Decentralized Coordination Control for Dynamic Multiagent Systems

by

Karla Kvaternik

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Electrical and Computer Engineering
University of Toronto

Abstract

Decentralized Coordination Control for Dynamic Multiagent Systems

Karla Kvaternik
Doctor of Philosophy
Graduate Department of Electrical and Computer Engineering
University of Toronto
2015

Engineered networks such as power grids, communication networks and logistics operations are becoming increasingly interdependent and comprised of intelligent components that are capable of autonomously processing and influencing their local environment, as well as interacting with other such components. These emerging capabilities are poised to enable unprecedented efficiency gains across previously disparate application domains. Fully harnessing this potential is contingent in part on our ability to effect coordinated local interactions among such components, ensuring their optimal collective behavior.

In this thesis we formulate several Decentralized Coordination Control Problems (DCCPs) for networked multiagent systems involving either static or dynamic agents. We propose that Consensus Optimization (CO) methods constitute a viable foundation for the synthesis of coordination control strategies that address large classes of DCCPs.

In order to understand both the potential and the limitations of such strategies, we develop a new framework for the convergence analysis of general CO schemes. In contrast with existing analytic approaches, our framework is based on powerful system-theoretic techniques enabling the study of DCCP scenarios in which the agent dynamics interact with those of the CO algorithm itself, thereby affecting its performance. Aside from fulfilling its intended purpose, our analytic viewpoint also leads to the relaxation of several standard assumptions imposed on numerical CO algorithms, and suggests a methodical approach to deriving conditions under which existing centralized optimization methods can be decentralized.

In addition, we propose the Reduced Consensus Optimization (RCO) algorithm – a streamlined variant of CO which is more innately suited to the DCCP context. Agents implementing RCO generally need not be aware of the network size and topology, and the overall processing and communication overhead associated with agent coordination may be substantially reduced relative to CO. On the basis of RCO, we propose a methodology for the design of decentralized content caching strategies in information-centric networks.

Dedication

I dedicate this thesis Maureen Jackson, who showed me kindness and humanity.

Acknowledgements

I am grateful to my supervisor, L. Pavel, for proofreading my work, funding my participation at several conferences, and for giving me ample freedom to pursue my own research interests. I would like to thank Professors M. Maggiore, T. Davison and J. Liebeherr for providing valuable feedback that has improved the quality of this thesis. I would also like to thank my external examiner, Professor A. Nedić, for reading the thesis and pointing out important directions for future research. The work in Chapter 7 was done in collaboration with J. Llorca at Bell Labs, and I am grateful for our informative discussions. I acknowledge the generous financial support I received from the NSERC Vanier fund.

Contents

List of Figures

Nomenclature

ACHR Node-averaged cache hit ratio, page 141

AHT Average hops travelled, page 141

CCS Content caching strategy, page 123

CDN Content delivery network, page 123

CHR Cache hit ratio, page 141

CO Consensus optimization, page 19

CT-SDCCP Continuous-time, static DCCP, page 16

DCCP Decentralized coordination control problem, page 7

DOpt decentralized optimization, page 43

DPE Decentralized parameter estimation, page 20

DT-DDCCP Discrete-time, dynamic DCCP, page 13

DT-SDCCP Discrete-time, static DCCP, page 15

ESC Extremum-seeking control, page 93

ICN Information-centric networking, page 123

LFU Least frequently used, page 140

NCC Network caching cost, page 141

NMAS Networked multiagent systems, page 6

NTC Network transport cost, page 140

NUM Network utility maximization, page 18

OSNR Optical signal-to-noise ratio, page 92

RCO Reduced consensus optimization, page 114

SPAS Semiglobally, practically asymptotically stable, page 38

Chapter 1

Introduction

1.1 The Vision

ANTEATER: [Aunt Hillary] is certainly one of the best-educated ant colonies I have ever had the good fortune to know. The two of us have spent many a long evening in conversation on the widest range of topics.

ACHILLES: I thought anteaters were devourers of ants, not patrons of ant intellectualism!

ANTEATER: Well, the two are not mutually inconsistent. I am on the best of terms with ant colonies. It's just ANTS that I eat, not colonies... [And,] I beg your pardon. "Anteater" is not my profession; it is my species. By profession, I am a colony surgeon. I specialize in correcting nervous disorders of the colony by the technique of surgical removal.

ACHILLES: Oh, I see. But what do you mean by "nervous disorders" of an ant colony?

ANTEATER: Most of my clients suffer from some sort of speech impairment. You know, colonies which have to grope for words in everyday situations. It can be quite tragic. I attempt to remedy the situation by, uhh – removing – the defective part of the colony.

ACHILLES: Well, I can vaguely see how it might be possible for a limited and regulated amount of ant consumption to improve the overall health of a colony – but what is far more perplexing is all this talk about having conversations with ant colonies. That's impossible. An ant colony is simply a bunch of individual ants running around at random looking for food and making a nest.

ANTEATER: You could put it that way if you want to insist on seeing the trees but missing the forest, Achilles. In fact, ant colonies, seen as wholes, are quite well-defined units, with their own qualities, at times including the mastery of language.

ACHILLES: There must be some amazingly smart ants in that colony, I'll say that.

ANTEATER: I think you are still having some difficulty realizing the difference in levels here. Just as you would never confuse an individual tree with a forest, so here you must not take an ant for the colony. You see, all the ants in Aunt Hillary are as dumb as can be. They couldn't converse to save their little thoraxes!

ACHILLES: Well then, where does the ability to converse come from? It must reside somewhere inside the colony! I don't understand how the ants can all be unintelligent, if Aunt Hillary can entertain you for hours with witty banter.

TORTOISE: It seems to me that the situation is not unlike the composition of a human brain out of neurons. Certainly no one would insist that individual brain cells have to be intelligent beings on their own, in order to explain the fact that a person can have an intelligent conversation.

