{k=1}^{N+2(k-1)} \ell_k^{N+2k}, \tag{6.10a}$$

j*'s cost-to-target at the second last switch under* ℓ^N *is no more than its cost-to-target at the second last switch under* ℓ^{N+2k}

$$\mathtt{ct}^j_{N-2}(\ell_1^N, \ldots, \ell_{N-2}^N) \leq \mathtt{ct}^j_{N+2(k-1)}(\ell_1^{N+2k}, \ldots, \ell_{N+2(k-1)}^{N+2k}), \tag{6.10b}$$

i*'s cost-to-targets at the second last switch under both VOLDs are the same*

$$\mathtt{ct}^i_{N-2}(\ell_1^N, \ldots, \ell_{N-2}^N) = \mathtt{ct}^i_{N+2(k-1)}(\ell_1^{N+2k}, \ldots, \ell_{N+2(k-1)}^{N+2k}), \tag{6.10c}$$

and the final cost-to-targets for i *and* j *are the same under both VOLDs*

$$\mathtt{ct}^j_N(\ell^N) = \mathtt{ct}^j_{N+2k}(\ell^{N+2k}), \tag{6.10d}$$

$$\mathtt{ct}^i_N(\ell^N) = \mathtt{ct}^i_{N+2k}(\ell^{N+2k}). \tag{6.10e}$$

Then, $\ell_{N-1}^N < \ell_{N+2k-1}^{N+2k}$ *and* $\ell_N^N < \ell_{N+2k}^{N+2k}$.

Proof. Suppose that $k = 1$ and N is even. Recall (cf. Section 6.2.2) that $\mathtt{ct}_N^{i+j}$ is strictly decreasing and convex on $\mathbb{L}_N$. Also, $\mathtt{ct}_N^{i+j}$ has the convexity-like property

$$0 < \mathtt{ct}_N^{i+j}(L) - \mathtt{ct}_{N-2}^{i+j}(L) < \mathtt{ct}_{N+2}^{i+j}(L) - \mathtt{ct}_N^{i+j}(L).$$

Therefore, if for some $L_1 \leq L_2$ and $a, b > 0$, for some $a, b > 0$,

$$0 > \mathtt{ct}_N^{i+j}(L+a) - \mathtt{ct}_{N-2}^{i+j}(L) > \mathtt{ct}_{N+2}^{i+j}(L+b) - \mathtt{ct}_N^{i+j}(L),$$

then it must be that $a < b$. Building on this fact,

$$\begin{aligned} 0 &> \mathtt{ct}_N^{i+j}(L_1+a) - \mathtt{ct}_{N-2}^{i+j}(L_1) \\ &> \mathtt{ct}_{N+2}^{i+j}(L_2+b) - \mathtt{ct}_N^{i+j}(L_2), \end{aligned} \tag{6.11}$$

then $a < b$. Take $L_1 = \sum_{k=1}^{N-2} \ell_k^N$, $L_2 = \sum_{k=1}^{N+2k-2} \ell_k^{N+2k}$, $a = \ell_{N-1}^N + \ell_N^N$, and $b = \ell_{N+1}^{N+2} + \ell_{N+2}^{N+2}$. Now note that, expressing $(ii) - (iv)$ in terms of the combined fuel functions, shows that the condition (6.11) is satisfied for this choice of values. Thus $a = \ell_{N-1}^N + \ell_N^N < b = \ell_{N+1}^{N+2} + \ell_{N+2}^{N+2}$. Next, we show by contradiction that $\ell_{N-1}^N < \ell_{N+1}^{N+2}$ and $\ell_N^N < \ell_{N+2}^{N+2}$. Suppose $\ell_{N-1}^N \geq \ell_{N+1}^{N+2}$. Thus

$$\begin{aligned} &\mathtt{ct}_{N-2}^j(\ell_1^N, \ldots, \ell_{N-2}^N) - \mathtt{ct}_{N-1}^j(\ell_1^N, \ldots, \ell_{N-1}^N) \\ &> \mathtt{ct}_N^j(\ell_1^{N+2}, \ldots, \ell_N^{N+2}) - \mathtt{ct}_{N+1}^j(\ell_1^{N+2}, \ldots, \ell_{N+1}^{N+2}). \end{aligned}$$

Since (ii) is true and following is more beneficial to j earlier in the formation, the above implies that $\mathtt{ct}_{N-1}^j(\ell_1^N, \ldots, \ell_{N-1}^N) < \mathtt{ct}_{N+1}^j(\ell_1^{N+2}, \ldots, \ell_{N+1}^{N+2})$. To satisfy (iv), this would mean

$$\mathtt{ct}_N^j(\ell^N) - \mathtt{ct}_{N-1}^j(\ell_1^N, \ldots, \ell_{N-1}^N) > \mathtt{ct}_{N+2}^j(\ell^{N+2}) - \mathtt{ct}_{N+1}^j(\ell_1^{N+2}, \ldots, \ell_{N+1}^{N+2}).$$

Since leading is more costly further in the formation, the above can only be satisfied if $\ell_N^N > \ell_{N+2}^{N+2}$. However, this would contradict $\ell_{N-1}^N + \ell_N^N < \ell_{N+1}^{N+2} + \ell_{N+2}^{N+2}$. Reasoning instead with i's cost-to-target and starting with $\ell_N^N > \ell_{N+2}^{N+2}$, a similar contradiction can be reached. N odd and $k \geq 2$ can be handled similarly. □

Corollary 6.4.6. (Sufficient condition to benefit from switch removal). *Let ℓ solve* (6.6) *for $N \in \mathbb{N}$. If*

$$\mathtt{ct}_{N-2(k+1)}^j(\ell_1, \ldots, \ell_{N-2(k+1)}) \leq \mathtt{ct}_{N-2}^j(\ell_1, \ldots, \ell_{N-2}),$$

for some $k \leq N/2 - 2$ then $C_r^j(N-2k) < C_r^j(N)$.