(*The Ant Fugue*, [66][1])

This excerpt from Hofstadter's delightfully orchestrated *Ant Fugue* dialogue strikes at the core of our ongoing fascination with the phenomena of "emergence" and "complexity". Though one would be hard pressed to find universally accepted, formal definitions of these terms, it is easy to find examples of systems both in nature and engineering that are unambiguously characterized as being complex, and exhibiting emergent properties. Roughly speaking, a system is considered to be "complex" if it consists of a large number of interacting components, and the nature of these interactions plays an important role in the behavior of the whole. Such systems

[1]This excerpt has been shortened and slightly rearranged in the interest of brevity. Consequently, it cannot possibly reflect all of the aesthetic qualities and philosophical depth of the original Ant Fugue, which ought to be savoured in its entirety!

exhibit "emergent" properties if the collective behavior of these interacting components is in some sense distinctly more sophisticated than the behavior of the individual components themselves [106]. As Hofstadter's Ant Fugue so aptly illustrates, it is often difficult to reconcile the holistic view of such systems with the reductionist understanding of their constituent parts; "more" is indeed different [5].

Though most real ant and termite colonies may not be quite as charming as Aunt Hillary, they are nonetheless capable of accomplishing some impressive collective feats. It is interesting to contemplate that an ant colony, when regarded as a cohesive whole, behaves very much like an organism unto itself. From "birth" it grows, and upon reaching a certain stage of maturity, it reproduces and thereafter maintains a roughly constant size. Colonies compete for resources with nearby colonies, just like other macro organisms do. Remarkably, the colony appears to "know" when and how to pool and divide its workforce among competing priorities such as foraging and nest maintenance, and it does so adaptively in response to changing environmental conditions [58]. When foraging, the pathway that emerges between the colony and the food source is often the shortest possible, making it seem as though the colony has directly reached out to the food source.

In addition, many ant and termite species build large, elaborate nesting structures that serve strikingly sophisticated functions. The African *Macrotermitinae* termites construct "cathedrals" featuring carefully architected networks of ventilation ducts that aid in thermal regulation and promote the exchange of respiratory gasses [82]. Inside the mound one finds structural support pillars and a network of chambers, arranged according to function; some chambers are designated as nurseries, and others for the purpose of cultivating fungi that help feed the colony. It is astonishing that such a sophisticated structure, which can be up to a hundred thousand times larger than the average termite that participates in its construction [17], emerges out of simplistic actions and local interactions among thousands of individuals, none of whom are individually capable of cognizing the apparent aims of the collective, or the contribution that their individual actions make to its behavior. The entire construction process is accomplished in the absence of centralized control. There are no blueprints, and there is no organizational hierarchy.

Slime mold is another intriguing, and yet more basic example of how interactions among primitive modules leads to emergent, "superorganism" behavior [52]. The cellular slime mold *Dictyostelium discoideum* spends most of its life as a single-celled, amoebae-like organism, until food becomes scarce. The shortage of food triggers a process by which individual amoebae spontaneously aggregate into colonies of thousands of individuals, which then behave in many ways like a single organism.[2] So far, biologists have accumulated a substantive understanding of the mechanisms governing the behaviors and capabilities of individual cells. For example, it is now well known that individual amoebae interact with one another by periodically broadcasting a chemical abbreviated as cAMP, and that this primitive form of communication plays a critical role in the aggregation process [18]. The aggregation process itself involves several stages featuring emergent collective phenomena. For instance, in an initial stage, individual cells appear to spontaneously differentiate into a small number of "master" cells, which initiate the cAMP pulses autonomously, and a large number of "slave" cells, which relay those pulses. Curiously, the designation of "master" or "slave" does not appear to be an intrinsic, a priori property of individual cells, since it has been observed that in fixed-density populations, the fraction of master cells is inversely proportional to the size of the population; in other words, the emergence of sepcialization among cells appears to be "decided" by the amoeba collective [52].

[2] In the world of slime molds, the popularity of Dictyostelium is overshadowed by that of its relative, *Physarum polycephalum*, also known as true slime mold. Physarum has recently garnered fame for being able to solve mazes [111] and design optimal transport systems [151], as well as being a picky eater [46]. Though it accomplishes all these things in the absence of a centralized coordination center, in the present work Physarum is relegated to a footnote because it aggregates early on, and thereafter spends most of its life as a single, giant, multinucliate cell. As an example system, Physarum is therefore less germane to the advancement of our thesis than the humble Dictyostelium.

Our third example of emergent phenomena comes from the study of animal collectives that engage in herding, schooling or flocking. An interesting hypothesis is that banding together greatly enhances the ability of individual animals in such collectives to acquire and process information about their environment [36]. For example, in bird flocks and fish schools, individuals tend to align the velocity and direction of their motion with that of their neighbors, which tends to result in collective behavior that appears highly cohesive and well coordinated [158]. By the same mechanisms, an abrupt change in the velocity of an individual that has just perceived a predatory threat can propagate very quickly throughout the collective and be perceived by a distant conspecific for whom the threat is not even within sensory range [37].

It is postulated that in addition to extending an individual's effective sensory range, being part of a collective provides individuals with access to sophisticated signal processing capabilities [36]. For example, for the purposes of migration, environmental signals such as temperature or resource gradients must be accurately estimated in the presence of various sensory uncertainties. Models suggest that individuals within migrating collectives temper their response to personal measurements of environmental signals with how they perceive their neighbors to be responding [120]. This process is not unlike that of distributed averaging, which is often used in sensor networks to diminish the effects of noise [154]. Thus, the action of the collective is effectively to "filter" noisy environmental signals in a way that improves the accuracy of the collective estimate. Equally interesting is that such animal collectives appear to "know" how to adapt their collective "filter tuning" based on context. For example, the processing of information related to predatory threats involves filtering that is tuned to be more sensitive to small variations in individuals' movements than the filtering involved in the processing of migration-related information [37]. Questions into how exactly animal collectives are able to adapt the trade-off between the speed and the accuracy of signal estimation remain largely unanswered at this time [37].