Proof. Suppose that N is even and let ℓ^{N-2k} satisfy (6.8b)-(6.8f) for $N-2k$ and $C_r^j(N)$. Note that

$$\ell_n^{N-2k} = \ell_n, \quad \text{for } n = 1, \ldots, N-2(k+1). \tag{6.12}$$

Since the assumptions of Lemma 6.4.5 are satisfied, $\ell_{N-2k}^{N-2k} < \ell_N^N$. Thus

$$\begin{aligned} &\texttt{ct}_{N-2k}^j(\ell^{N-2k}) - \texttt{ct}_{N-2k-1}^j(\ell_1^{N-2k}, \ldots, \ell_{N-2k-1}^{N-2k}) \\ &\qquad\qquad < \texttt{ct}_N^j(\ell^N) - \texttt{ct}_{N-1}^j(\ell_1^N, \ldots, \ell_{N-1}^N), \\ \Rightarrow\ &\texttt{ct}_{N-2k-1}^j(\ell_1^{N-2k}, \ldots, \ell_{N-2k-1}^{N-2k}) > \texttt{ct}_{N-1}^j(\ell_1^N, \ldots, \ell_{N-1}^N) \geq C_r^j(N) - \varepsilon^j. \end{aligned}$$

In other words, (6.8g) is satisfied for ℓ^{N-2k}, $C_r^j(N)$ is feasible for (6.8) for VOLDs of cardinality $N-2k$. and thus $C_r^j(N-2k) < C_r^j(N)$ (this relation is strict because (6.8g) is not active and thus there exists a feasible $c^j < C_r^j(N)$). N odd can be dealt with analogously. □

We may now prove Theorem 6.4.4

Proof of Theorem 6.4.4. The result follows from the combination of:

(S1) If $C_r^j(N) < C_r^j(N+2)$ for some $N \in \mathbb{N}$, then $C_r^j(N) \leq C_r^j(N+2k)$ for all $k \in \mathbb{N}$. The inequality is strict iff (6.8) is feasible for $N+2$.

(S2) If $C_r^j(N) < C_r^j(N-2)$ for some $N \in \mathbb{N}$, then $C_r^j(N) \leq C_r^j(N-2k)$ for all $k \in \mathbb{N}$ such that $k \leq N/2 - 2$. The inequality is strict iff (6.8) is feasible for $N-2$.

Suppose N is even. Consider first (S1). If $C_r^j(N) < C_r^j(N+2)$ then $C_r^j(N) \notin \mathcal{F}_r(N+2)$. That is, for $N+2$ and $c^j = C_r^j(N)$, either (S1.1) constraints (6.8b)-(6.8f) are violated or (S1.2) the constraint (6.8g) is violated.

Consider (S1.1). Let $c_{\min}^j(N)$ be the minimum c^j such that (6.8b)-(6.8f) are satisfied for N. As per the analysis in the proof of Theorem 6.4.3, $c_{\min}^j(N) \equiv \min_L \texttt{ct}_N^{i+j}(L) - C^i$. By the properties of $\texttt{ct}_N^{i+j}$, it follows that $c_{\min}^j(N) < c_{\min}^j(N+2k)$ for any $k \in \mathbb{N}$. Therefore, if (S1.1) is true, this means that $C_r^j(N) \notin \mathcal{F}_r(N+2k) \Rightarrow C_r^j(N) < C_r^j(N+2k)$ and this would prove (S1).

Consider (S1.2). Let ℓ^N (resp. ℓ^{N+2}) (resp. ℓ^{N+4}) satisfy (6.8g) for $C_r^j(N)$ and N (resp. $N+2$) (resp. $N+4$). We know $\ell_k^{N+2} = \ell_k^N$ for $k = 1, \ldots, N-1$. Denote $\ell_N^{N+2} = \ell_N^N + a$. Note that

$$\mathtt{ct}_N^i(\ell^N) = \mathtt{ct}_N^i(\ell_1^N, \ldots, \ell_N^N + a) + \varepsilon^i, \tag{6.13}$$

so $a > 0$. Also, because (6.8g) is violated

$$C_r^j(N) > \mathtt{ct}_{N+1}^j(\ell_1^N, \ldots, \ell_{N-1}^N, \ell_N^N + a, \ell_{N+1}^{N+2}) + \varepsilon^j. \tag{6.14}$$

For now let $\ell_k^{N+4} = \ell_k^{N+2}$ for $k = 1, \ldots, N+1$ and denote $\ell_{N+2}^{N+4} = \ell_{N+2}^{N+2} + b$. Define $\ell_{N+3}^{N+4}, \ell_{N+4}^{N+4}$ implicitly by

$$\begin{aligned} C^i &= \mathtt{ct}_{N+2}^i(\ell_1^{N+2}, \ldots, \ell_{N+2}^{N+2} + b) + \varepsilon^i, \\ c^j &= \mathtt{ct}_{N+4}^j(\ell_1^{N+4}, \ldots, \ell_{N+4}^{N+4}), \\ C^i &= \mathtt{ct}_{N+4}^i(\ell_1^{N+4}, \ldots, \ell_{N+4}^{N+4}). \end{aligned}$$

Likewise, $b > 0$ since

$$\mathtt{ct}_{N+2}^i(\ell^{N+2}) = \mathtt{ct}_{N+2}^i(\ell_1^{N+2}, \ldots, \ell_{N+2}^{N+2} + b) + \varepsilon^i.$$

Comparing the above and (6.13), we see that $a < b$. Thus $\mathtt{ct}_N^j(\ell_1^N, \ldots, \ell_N^N + a) < \mathtt{ct}_{N+2}^j(\ell_1^{N+2}, \ldots, \ell_{N+2}^{N+2} + b)$. Therefore, the conditions of Lemma 6.4.5 are satisfied and $\ell_{N+4}^{N+4} > \ell_{N+2}^{N+2}$. Since j leads last, this means that $\mathtt{ct}_{N+3}^j(\ell_1^{N+4}, \ldots, \ell_{N+3}^{N+4}) < \mathtt{ct}_{N+1}^j(\ell_1^{N+2}, \ldots, \ell_{N+1}^{N+2})$. Recalling (6.14), the above means that (6.8d) is violated. As a final step to creating ℓ^{N+4}, employ the strategy as in the proof of Theorem 6.4.1: decrease ℓ_{N+1}^{N+4} and increase ℓ_{N+3}^{N+4} by the same amount until (6.8b) is satisfied. However, increasing ℓ_{N+3}^{N+4} further violates (6.8d). Therefore, $C_r^j(N) < C_r^j(N+4)$. We can repeat this process for $N+6$ and so on to attain the desired result. So long as (6.8b)-(6.8f) are satisfied for $C_r^j(N)$ and $N+2k$, the above construction is valid. However, if $C_r^j(N+2) = \infty$, recall (S1.1). Then $C_r^j(N+2k) = \infty$ for all k. This completes the proof of (S1).