These three examples of naturally occurring multiagent systems share at least seven quintessential features of particular interest to the engineering standpoint:

(F1) The system is composed of numerous basic modules or agents, each having a limited capacity to sense, process and affect its environment.

(F2) Each agent is able to interact in basic ways with a small subset of other agents – typically those in its physical proximity. The nature of these interactions seems to play an important role in shaping the collective behavior.

(F3) The ongoing interactions among agents produce a collective behavior which often appears to be in the best interest of the collective. In other words, the collective, when regarded as a cohesive whole, appears to behave in a purposive way and act in its own best interest.

(F4) Each individual agent has little or no awareness of the state, size or behavior of the collective at any given time, and is almost certainly nescient with respect to its apparent aims.

(F5) An agent's individual actions suggest that its private goals are not necessarily coincident with those of the collective.

(F6) The functional capabilities of the collective are surprisingly sophisticated relative to those of the individual agents that comprise it.

(F7) There is no centralized coordination mechanism, no hierarchical organization, and no commonly available blueprint to guide the actions of individuals in producing the behavior of the collective.

Many more examples of both natural and artificial multiagent systems exhibiting these features have been studied by various research communities [118] [63], [168], [146], [142]. The fact that such systems actually exist – and appear to function rather effectively within their various ecological milieus – poses an irresistible temptation to reverse-engineer them, to uncover the fundamental mechanisms that give rise to their emergent properties, and to attempt synthesizing them.

The observation that features F1 to F7 appear to be common to such a wide variety of existing systems fuels the hope that there may actually be a set of fundamental mechanisms that operate universally to produce their emergent properties. However, the question of whether such mechanisms exist has not been conclusively resolved yet, despite many notable and diverse efforts to understand self-organizing behavior [118], [66], [63], [156][3]. Even if such universal mechanisms do not exist – or are not discernible – one still wonders to what extent it might be possible, even for restricted problem scenarios, to identify those ingredients necessary for emergent behavior to occur.

Though much may be known about the operation of slime molds, ant colonies and bird flocks, the true goal is to move beyond the development of models that mimic specific instances of such systems; the ultimate vision is to know exactly how to build and manipulate general classes of networked multiagent systems that embody features F1 to F7, and to develop unifying theoretical frameworks that would allow us to understand their full potential, as well as their functional limitations. Of particular interest are features F4, F6 and F7, and the possibility of *engineering* systems in which the interactions among basic, nescient elements can be maximally exploited to create collective functionalities that extend far beyond those of the individual elements themselves.

In developing the unifying theoretical frameworks engendered by this vision, there are many possible perspectives (q.v. [50], for example). One such framework might allow us to specify the system by its set of agents – their operational capabilities and constraints, their interaction network and its dynamics – and from these specifications enable us to methodically answer several key questions. For instance, given a collective objective – i.e., some arbitrary desired collective behavior or configuration – does there exist a set of local interaction rules by means of which the agents may collectively achieve this objective? If there is at least one such set of interaction rules, how do we go about constructing them? Can we characterize the set of achievable collective objectives, and how do we do so? Can we characterize the range of environmental conditions and perturbations under which a given set of interaction rules will nonetheless achieve a given collective objective?

Such questions are primarily of a structural nature, echoing those concerning the "controllability" and "observability" of dynamical systems, for example. Owing to the import of algebraic concepts into the theory of feedback control, such questions now have complete, systematic answers within the realm linear dynamical systems [77], and to some extent also within the realm of nonlinear dynamical systems [69]. However, analogous theoretical developments in the broad context of networked multiagent systems are still sparse and disparate, being mostly focused on specialized system classes or collective objectives. We are still missing a comprehensive theory that would enable the systematic design and analysis of coordination control methods for general classes of such systems, and for general classes of collective objectives.

With apologies to the reader (who is no doubt deeply engaged, and with very high hopes at this point), we confess that no such comprehensive theory is developed in this thesis. However, our work – present and future – is certainly motivated by the prospect of developing such frameworks, and their potential utility in enabling new solutions to important societal challenges. Our aim in this thesis is to contribute toward the overall vision just described.

[3]Interestingly, this is the very same Alan Turing known for the Turing machine and several monumental contributions to theoretical computer science.

1.1.1 What Motivates this Vision?

Apart from being intellectually gratifying, the pursuit of this vision is of tremendous practical value. Its general relevance is suggested by recent social and technological trends away from centralized, monolithic, vertically integrated organizational structures, to those that are more lateral and decentralized. In [130], Rifkin collects a number of compelling examples evidencing this shift, which, not surprisingly, has been advanced by the inception of the Internet. The proliferation of social media within the last decade and a half has made it easier for collectives of individuals to organize without centralized coordination, and under politically oppressive conditions, to initiate the Arab spring revolutions, for example. The freely available Wikipedia, which compiled and reviewed distributively by a multitude of contributors, rivals the accuracy and quality of the best commercially available encyclopaedias [57] whose creators centralize the compilation and review process among a comparably small number hired experts. The blogosphere and social media have made it possible for consumers to become prosumers of news and information, and to bypass many products sold by monolithic news corporations. Similarly, consumers of media are being morphed into prosumers by Peer-to-Peer (P2P) file sharing systems and services such as YouTube. This process is undermining the traditional, monolithic business practices of large record labels and film production companies. Companies like Airbnb are making it easy for anyone to be a hotelier, without having to aggregate a large, centralized capital investment typically needed for a profitable entry into the hospitality industry. Along with the Internet, technologies such as 3D printing are promoting the so called “distributed manufacturing” movement, characterized by the increasingly many manufacturing activities taking place in geographically dispersed microfactories, bringing the production of many goods closer to the end consumer [26], [35].