Next, we prove (S2). Let ℓ^N (resp. ℓ^{N-2}) (resp. ℓ^{N-4}) satisfy (6.8g) for $C_r^j(N)$ and N (resp. $N-2$) (resp. $N-4$). To reach a contradiction, suppose

$$C_r^j(N-4) \leq C_r^j(N) < C_r^j(N-2). \tag{6.15}$$

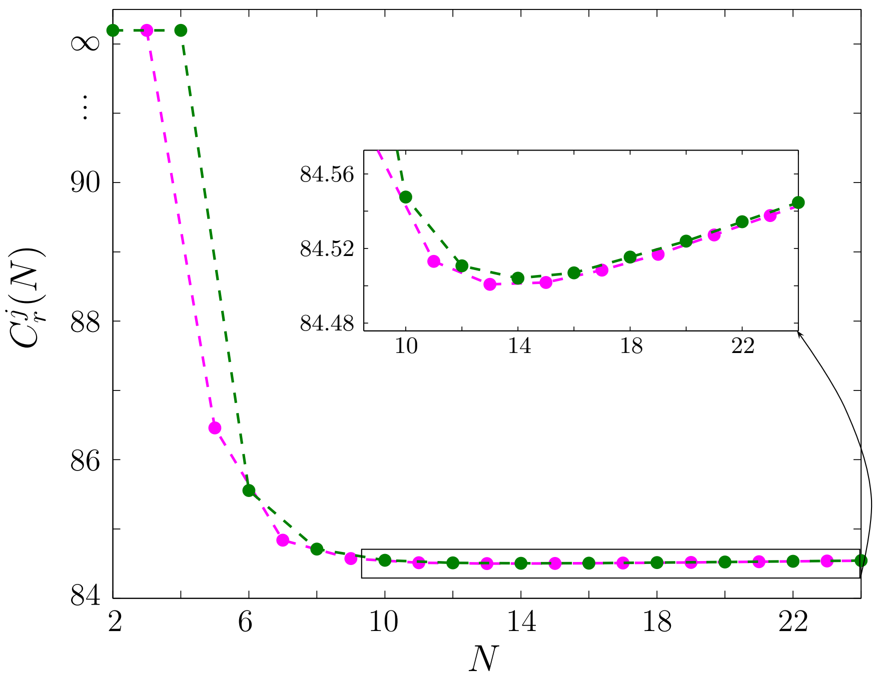


Figure 6.5: An example of the optimal value of (6.6) with respect to N. The magenta (resp. dark green) dots represent $C_r^j(N)$ for odd (resp. even) N.

First, let $a > b > 0$ be such that

$$C^i = \mathtt{ct}^i_{N-2}(\ell^N_1, \ldots, \ell^N_{N-3}, \ell^N_{N-2} - a),$$
$$= \mathtt{ct}^i_{N-4}(\ell^N_1, \ldots, \ell^N_{N-5}, \ell^N_{N-4} - b).$$

Let $L = \sum_{k=1}^{N-2} \ell^N_{N-2} - a$. If $c^j \geq \mathtt{ct}^j_{N-2}(\ell^N_1, \ldots, \ell^N_{N-2} - a)$ then $c^j + C^i > \mathtt{ct}^{i+j}_{N-2}(L)$ (i.e., the formation length is too long). Since we know $\ell^N_k = \ell^{N-2}_k$ for $k = 1, \ldots, N-4$, decreasing the formation distance must be accomplished by decreasing $\ell^N_{N-3} + \ell^N_{N-2} - a$. Since i's cost-to-target must be maintained at C^i, both ℓ^N_{N-3} and $\ell^N_{N-2} - a$ must decrease (i.e., $\ell^{N-2}_{N-3} \leq \ell^N_{N-3}$). But since (6.8g) is violated for ℓ^{N-2}, it must be that $\ell^{N-2}_{N-3} > \ell^N_{N-3}$ which is a contradiction. Thus, it must be that $c^j < \mathtt{ct}^j_{N-2}(\ell^N_1, \ldots, \ell^N_{N-2} - a)$. Under the assumption of (6.15), a similar argument can be made to show that $c^j \geq \mathtt{ct}^j_{N-4}(\ell^N_1, \ldots, \ell^N_{N-4} - b)$. Let us now reverse the change of a (resp. b) in ℓ_{N-2} (resp. ℓ_{N-4}). Then we see that $\mathtt{ct}^j_{N-4}(\ell^N_1, \ldots, \ell^N_{N-4}) < \mathtt{ct}^j_{N-2}(\ell^N_1, \ldots, \ell^N_{N-2})$. But, by Corollary 6.4.6, this would mean that $C^j_r(N-2) < C^j_r(N)$, contradicting (6.15). The claim can be extended analogously for cases where more switches are removed. Thus, $C^j_r(N) < C^j_r(N-4)$. So long as (6.8b)-

(6.8f) are satisfied for $C_r^j(N)$ and $N-2k$, the above construction is valid. Thus, $C_r^j(N-2)=\infty \Rightarrow C_r^j(N-4)=\infty$, and so on. This proves (S2). $\square$

Next, we design a method to find the optimal N. Define

$$\Delta_N := C_r^j(N) - C_r^j(N+2).$$

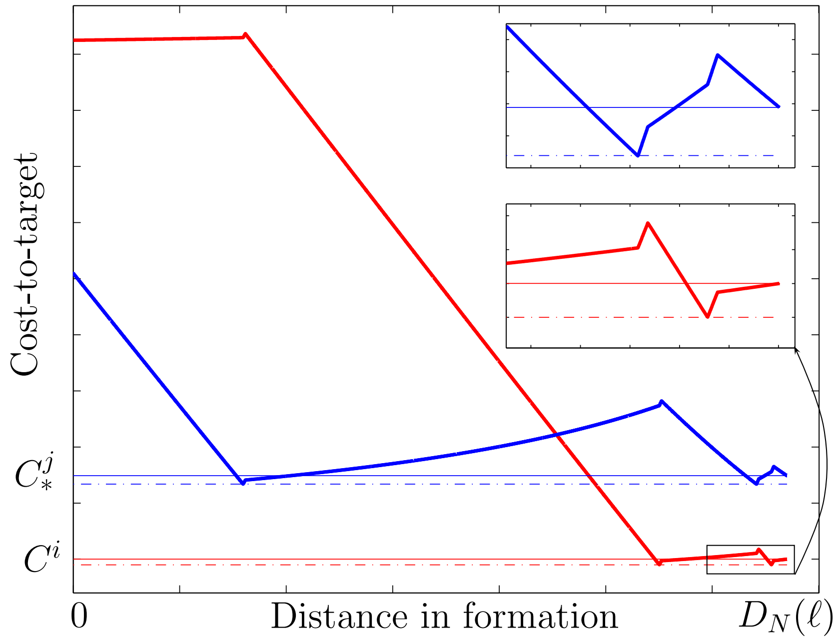
If $N^* \in \mathbb{E}$ is optimal, then Theorem 6.4.4 implies $\Delta_N > 0$ for all $N \in \mathbb{N}_{[2,N^*)} \cap \mathbb{E}$ and $\Delta_N < 0$ for all $N \in \mathbb{N}_{(N^*,\infty)} \cap \mathbb{E}$ (so long as Δ_N is finite). Also, $0 \in [\Delta_{N^*}, \Delta_{N^*-2}]$. Thus, the problem of finding an optimal N is well-suited for a binary search (see [31]), which is presented in Algorithm 1 adapted to our problem.

Algorithm 1 The BINARY SEARCH ALGORITHM

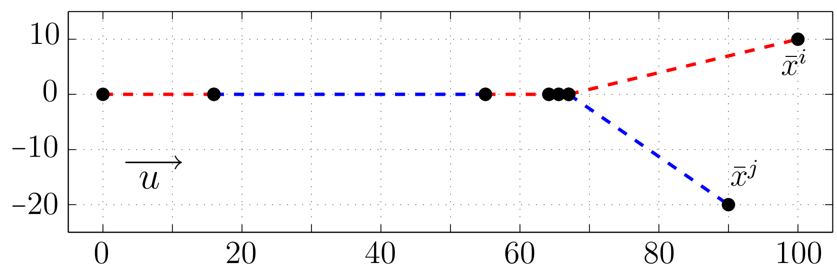
Input: N with Δ_N or Δ_{N+2} finite

if $\Delta_N \geq 0$ **then**
  $N_l := N$ and $N_u := N+2$
  while $\Delta_{N_u} \geq 0$ **do**
    $N_l := N_u$
    $N_u \mapsfrom 2N_u$
  end while
else $N_l := 2 + (N \mod 2)$ and $N_u := N$
end if
while $N_u - N_l \geq 4$ and $\Delta_N \neq 0$ **do**
  $N := (N_u - N_l)/2$
  if $\Delta_N \geq 0$
    $N_l := N$
  else $N_u := N$ **end if**
end while
return N if $\Delta_N \leq 0$, $N+1$ otherwise

The method is implemented for odd and even inputs, the optimal N being the lesser of the two outputs. The computational intensity of the BINARY SEARCH ALGORITHM stems from the evaluation of Δ_N, which solves (6.8).



(a) Cost-to-targets under an optimal VOLD



(b) Switch locations under the above VOLD

Figure 6.6: An optimal cooperation inducing VOLD. N^* is computed using the BINARY SEARCH ALGORITHM, $C_r^j(N^*) = 85$ is fed to the COST REALIZATION ALGORITHM to attain the optimal VOLD. Note the effect of costly switching on the cost-to-target function. The data for the simulation are: $C^i = 82$, $s = 0.2$, $\Gamma = 1.7$, $\gamma = 0.5$, $\varepsilon^i = 0.2$, $\varepsilon^j = 0.3$, $\bar{x}^i = (100, 10)$, $\bar{x}^j = (90, -20)$. The cost-benefits of the formation are 18 (18%) and 7 (7.6%) for i and j, resp. (slightly less for each UAV than the no-cost of switching case).

Corollary 6.4.7. (Correctness and complexity). *Suppose* (6.2) *is feasible and* N^* *is its minimizer. Let* $N_0^{\mathrm{o}} \in \mathbb{O}$ *(resp.* $N_0^{\mathrm{e}} \in \mathbb{E}$*) be a valid input to the* BINARY SEARCH ALGORITHM *with output* N^{o} *(resp.* N^{e}*). Then* $N^* \in \{N^{\mathrm{o}}, N^{\mathrm{e}}\}$*. Moreover, to determine* N^* *the problem* (6.8) *is solved at most* $4\lceil \log_2 N^* \rceil$ *times.*

The correctness result in Corollary 6.4.7 follows from Theorem 6.4.4 and the complexity result is inherited from binary search algorithms [31]. Figure 6.6 presents simulation results verifying the correctness of the BINARY SEARCH ALGORITHM. The main differences when compared to the simulations presented in Figure 6.4 pertain to the cost of switching ($s\Gamma$) being positive and the degree of cooperation between agents ($\varepsilon^i, \varepsilon^j$). The agents in Figure 6.4 are able to induce cooperation even when $\varepsilon^i, \varepsilon^j$ are small because adding a switch does not increase their final cost-to-target. On the other hand, when there is a cost associated with switching, the agents are not able to switch arbitrarily fast without increasing their final cost-to-targets. In fact, the problem (6.6) is infeasible for small $\varepsilon^i, \varepsilon^j$ under costly switching. In terms of real-world implementation, the BINARY SEARCH ALGORITHM is run prior to agents beginning their formation. For this reason, the implementation time of the BINARY SEARCH ALGORITHM is not reflected in the simulation of Figure 6.6. However, on-board processors must be able to run the BINARY SEARCH ALGORITHM within the time required for UAVs to fly from their current location to the formation rendezvous location. If executed online, the complexity bound for the binary search (cf. Corollary 6.4.7) provides guidance as to how fast a processor should be with respect to UAV motion.

This concludes our study of cooperation inducing mechanisms for UAV formation pairs.

Chapter 6, in part, is a reprint of the material [79] as it appears in ‘Optimal leader allocation in UAV formation pairs under no-cost switching’ by D. Richert and J. Cortés in the proceedings of the 2012 American Control Conference as well as [78] as it appears in ‘Optimal leader allocation in UAV formation pairs under costly switching’ by D. Richert and J. Cortés in the proceedings of the 2012 IEEE Conference on Decision and Control as well as [82] as it appears in ‘Optimal leader allocation in UAV formation pairs ensuring cooperation’ by D. Richert and J. Cortés in Automatica. The dissertation author was the primary investigator and author of these papers.

Chapter 7

Conclusions

In this thesis, we studied the design of control algorithms for network systems. As a natural starting point, we considered network control objectives posed as mathematical optimization problems. Focusing our attention on linear programs in particular, we investigated problems where the network structure was consistent with the sparsity structure of the linear constraints.