Rifkin identifies the most important changes along this direction to be taking place in the energy sector, where it is expected that increasingly many passive consumers will begin to actively participate in the generation of energy by means of small-scale installations of renewables. Rifkin, among others (q.v. §2.2 in [157], for example), foresees an “energy Internet”, in which almost everyone is an energy prosumer, and energy flows are traded and coordinated in a decentralized way reminiscent of P2P file sharing. In parts of North America and the European Union, some aspects of the so-called “Smart Grid” are already well developed and actively deployed. For instance, the use of smart meters in the Pacific Northwest Smart Grid Demonstration project allowed consumers to monitor and schedule their energy consumption in response to real-time fluctuations in energy prices, resulting in substantial reductions in peak energy demands [1]. On the supply side however, the large-scale integration of distributed energy sources into the grid remains a technological challenge; it is difficult to coordinate the generation and flow of energy aggregated from a multitude of intermittent and geographically distributed sources in such a way that supply is matched to demand, and network stability and power quality are reliably maintained [137].

Equally as interesting is the notion of an “Internet of Things” (IoT), and the possibilities it represents; it is anticipated that the evolution of the IoT will eventually lead to the merger of the communication Internet, the energy Internet, and various logistics and transportation networks into one seamless, easily accessible network [130], [157]. The IoT, a concept which is being heeded rather seriously by leading industries[4], is envisioned as consisting of countless Internet-enabled sensors and actuators attached to various machines, appliances, buildings, vehicles, goods and people. The IoT will thus form enormous, highly interconnected, global networks of heterogeneous objects which are equipped with various basic sensory, processing and actuation capabilities, and which can communicate and interact with one another via the Internet.

[4]See for instance IBM’s “Smart Planet” initiative, CISCO and NASA’s “Planetary Skin”, GE’s “Industrial Internet”, Apple’s “HomeKit” developer tools, Bosch’s IoT Labs, or Siemens’ “Decentralized Energy Management System”, among others.

The pervasive presence of sensors collecting unimaginable quantities of streaming data, and of geographically distributed actuators that can be used to respond to insights mined from this data, together pose the intriguing possibility of fine-grained, *joint* control and *joint* optimization of networked systems spanning previously disjoint application domains. One may imagine energy coordination decisions being made based on weather forecasts and real-time traffic patterns, or supply chain management decisions being made based in part on trends mined from local streams of social media metadata. If managed correctly, these technological trends may truly culminate in an "Industrial Revolution" with unprecedented potential for improving the efficiency and reliability of a wide variety of interdependent social, economic and industrial processes. Though somewhat speculative, a captivating projection is that our overall social and technological trends toward decentralization have the potential to result in such efficiency gains as may ultimately lead to a near "zero marginal cost society", structured primarily as a "collaborative commons" [131].

As utopian as this vision might seem, there are a number of serious ethical, political and technological challenges to its realization. In our view, there are two predominant technological issues. The first relates to the sheer quantity of information that would conceivably be gathered by networks of geographically distributed sensors, and our ability to mine meaningful insights from this "Big Data" in real-time. Considering that the aggregation of data at a centralized processing station inevitably incurs delays and communication overhead, it seems unlikely that centralized solutions to this challenge would be feasible for every application that is yet to be dreamt up.

The second issue concerns the effective use of geographically distributed actuation points within this "Internet of Everything", and motivates the objectives of this thesis. The nature of the IoT as it is presently envisioned suggests that many of its yet unborn applications will precipitate numerous engineering problems that can be abstracted as coordination control problems for networked multiagent systems (NMAS). Some of these problems will involve networks of cooperative agents whose actions will need to be coordinated in order to achieve some collective, network-level configuration or performance objective. In analogy to problems associated with aggregating large quantities of data, one faces the problem of disseminating centrally-computed coordination control signals to each agent in the network. Since the stability and robustness of nearly every feedback control scheme is adversely affected by communication errors and delays, it is of utmost importance to minimize the distance and the number of hops that any control-related signal traverses. For these reasons it is also unlikely that centralized, or even hierarchical coordination control strategies alone would suffice for all future applications (q.v. Remark 1.1.1). In this sense, the realization of an energy Internet or an IoT-enabled "smart planet" depends not only on the evolution of business models and government policies, but also crucially on the development of engineering technologies such as decentralized coordination control algorithms for NMAS.

Whether a near-zero marginal cost society materializes or not, we can at least be certain that the accelerated evolution of information and communication technologies witnessed in the last two decades is bound to effect fascinating new engineering problems, many of which presently lie beyond our imaginations. Many of these problems are likely to be related to the regulation and management of complex systems, and as control theorists, we are uniquely positioned to contribute to their solutions. Unquestionably, exciting new developments lie ahead for the field of control theory.

Remark 1.1.1 (On Hierarchical vs. Decentralized Designs)**:** Even when a hierarchical control structure is most appropriate [102], it may often be the case that modules within a given layer of the hierarchy could benefit from coordinating their actions in a decentralized way. For instance in the regulation of islanded microgrids, the bottom layer may be identified with the primary control scheme responsible for regulating the behavior of individual loads and microgeneration units [78]. The upper layer of the control hierarchy may be identified with the secondary control scheme, which is responsible for coordinating power flows within the microgrid and regulating the frequencies and voltages at its point of common coupling [61]. As is typically the case in hierarchical control schemes [76], there is a time-scale separation between the fast primary control and the slow secondary control. For this example, one may consider decentralizing the implementation of the upper layer secondary control scheme in this two-layer hierarchy [93], [101]. $\diamondsuit$

1.1.2 Progress Within the Control Community

There are a number of somewhat segregated bodies of literature concerned broadly with the subject of decentralized coordination control for cooperative multiagent systems. From our preceding observations, it is not surprising that this topic is presently of great interest [2] within the control community.