Building the foundation for the rest of the thesis, Chapter 3 solved the problem of designing a robust continuous-time distributed dynamics to solve linear programs. In this context, the network objective was for the aggregate of the agents' states to converge to a solution of the linear program. We proposed an equivalent formulation of this problem in terms of finding the saddle points of a modified Lagrangian function. To make an exact correspondence between the solutions of the linear program and saddle points of the Lagrangian we incorporated a nonsmooth penalty term. This formulation naturally led us to study the associated saddle-point dynamics, for which we established the point-wise convergence to the set of solutions of the linear program. Based on this analysis, we introduced an alternative algorithmic solution with the same asymptotic convergence properties. In this chapter, we also studied the robustness against disturbances and link failures of the dynamics. We showed that it is integral-input-to-state stable but not input-to-state stable (and, in fact, no algorithmic solution for linear programming is). These results allowed us to formally establish the resilience of our distributed dynamics to disturbances of finite variation and recurrently disconnected communication graphs.

Towards a more realistic implementation of the continuous-time dynamics, Chapter 4 sought to design an event-triggered communication protocol. Rather than having continuous flow of information between agents, we looked at a more practical model where agents autonomously and opportunistically decide when to broadcast their state to neighbors. Our methodology combined elements from linear programming, switched and hybrid systems, event-triggered control, and Lyapunov stability theory to provide provably correct centralized and distributed strategies. We rigorously characterized the asymptotic convergence of persistently flowing executions to a solution of the linear program. We also identified a sufficient condition for executions to be persistently flowing, and based on it, we conjecture that they all are. We also identified the area of switched and hybrid systems as benefiting from the contributions of this chapter. To our knowledge, this thesis is the first to consider event-triggered protocols for that class of systems.

Turning our attention to a more specific network control problem, in Chapter 5 we considered bargaining between agents in an exchange network. In particular, we studied dyadic-exchanges where an agent can pair with at most one other agent. In the end, players had to autonomously decide with whom (if any) to match and agree on an allocation of a common good. We designed continuous-time distributed dynamics to converge to each of stable, balanced, and Nash outcomes. The robust and distributed linear programming dynamics developed in Chapter 3 were instrumental for agents to find stable outcomes. The distributed balancing dynamics we proposed had an intuitive interpretation of agents adjusting their allocations based on an error measuring how far the allocations of matched agents were from being balanced. The proof of convergence of the balancing dynamics made use of powerful results from nonsmooth analysis and set-valued dynamical systems. The final Nash bargaining dynamics used a clever combination of both the dynamics to find stable outcomes and the dynamics to find balanced outcomes. Applying the Nash bargaining dynamics to a wireless communication scenario, we showed how agent collaborations can, in a fair way, improve both individual and network-wide performance. In particular, agents who had the option of sharing their allocation of a communication channel achieved an improved effective bit rate.

Finally, we questioned how matched agents in a bargaining outcome can realize their promised allocations. We made our point by studying the optimal leader allocation in UAV formation pairs in Chapter 6. We studied how strategic allocations of the leading role could induce cooperation between selfish UAVs. If UAVs were completely selfish, then such mechanisms did not exist. However, UAVs that were ε-cooperative could indeed construct optimal leader allocations. Formulated as a nonlinear program, the problem posed two distinct challenges: (i) determining the optimal leader allocation when given a fixed number of leader switches, and (ii) finding the optimal number of leader switches. We showed that, when switching the lead has no cost, the optimal value can be obtained via a convex program and we designed the COST REALIZATION ALGORITHM to determine an optimal cooperation-inducing leader allocation. In the costly switching case, we restricted the feasible set of allocations to mimic the structure of the solutions provided by this policy. The resulting restriction has the same optimal value and, for a fixed number of leader switches, is convex. We also unveiled a quasiconvexity-like property of the optimal value as a function of the number of switches and designed the BINARY SEARCH ALGORITHM to find the optimal number in logarithmic time.

7.1 Future research directions

This thesis provides a constructive framework for future developments. Our control systems approach to each problem leaves open the possibility of numerous extensions and additional analysis. Here we outline some general future research directions and then a few in the context of the specific contributions of this thesis.

User privacy in networks has become a critical consideration of late. Therefore, it is increasingly important that messages passed between agents do not reveal sensitive data to adversaries. A distributed approach to privately solving linear programs has not yet been explored, and we believe that the results of this thesis could be applied in that direction. Also, linear programs have manifested themselves in recent work on unreliable sensor networks and power distribution systems. We would like to further explore how our contributions in this thesis would apply to those problems.

With regards to robust distributed linear programming, the rate of convergence of the algorithm has yet to be determined. Knowing it would open up the study of this dynamics to scenarios where the data of the linear program is changing in time. Related to this, we would like to characterize the behavior of the dynamics when the local information of an agent is inconsistent with its neighbors. Also, we observed that this dynamics is robust to more general link failure models, beyond recurrently connected graphs. Specifically, we strongly believe that the dynamics is robust to graphs that are uniformly jointly connected. On the topic of robustness, we would like to explore the convergence of our dynamics under disturbance signals with certain statistical properties, such as white noise. Another direction we would like to explore is the extension of the dynamics to more general convex optimization problems, establishment of the robustness properties in those cases, and the comparison to existing optimization algorithms. Finally, we plan to explore the benefits of the proposed distributed dynamics in a number of engineering scenarios, including for example the smart grid and power distribution and model predictive control.

In terms of event-triggered optimization, we would like to formally extend our contributions to switched and hybrid systems in general. Additionally, it is yet to be established that all solutions of the event-triggered implementation are persistently flowing, though we strongly believe them to be. Further characterizing the robustness properties (i.e., beyond robust asymptotic stability) of our design would also be constructive. Finally, implementing the results of Chapter 4 on a multi-agent testbed would also provide insight into other implementation issues that may exist.

Future research in the area of network bargaining will include considering other solution concepts on dyadic-exchange networks such as the nucleolus. Applying our techniques to multi-exchange networks (i.e., allowing coalitions of more than two) would also be interesting. In addition, we would like to study the rate of convergence and establish more quantifiable robustness properties of the balancing dynamics; in particular, the effects of time delays, adversarial agents, and dynamically changing system data. Finally, we wish to apply our dynamics to other coordination tasks and implement them on a real multi-agent testbed.

Extensions in the area of cooperation-inducing mechanisms would include exploring more abstract problem setups, thus broadening the utility of our contributions beyond UAV formation pairs. Another interesting question is whether our results can apply to formations of more than two UAVs or scenarios with obstacle avoidance/no-fly zones. Also, formally connecting the results of Chapter 5 and Chapter 6 would allow us to consider networks of multiple UAVs.

Bibliography

[1] B. Açıkmeşe and L. Blackmore. Lossless convexification of a class of optimal control problems with non-convex control constraints. *Automatica*, 47(2):341–347, 2011.

[2] R. Alberton, R. Carli, A. Cenedese, and L. Schenato. Multi-agent perimeter patrolling subject to mobility constraints. In *American Control Conference*, pages 4498–4503, Montreal, 2012.

[3] S. H. Ali, K. Lee, and V. C. M. Leung. Dynamic resource allocation in OFDMA wireless metropolitan area networks. *IEEE Wireless Communications*, 14(1):6–13, Feb. 2007.

[4] T. Alpcan and T. Basar. A globally stable adaptive congestion control scheme for internet-style networks with delay. *IEEE/ACM Transactions on Networking*, 13(6):1261–1274, 2005.

[5] T. Alpcan, T. Basar, R. Srikant, and E. Altman. CDMA uplink power control as a noncooperative game. *Wireless Networks*, 8:659–670, 2002.

[6] D. Angeli, E. D. Sontag, and Y. Wang. A characterization of integral input-to-state stability. *IEEE Transactions on Automatic Control*, 45(6):1082–1097, 2000.

[7] A. Anta and P. Tabuada. To sample or not to sample: self-triggered control for nonlinear systems. *IEEE Transactions on Automatic Control*, 55(9):2030–2042, 2010.

[8] K. Arrow, L Hurwitz, and H. Uzawa. *Studies in Linear and Non-Linear Programming*. Stanford University Press, Stanford, California, 1958.

[9] J. P. Aubin and A. Cellina. *Differential Inclusions*. Springer, New York, 1994.

[10] Y. Azar, B. Birnbaum, L. Celis, N. Devanur, and Y. Peres. Convergence of local dynamics to balanced outcomes. 2009. Available at http://arxiv.org/abs/0907.4356.

[11] A. Bangla and D. Castanon. Auction algorithm for nonlinear resource allocation problems. In *IEEE Conf. on Decision and Control*, pages 3920–3925, Altanta, GA, December 2010.

[12] M. Bateni, M. Hajiaghayi, N. Immorlica, and H. Mahini. The cooperative game theory foundations of network bargaining games. In *International Colloquium on Automata, Languages and Programming*, pages 67–78, Bordeaux, France, July 2010.

[13] M. Bayati, C. Borgs, J. Chayes, Y. Kanoria, and A. Montanari. Bargaining dynamics in exchange networks. *Journal of Economic Theory*, 2014. To appear.

[14] D. P. Bertsekas. *Network Optimization: Continuous and Discrete Models*. Athena Scientific, 1998.

[15] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 2nd edition, 1999.

[16] D. P. Bertsekas and D. A. Castañón. Parallel synchronous and asynchronous implementations of the auction algorithm. *Parallel Computing*, 17:707–732, 1991.

[17] D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, Belmont, MA, 1st edition, 2003.

[18] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.

[19] D. Bertsimas, D. B. Brown, and C. Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011.

[20] D. Bertsimas and J. N. Tsitsiklis. *Introduction to Linear Optimization*, volume 6 of *Optimization and Neural Computation*. Athena Scientific, Belmont, MA, 1997.

[21] F. Borrelli, T. Keviczky, and G. J. Balas. Collision-free UAV formation flight using decentralized optimization and invariant sets. In *IEEE Conf. on Decision and Control*, pages 1099–1104, Atlantis, Paradise Island, Bahamas, Dec. 2004.

[22] S. Boyd, N. Parikh, E. Chu ad B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[23] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2009.

[24] F. Bullo, J. Cortés, and S. Martínez. *Distributed Control of Robotic Networks*. Applied Mathematics Series. Princeton University Press, 2009. Electronically available at http://coordinationbook.info.

[25] M. Burger, G. Notarstefano, F. Bullo, and F. Allgower. A distributed simplex algorithm for degenerate linear programs and multi-agent assignment. *Automatica*, 48(9):2298–2304, 2012.

[26] C. Cai, A. R. Teel, and R. Goebel. Smooth Lyapunov functions for hybrid systems part II: (pre)asymptotically stable compact sets. *IEEE Transactions on Automatic Control*, 53(3):734–748, 2008.

[27] B. W. Carabelli, A. Benzing, F. Dürr, B. Koldehofe, K. Rothermel, G. Seyboth, R. Blind, M. Burger, and F. Allgower. Exact convex formulations of network-oriented optimal operator placement. In *IEEE Conf. on Decision and Control*, pages 3777–3782, Maui, December 2012.

[28] T. Chakraborty, S. Judd, M. Kearns, and J. Tan. A behavioral study of bargaining in social networks. In *ACM Conference on Electronic Commerce*, pages 243–252, June 2010.

[29] F. H. Clarke. *Optimization and Nonsmooth Analysis*. Canadian Mathematical Society Series of Monographs and Advanced Texts. Wiley, 1983.

[30] K. S. Cook and R. M. Emerson. Power, equity and commitment in exchange networks. *American Sociological Review*, 43:721–739, October 1978.

[31] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, Cambridge, MA, 3rd edition, 2009.

[32] J. Cortés. Discontinuous dynamical systems - a tutorial on solutions, nonsmooth analysis, and stability. *IEEE Control Systems Magazine*, 28(3):36–73, 2008.

[33] J. Cortés. Distributed algorithms for reaching consensus on general functions. *Automatica*, 44(3):726–737, 2008.

[34] J. Cortés, S. Martínez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.

[35] G. B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, NJ, 1963.

[36] G. B. Dantzig. *Linear Programming: 1: Introduction*. Springer, New York, 1997.

[37] D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson. Distributed event-triggered control for multi-agent systems. *IEEE Transactions on Automatic Control*, 57(5):1291–1297, 2012.

[38] R. Dorfman, P. A. Samuelson, and R. Solow. *Linear programming in economic analysis*. McGraw Hill, New York, Toronto, and London, 1958.

[39] D. Feijer and F. Paganini. Stability of primal-dual gradient dynamics and applications to network optimization. *Automatica*, 46:1974–1981, 2010.