The most recent and prolific body of literature in this genre focuses on the design of coordination control strategies for mobile robotics applications [138], [9], [25], [129], [128], [59], [87], [110], [141]. Problems of interest include formation control, the rendezvous problem, target localization and tracking, source seeking, gradient climbing and optimal area coverage, among others. Many of the contributions in this area are intended to enable mobile sensor network technologies aimed at end applications such as environmental monitoring, military reconnaissance, search and rescue, and space exploration. As such, the problem settings considered also often address distributed parameter estimation and sensor fusion problems. A focal consideration for many contributions in this literature is the communication structure that models the information flow among agents, and its effect on the collective behavior. Consequently, the well-studied consensus dynamics [71], [108], [136] play an important role in most of these results.

Within the control community, the earliest interest in decentralized coordination control appears in the late 1960s, and we refer to the associated body of literature as "classical decentralized control" [76], [170], [161], [162], [40], [164], [41], [104]. Early developments in classical decentralized control appear to have been motivated primarily by aerospace, power system and industrial control applications in which the plant model may involve hundreds or thousands of dynamically coupled state variables, and multiple control inputs and measured outputs. Many contributions in this area are concerned with questions of how to reduce a single, very large-scale dynamical system [6], or decompose it into smaller, weakly coupled subsystems that can be separately controlled by standard state, or output feedback methods [40], [170]. In contrast with the aforementioned developments in coordination control for mobile robotic applications, classical decentralized control is less focused on exploiting communication among separate controllers in order to coordinate their actions[5].

Other bodies of control literature related to our subject include networked control, which focuses on how communication channel properties affect closed-loop performance [12], and distributed model predictive control, which is an on-line, optimization-based technique successfully employed in applications such as industrial process control, traffic congestion control, and supply chain management [117], [34].

1.2 Decentralized Coordination Control

Motivated by the ideas in §1.1, we are interested in developing general, yet concrete and practical theories that address what we will refer to as the *decentralized coordination control problem* (DCCP). We will refine the

[5]This contrast can be explained by observing that information and communication technologies experienced accelerated development only after the late 1980s. Still, the possibility of exploiting communication among individual controllers to improve performance was not entirely ignored in the early days of classical decentralized control; the famous Witsenhausen counterexample shows that for some systems, certain decentralized reformulations of optimal control fail to perform as well as an ad-hoc strategy that entails implicit communication between two controllers, with the system's coupled dynamics themselves as the communication medium [76], [166].

definition of this problem in later sections, but in essence, the problem can be stated as follows.

Problem 1.2.1 (General DCCP)**:** For a given set of agents, design a set of rules by which agents individually operate and interact in order to produce a desired collective behaviour, or accomplish some non-trivial task in a way that leverages the synergism of the collective.

In other words, we consider cooperative problem scenarios in which individual agents are either devoid of private interests, or their capacity to act in pure self-interest and game the system for their own advantage is assumed to be removed by certain technological constraints or governing policies. Examples of problem scenarios involving agents without private interests include those of engineered systems such as networks operated by a single internet service provider, or a generation, transmission and distribution network owned by a single utility, or a sensor network of mobile robots deployed for a common purpose. On the other hand, an example of a multiagent system in which cooperation might be enforcible by policy or technology is an islanded microgrid in which there are restrictions imposed on energy-trading practices among participants.

In this thesis, we approach the DCCP from a control theoretic point of view. Consequently, we restrict the set of all conceivable problem scenarios that fall under Problem 1.2.1, to those characterized by the following features.[6]

P1 (**Agent Communication**)**:** There is a system of N agents, indexed by the set $\mathscr{V} = \{1,\dots,N\}$. In analogy to feature F2 in §1.1, each agent may interact with some subset of other agents. We let $\mathscr{E}_C \subseteq \mathscr{V} \times \mathscr{V}$ specify those agent pairs that can at any time exchange information directly with one another, and let $\mathscr{N}_C(i)$ denote the set of agents that agent i may communicate with – i.e., $j \in \mathscr{N}_C(i)$ iff $(j,i) \in \mathscr{E}_C$. We refer to $\mathscr{G}_C = (\mathscr{E}_C, \mathscr{V})$ as the *communication graph*, and we consider scenarios in which $\mathscr{G}_C$ is given in the formulation of the DCCP. $\diamondsuit$

P2 (**Agent Actions**)**:** The set of *actions*, or *decisions* available to each agent $i \in \mathscr{V}$ can be represented by an m_i-dimensional vector of real values, denoted as u_i. We refer to $u = [u_1^T,\dots,u_N^T]^T \in \mathscr{U} \subset \mathbb{R}^m = \mathbb{R}^{m_1} \times \cdots \times \mathbb{R}^{m_N}$ as the *collective decision*, and to $\mathscr{U}$ as the *collective decision space*. $\diamondsuit$

P3 (**Agent States**)**:** The *state* of the i'th agent (or its proximal environment) can be specified by finitely many real numbers collected into $x_i \in \mathbb{R}^{n_i}$, and modelled as being either static, or evolving according to a given system of continuous-time, deterministic, ordinary differential equations. We refer to $x = [x_1^T,\dots,x_N^T]^T \in \mathbb{R}^n = \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_N}$ as the *collective state*, or the *collective configuration*.