[40] C. A. Floudas and C. E. Gounaris. A review of recent advances in global optimization. *Journal of Global Optimization*, 45(1):3–38, 2009.

[41] B. Gharesifard and J. Cortés. Distributed continuous-time convex optimization on weight-balanced digraphs. *IEEE Transactions on Automatic Control*, 59(3):781–786, 2014.

[42] F. Giulietti, L. Pollini, and M. Innocenti. Autonomous formation flight. *IEEE Control Systems Magazine*, 20(6):34–44, 2000.

[43] R. Goebel, R. G. Sanfelice, and A. R. Teel. Hybrid dynamical systems. *IEEE Control Systems Magazine*, 29(2):28–93, 2009.

[44] R. Goebel, R. G. Sanfelice, and A. R. Teel. *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. Princeton University Press, 2012.

[45] G. H. Hardy, J. E. Littlewood, and G. Polya. *Inequalities*. Cambridge University Press, Cambridge, UK, 1952.

[46] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada. An introduction to event-triggered and self-triggered control. In *IEEE Conf. on Decision and Control*, pages 3270–3285, Maui, HI, 2012.

[47] J. P. Hespanha. Uniform stability of switched linear systems: Extensions of LaSalle's Invariance Principle. *IEEE Transactions on Automatic Control*, 49(4):470–482, 2004.

[48] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[49] Q. Hui and W. M. Haddad. Semistability of switched dynamical systems, Part 1: Linear systems theory having a continuum of equilibria. *Nonlinear Analysis: Hybrid Systems*, 3(3):343–353, 2009.

[50] D. Hummel. Aerodynamic aspects of formation flight in birds. *Journal of Theoretical Biology*, 104(3):321–347, 1983.

[51] M. Ji, S. Azuma, and M. Egerstedt. Role-assignment in multi-agent coordination. *International Journal of Assistive Robotics and Mechatronics*, 7(1):32–40, 2006.

[52] B. Johansson, T. Keviczky, M. Johansson, and K. H. Johansson. Subgradient methods and consensus algorithms for solving convex optimization problems. In *IEEE Conf. on Decision and Control*, pages 4185–4190, Cancun, Mexico, 2008.

[53] S. S. Kia, J. Cortés, and S. Martínez. Distributed convex optimization via continuous-time coordination algorithms with discrete-time communication. *Automatica*, 2014. Submitted.

[54] J. Kleinberg and É. Tardos. Balanced outcomes in social exchange networks. In *Proceedings of the Annual ACM Symposium on Theory of Computing*, pages 295–304, Victoria, Canada, May 2008.

[55] M. Kraning, E. Chu, J. Lavaei, and S. Boyd. Dynamic network energy management via proximal message passing. *Foundations and Trends in Optimization*, 1(2):70–122, 2013.

[56] D. R. Kuehn and J. Porter. The application of linear programming techniques in process control. *IEEE Transactions on Applications and Industry*, 83(75):423–427, 1964.

[57] D. Liberzon. *Switching in Systems and Control*. Systems & Control: Foundations & Applications. Birkhäuser, 2003.

[58] F. Lin, M. Fardad, and M. R. Jovanovic. Design of optimal sparse feedback gains via the alternating direction method of multipliers. *IEEE Transactions on Automatic Control*, 58(9):2426–2431, 2013.

[59] S. H. Low. Convex relaxation of optimal power flow – part i: Formulations and equivalence. *IEEE Transactions on Control of Network Systems*, 1(1):15–27, 2014.

[60] O. L. Mangasarian and R. R. Meyer. Nonlinear perturbation of linear programs. *SIAM Journal on Control and Optimization*, 17(6):745–752, 1979.

[61] S. Mathur, L. Sankaranarayanan, and N. Mandayam. Coalitions in cooperative wireless networks. *IEEE Journal on Selected Areas of Communication*, 26(7):1104–1115, 2008.

[62] M. Mazo Jr. and P. Tabuada. Decentralized event-triggered control over wireless sensor/actuator networks. *IEEE Transactions on Automatic Control*, 56(10):2456–2461, 2011.

[63] M. Mesbahi and M. Egerstedt. *Graph Theoretic Methods in Multiagent Networks*. Applied Mathematics Series. Princeton University Press, 2010.

[64] B. J. Moore and K. M. Passino. Distributed task assignment for mobile agents. *IEEE Transactions on Automatic Control*, 52(4):749–753, April 2007.

[65] J. Nash. The bargaining problem. *Econometrica*, 8(2):155–162, 1950.

[66] A. Nedić and A. Olshevsky. Distributed optimization over time-varying directed graphs. 2013. arXiv:1303.2289.

[67] A. Nedic and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

[68] A. Nedic, A. Ozdaglar, and P. A. Parrilo. Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, 55(4):922–938, 2010.

[69] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007.

[70] M. Nokleby and B. Aazhang. User cooperation for energy-efficient cellular communications. In *IEEE Int. Conf. on Communications*, pages 1–5, Cape Town, SA, May 2010.

[71] A. Nosratinia, T. E. Hunter, and A. Hedayat. Cooperative communication in wireless networks. *IEEE Communications Magazine*, 42(10):74–80, 2004.

[72] G. Notarstefano and F. Bullo. Distributed abstract optimization via constraints consensus: Theory and applications. *IEEE Transactions on Automatic Control*, 56(10):2247–2261, 2011.

[73] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[74] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, 2004.

[75] M. Ouimet and J. Cortés. Hedonic coalition formation for optimal deployment. *Automatica*, 49(11):3234–3245, 2013.

[76] W. Ren and R. W. Beard. *Distributed Consensus in Multi-vehicle Cooperative Control*. Communications and Control Engineering. Springer, 2008.

[77] G. Ribichini and E. Frazzoli. Efficient coordination of multiple-aircraft systems. In *IEEE Conf. on Decision and Control*, pages 1035–1040, Maui, Hawaii, 2003.

[78] D. Richert and J. Cortés. Optimal leader allocation in UAV formation pairs under costly switching. In *IEEE Conf. on Decision and Control*, pages 831–836, Maui, Hawaii, 2012.

[79] D. Richert and J. Cortés. Optimal leader allocation in UAV formation pairs under no-cost switching. In *American Control Conference*, pages 3297–3302, Montréal, Canada, 2012.

[80] D. Richert and J. Cortés. Distributed linear programming and bargaining in exchange networks. In *American Control Conference*, pages 4624–4629, Washington, D.C., 2013.