Naturally, we assume that an agent's actions affect its state. If agent i is modelled as a static entity, then its state is understood to be synonymous with its decision – i.e., $x_i \equiv u_i$, and $n_i = m_i$.[7] If agent i is modelled as a dynamic entity, then we let Σ_i denote its dynamics, which we assume to be given by

$$\Sigma_i : \Big\{ \dot{x}_i = f_i(x, u_i), \quad \forall i \in \mathscr{V}, \tag{1.1}$$

where $f_i : \mathbb{R}^n \times \mathbb{R}^{m_i} \to \mathbb{R}^{n_i}$ is locally Lipschitz continuous. We often refer to the collective dynamics (1.1) as Σ.

We note that in many practical applications, the state of one agent may influence the evolution of another's state. Allowing $f_i(\cdot, u_i)$ to depend generally on the collective state x reflects this possibility. $\diamondsuit$

P5′ (**The Collective Objective**)**:** We assume that the desired collective behavior or configuration of the multiagent system corresponds to a set $X^* \subset \mathbb{R}^n$ of collective states. We refer to X^* as the *goal set*, and we say that the collective objective is achieved whenever $x \in X^*$. $\diamondsuit$

---

[6]Notation which is not explicitly defined throughout the text may be found in Appendix A. In addition, Chapter 2 provides a brief exposition of mathematical preliminaries that are relevant to our developments.

[7]We emphasize this point because in those chapters of this thesis dealing with static agents we denote agent i's decision by x_i.

P6′ (**Agent Nescience**): In analogy to feature F4 in §1.1, an agent may not generally have direct knowledge of the goal set X^*, of other agents' individual performance measures, their states, their actions, and the environmental conditions affecting them. Consequently, individual agents are nescient with respect to the collective objective and the overall progress of the agent collective toward achieving it. $\diamondsuit$

We intend to express properties P5′ and P6′ more precisely in the sequel. Before we can do so, we introduce the notion of an *equilibrium map*, and property P4, which depends on it.

1.2.1 Asymptotic Decentralized Coordination Control

In consideration of properties P1-P2, and P5′-P6′, the remainder of this section is dedicated to the further refinement Problem 1.2.1, and a gradual exposition of our intended approach to solving it. Clearly, the setting so far requires us to design state, or output-feedback laws for the agents' control (i.e. decision) variables u, in order to drive the collective state x toward the goal set X^*.

We wish to point out that the formulation thus far is general enough to accommodate at least two distinct approaches to the said design. One approach is to consider the problem form an asymptotic equilibrium (set) stabilization point of view – that is, design u such that x approaches some subset of X^* asymptotically. However, this approach is only possible when the set X^* is *compatible* with the given collective dynamics Σ. We explain what we mean by this with the following example.

Example 1.2.1 (Compatibility of X^* with Σ): Consider a system in which $\mathscr{V} = \{1\}$, $x = x_1 = [x_{1,1}, x_{1,2}]^T \in \mathbb{R}^2$, and

$$\hat{\Sigma}_1 : \begin{cases} \dot{x}_{1,1} = x_{1,2} \\ \dot{x}_{1,2} = v_1 \end{cases}$$

Suppose that $X^* = [3,4] \times [1,2]$. Clearly, no possible state feedback could ever change the structure of Σ_1 such that some $x^* \in X^*$ becomes a stabilizable equilibrium. On the other hand, if $X^* = [3,4] \times [-1,2]$, we may consider a control law of the form $v_1 = -x_{1,1} - kx_{1,2} + u_1$, where k is any positive, real number, and u_1 is an auxiliary control variable. This control law renders any point of the form $[u_1, 0]^T$, an asymptotically stable equilibrium for the closed-loop system. Then, supposing that X^* is *known*, letting u_1 take any value in the interval $[3,4]$ accomplishes the objective asymptotically: $\lim_{t\to\infty} x(t) \in X^*$. We will say that $X^* = [3,4] \times [-1,2]$ is *compatible* with $\hat{\Sigma}_1$, while $X^* = [3,4] \times [1,2]$ is not. $\diamondsuit$

More generally, we say that a set X^* is compatible with a dynamic system Σ if there exists some static or dynamic state feedback control law that renders some subset of X^* asymptotically stable for Σ. Clearly then, the compatibility of X^* and Σ is necessary in order to approach the DCCP from the equilibrium (set) stabilization point of view.

In the absence of compatibility, one may alternatively approach the DCCP by recasting it as a *reach-control problem*. Specifically, the goal would be to design u such that the collective state enters X^* in finite time, and thereafter continually visits, or re-visits certain states within X^* – perhaps those that require minimal control effort to move between. The work pursued in [24] could be relevant here, though the assumed agent nescience (q.v. property P6′) may present an interesting challenge to this approach.

Equilibrium Maps and the Compatibility of X^* with Σ

In this thesis we assume that the given X^* is compatible with the given collective dynamics Σ, and we approach the DCCP from an equilibrium (set) stabilization point of view. In Example 1.2.1, we designed the static state feedback control in two stages: the *primary*, or inner-loop control is given by $v_1 = -x_{1,1} - kx_{1,2} + u_1$, and its task is to stabilize any point of the form $[u_1, 0]^T$. The *secondary*, or outer-loop control task is to design u_1 so as to guide the state of $\hat{\Sigma}_1$ toward X^*.

In this thesis we focus on the design of the secondary control laws, which is an important task in many NMAS applications, such as power grids (q.v. Remark 1.1.1). Specifically, in problems involving dynamic agents, we consider scenarios in which the collective dynamics Σ have an asymptotically stable *equilibrium map*, which is defined as follows.

Definition 1.2.1 (Equilibrium Map)**:** Consider a system of the form

$$\Sigma : \begin{cases} \dot{x} = f(x,u), \end{cases}$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$. A function $l : \mathbb{R}^m \to \mathbb{R}^n$ is an *equilibrium map* for Σ if $f(x,u) = 0$, iff $x = l(u)$, for some $u \in \mathbb{R}^m$. $\diamondsuit$

With this definition, we further refine our problem scenarios by means of the following property.

P4 (**Asymptotically Stable Equilibrium Map and Compatibility with** X^*)**:** We assume that there exists an equilibrium map $l : \mathbb{R}^m \to \mathbb{R}^n$ for the collective dynamics Σ in (1.1), such that for every $u \in \mathscr{U}$, the point $l(u)$ is asymptotically stable for Σ. Moreover, $l(\mathscr{U}) \cap X^* \neq \emptyset$. $\diamondsuit$

The closed-loop system

$$\Sigma_1 : \begin{cases} \dot{x}_{1,1} = x_{1,2} \\ \dot{x}_{1,2} = -x_{1,1} - kx_{1,2} + u_1 \end{cases}$$

from Example 1.2.1 satisfies P4. Its equilibrium map is given by $l(u_1) = [u_1, 0]^T$, and is clearly such that $l(\mathbb{R}) \cap X^* \neq \emptyset$. We note that requiring P4 to hold is stronger than requiring X^* to be compatible with Σ; clearly, both Σ_1 and $\hat{\Sigma}_1$ are compatible with $X^* = [3,4] \times [-1,2]$.

We are now ready to make statements P5′ and P6′ more precise.

1.2.2 Agent Performance, Measurements, and the Collective Objective

For DCCPs characterized by P4, we may assume that an inner-loop control has already been designed. In that case, the collective decision u represents a reference input which is to be adjusted by individual agents in a coordinated manner, in order to guide the equilibrium state of the collective dynamics toward the desired configuration X^*.

When X^* is explicitly known (as was supposed in Example 1.2.1), the solution to the DCCP might be trivial under P4. However, in many applications of interest, the set X^* may not be known *a priori*, and may need to be discovered online (q.v. P6′). This may happen when the desired collective state configuration depends on certain external environmental conditions encountered by the agent collective in real-time. As in the naturally occurring multiagent systems described in §1.1, an added challenge is that each agent may be subjected to a unique set of local environmental conditions, so that the form of X^* is something that must emerge from the ongoing interactions among the agents.[8]

In order for this to be possible, it is natural to assume that each agent has access to some measure of its individual performance, which may in general depend on its state, its actions, and the local environmental conditions affecting it. An agent's performance may also be affected by the states and actions of other agents. The

[8]It is interesting to note that the same species of termite, when placed in two different environments with distinct climatic conditions, will produce mounds with distinctly different features. These features are are in fact adapted for better functionality in the climatic conditions encountered by the termite collective [65].

collective objective then ought to relate to the maximization of some measure of group performance, which is naturally assumed to be some function of agents' individual performance measures. In analogy to feature F5 in §1.1, an agent's individual performance is not necessarily maximized when the collective objective is met; the fact that one agent's actions may interfere with another's performance implies that some degree of coordination and compromise is generally needed in order for the collective objective to be achieved.

One way to express a broad variety of NMAS coordination tasks is to appeal to the formalism of optimization. In particular, finite-dimensional convex programming offers a rich assortment of tools and techniques, and covers an impressive diversity of applications [22], [15]. For these reasons, we consider problem settings in which the collective objective can be encoded as a solution to a convex program. The "recognition that many (if not all) optimization problems of interest can be cast as convex programs" [127] suggests the generality, and thence the utility of this approach.

We therefore consider DCCP scenarios with the following features.

P5 (Agent Performance and The Collective Objective): Let $d_i(t) \in \mathbb{R}^{p_i}$ denote the collection of exogenous signals representing all relevant environmental conditions affecting agent i, and let $d(t) = [d_1(t)^T, \dots, d_N(t)^T]^T \in \mathbb{R}^p$.

For the case in which the agents are dynamic entities, the measure of agent i's performance is given by the value of a function $J_i : \mathbb{R}^n \to \mathbb{R}$, which is parametrized by $d_i(t)$. We assume that the goal set is given by

$$X^*(t) = l(U^*(t)), \tag{1.2}$$

where $l : \mathbb{R}^m \to \mathbb{R}^n$ is the equilibrium map associated with Σ, and

$$U^*(t) = \arg\min_{u \in \mathscr{U}} \sum_{i \in \mathscr{V}} J_i(l(u); d_i(t)). \tag{1.3}$$

We refer to

$$J(l(u); d(t)) = \sum_{i \in \mathscr{V}} J_i(l(u); d_i(t)) \tag{1.4}$$

as the *collective cost*, and to $J_i(l(u); d_i(t))$ as agent i's *individual cost*. We assume that the collective cost $u \mapsto J(l(u); d(t))$ is convex for each $d(t) \in \mathbb{R}^p$.

For the case in which the agents are static entities, the measure of agent i's performance is given by the value of a function $J_i : \mathbb{R}^m \to \mathbb{R}$, which is parametrized by $d_i(t)$. In this case the goal set is given by

$$X^*(t) \equiv U^*(t) = \arg\min_{u \in \mathscr{U}} J(u; d(t)), \tag{1.5}$$

where

$$J(u; d(t)) = \sum_{i \in \mathscr{V}} J_i(u; d_i(t)). \tag{1.6}$$

As in the dynamic case, we assume that the collective cost $u \mapsto J(u; d(t))$ is convex in u for every $d(t) \in \mathbb{R}^p$. $\diamondsuit$

Remark 1.2.1: In expressing the collective objective according to (1.2), we ensure that X^* is compatible with the collective dynamics Σ, and that we may pursue our DCCPs of interest from the point of view of asymptotic (set) stabilization. Intuitively, (1.2) and (1.3) imply that we are looking for the best collective state configuration that corresponds to an equilibrium for the dynamics (1.1). Alternatively, we may regard (1.3) as a joint optimization over $x \in \mathbb{R}^n$ and $u \in \mathscr{U}$, subject to the constraint $x = l(u)$. $\diamondsuit$