[81] D. Richert and J. Cortés. Integral input-to-state stable saddle-point dynamics for distributed linear programming. In *IEEE Conf. on Decision and Control*, pages 7480–7485, Florence, Italy, 2013.

[82] D. Richert and J. Cortés. Optimal leader allocation in UAV formation pairs ensuring cooperation. *Automatica*, 49(11):3189–3198, 2013.

[83] D. Richert and J. Cortés. Robust distributed linear programming. *IEEE Transactions on Automatic Control*, 2013. Submitted. Available at `http://carmenere.ucsd.edu/jorge`.

[84] D. Richert and J. Cortés. Distributed bargaining in dyadic-exchange networks. *IEEE Transactions on Control of Network Systems*, 2014. Submitted.

[85] D. Richert and J. Cortés. Distributed event-triggered optimization for linear programming. In *IEEE Conf. on Decision and Control*, Los Angeles, CA, 2014. Submitted.

[86] D. Richert and J. Cortés. Distributed linear programming with event-triggered communication. *SIAM Journal on Control and Optimization*, 2014. Submitted.

[87] C. J. Romanowski, R. Nagi, and M. Sudit. Data mining in an engineering design environment: OR applications from graph matching. *Computers and Operations Research*, 33(11):3150–3160, Nov. 2006.

[88] A. E. Roth. The evolution of the labor market for medical interns and residents: A case study in game theory. *Journal of Political Economy*, 92(6):991–1016, Dec. 1984.

[89] W. Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, 1953.

[90] W. Saad, Z. Han, M. Debbah, and A. Hjørungnes. A distributed coalition formation framework for fair user cooperation in wireless networks. *IEEE Transactions on Wireless Communications*, 8(9):4580–4593, 2009.

[91] W. Saad, Z. Han, Z. Debbah, M. Hjørungnes, and T. Başar. Coalitional game theory for communication networks: A tutorial. *IEEE Signal Processing Magazine, Special Issue on Game Theory*, 26(5):77–97, 2009.

[92] S. Samar, S. Boyd, and D. Gorinevsky. Distributed estimation via dual decomposition. In *European Control Conference*, pages 1511–1516, Kos, Greece, July 2007.

[93] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley, New York, 2000.

[94] B. Seanor, G. Campa, Y. Gu, M. Napolitano, L. Rowe, and M. Perhinschi. Formation flight test results for UAV research aircraft models. In *AIAA Intelligent Systems Technical Conference*, pages 1–14, Chicago, IL, Sept. 2004.

[95] S. Shakkattai and R. Srikant. Network optimization and control. *Foundations and Trends in Networking*, 2(3):271–379, 2007.

[96] W. F. Sharpe. A linear programming algorithm for mutual fund portfolio selection. *Management Science*, 13(7):499–510, 1967.

[97] E. D. Sontag. Further facts about input to state stabilization. *IEEE Transactions on Automatic Control*, 35:473–476, 1989.

[98] E. D. Sontag. Comments on integral variants of ISS. *Systems & Control Letters*, 34(1-2):93–100, 1998.

[99] W. C. Stirling and M. S. Nokleby. Satisficing coordination and social welfare for robotic societies. *Int. Journal of Social Robotics*, 1(1):53–69, 2009.

[100] J. Trdlicka, Z. Hanzalek, and M. Johansson. Optimal flow routing in multihop sensor networks with real-time constraints through linear programming. In *IEEE Conf. on Emerging Tech. and Factory Auto.*, pages 924–931, 2007.

[101] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9):803–812, 1986.

[102] M. J. Vachon, R. Ray, K. Walsh, and K. Ennix. F/A-18 aircraft performance benefits measured during the autonomous formation flight. In *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, Monterey, CA, Aug. 2002. Paper 4491.

[103] L. Vig and J. A. Adams. Multi-robot coalition formation. *IEEE Transactions on Robotics*, 22(4):637–649, 2006.

[104] P. Wan and M. D. Lemmon. Event-triggered distributed optimization in sensor networks. In *Symposium on Information Processing of Sensor Networks*, pages 49–60, San Francisco, CA, 2009.

[105] C. Wang, X. Hong, X. Ge, G. Zhang, and J. Thompson. Cooperative MIMO channel models: A survey. *IEEE Communications Magazine*, 48(2):80–87, 2010.

[106] J. Wang and N. Elia. A control perspective for centralized and distributed convex optimization. In *IEEE Conf. on Decision and Control*, pages 3800–3805, Orlando, Florida, 2011.

[107] X. Wang and M. D. Lemmon. Self-triggered feedback control systems with finite-gain L_2 stability. *IEEE Transactions on Automatic Control*, 54(3):452–467, 2009.

[108] X. Wang and M. D. Lemmon. Event-triggering in distributed networked control systems. *IEEE Transactions on Automatic Control*, 56(3):586–601, 2011.

[109] E. Wei and A. Ozdaglar. Distributed alternating direction method of multipliers. In *IEEE Conf. on Decision and Control*, pages 5445–5450, Maui, HI, 2012.

[110] H. Weimerskirch, J. Martin, Y. Clerquin, P. Alexandre, and S. Jiraskova. Energy saving in flight formation. *Nature*, 413(6857):697–698, 2001.

[111] R. J. B. Wets. On the continuity of the value of a linear program and of related polyhedral-valued multifunctions. *Mathematical Programming Study*, 24:14–29, 1985.

[112] G. Yarmish and R. Slyke. A distributed, scalable simplex method. *Journal of Supercomputing*, 49(3):373–381, 2009.

[113] M. Zargham, A. Ribeiro, A. Ozdaglar, and A. Jadbabaie. Accelerated dual descent for network flow optimization. *IEEE Transactions on Automatic Control*, 59(4):905–920, 2014.

[114] Z. Zhaoyang, S. Jing, C. Hsiao-Hwa, M. Guizani, and Q. Peiliang. A cooperation strategy based on Nash bargaining solution in cooperative relay networks. *IEEE Transactions on Vehicular Technology*, 57(4):2570–2577, July 2008.

[115] K. Zhou, J. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice Hall, Englewood Cliffs, NJ, 1995.

[116] M. Zhu and S. Martínez. On distributed convex optimization under inequality and equality constraints. *IEEE Transactions on Automatic Control*, 57(1):151–164, 2012.

[117] M. Zhu and S. Martínez. An approximate dual subgradient algorithm for distributed non-convex constrained optimization. *IEEE Transactions on Automatic Control*, 58(6):1534–1539, 2013.