Remark 1.2.2 (Relationship to Game Theory)**:** It is worth noting that the problem settings described so far have a close resemblance to multiagent problems that can be abstracted as continuous-kernel games [11]. In particular, a *noncooperative, continuous-kernel game* is specified by a set $\mathscr{V}$ of players, the ith player being characterized by an *action variable* u_i and a convex cost $J_i(u_1,\ldots,u_N)$. Each agent's objective is to minimize its own cost, without regard for other agents' costs. The solution concept of interest is that of the *Nash equilibrium*, which is some point u^* in the collective decision space $\mathscr{U}$. The Nash equilibrium $u^* = [u_1^{*T},\ldots,u_N^{*T}]^T$ is comprised of agents' individual actions u_i^*, each of which constitutes a *best response* to all other agents' actions. In other words, u^* is such that $\forall i \in \mathscr{V}$, u_i^* minimizes $u_i \mapsto J_i(u_1^*,\ldots,u_{i-1}^*,u_i,u_{i+1}^*,\ldots,u_N^*)$, so that no individual agent has any incentive to unilaterally deviate from u^*.

The fact that agent's individual cost functions are *coupled* (i.e. the actions of some agents may affect others' costs) means that in general, the set of Nash equilibria associated with the game $(\mathscr{V},\mathscr{U},\{J_1(\cdot),\ldots,J_N(\cdot)\})$ does not correspond to the set of social optima U^* in (1.5). Indeed, it is well known that decision processes designed to locate a Nash equilibrium of a given game may not lead to socially optimal configurations. In particular, for some games the sum of agent costs can be arbitrarily large at a Nash equilibrium. This phenomenon is often referred to as *the price of anarchy*, or *the tragedy of the commons* [84], and it is an inherent consequence of purely self-interested decision making. The potential social inefficiency of Nash equilibria is regarded as a downside to using game-theoretic decision updating processes as a means of achieving coordination in multiagent systems. However in some applications, game-theoretic decision updating methods are favoured because they require no communication among agents, and thus incur little or no coordination overhead [47]. $\diamondsuit$

In light of the specification of the goal set X^* given in P5, property P6$'$ can be made more precise as follows.

P6 (**Agent Nescience**)**:** Agent i has no direct knowledge of the quantities $J_j(\cdot;d_i(t))$, u_j, $d_j(t)$, x_j and Σ_j, for any $j \neq i$. Therefore, agent i has no direct knowledge of $J(\cdot;d(t))$, $U^*(t)$, $l(\cdot)$ and X^*. $\diamondsuit$

Remark 1.2.3: Properties P5 and P6 reflect features F5 and F4 in §1.1. We note that the minimizers of agents' individual cost functions may not generally coincide; a collective decision that is optimal for one agent may be costly to another (c.f. Remark 1.2.2). Moreover, a collectively optimal decision $u^* \in U^*$ may not generally minimize any of the $J_i(\cdot,\cdot;d_i(t))$ individually. $\diamondsuit$

Though agent nescience is an important characteristic of many DCCP scenarios, some assumptions need to be made on what information *is* available to each agent.

P7 (**Agent Measurements**)**:** In either the static or dynamic case, agent i is assumed to have access to measurements of *some* quantity which is related to its private cost $J_i(\cdot;d_i)$. Let $y_i \in \mathbb{R}^{r_i}$ represent the collection of variables that can be measured by agent i at any time, and let y_i be given by

$$y_i = h_i(J_i(x;d_i(t)), \tag{1.7}$$

where $h_i(\cdot)$ is some operator for which there may be a number of relevant specifications, depending on the application under consideration. $\diamondsuit$

Remark 1.2.4 (Examples of Measured Quantities)**:** There are various ways that the operator $h_i(\cdot)$ in (1.7) may be specified, depending on the application being considered, and the knowledge typically available to each agent in that application. Among others, possible specifications for $h_i(\cdot)$ include the following.

- $h_i(J_i(x;d_i(t)) = J_i(x;d(t))$.
- $h_i(J_i(x;d_i(t)) = \nabla_x J_i(x;d_i(t))$.
- $h_i(J_i(x;d_i(t)) = \nabla_u J_i(l(u);d_i(t))\big|_{l(u)=x}$.

◊

Together, problem features P1 to P7 allow us to express more precisely a subset of the DCCPs represented by Problem 1.2.1.

1.2.3 Decentralized Coordination Control Problem Formulations

Narrowing our focus to problem scenarios characterized by P1 to P7, we present three specific DCCP formulations that we address throughout this thesis. We separate the problem formulations pertaining to scenarios involving dynamic agents, from those involving static agents. We also consider formulations involving candidate coordination control algorithms that operate in either discrete-time or continuous-time.

Dynamic Decentralized Coordination Control Problems

In Problem 1.2.2 below, we formulate a DCCP involving a sampled-data feedback interconnection between the continuous-time collective dynamics Σ in (1.1), and a discrete-time decision updating process.

Problem 1.2.2 (Discrete-Time Dynamic Decentralized Coordination Control Problem (DT-DDCCP))**:** Consider a set of mulitagent DCCP scenarios characterized by features P1 to P7, in which a given networked multiagent system is modelled according to

$$\Sigma_i : \begin{cases} \dot{x}_i = f_i(x, u_i), \quad \forall i \in \mathscr{V} \\ y_i = h_i(J_i(x; d_i(t)). \end{cases} \tag{1.8}$$

Also consider a candidate coordination control algorithm of the form

$$\mathscr{D}_i : \begin{cases} \xi_i(t_{k+1}) = \phi_i(\xi_i(t_k), (v_j(t_k))_{j \in \mathscr{N}_C(i)}, Y_i(t_{k+1}^-)), \quad \forall i \in \mathscr{V}, \\ u_i(t) = \eta_i(\xi_i(t_k)), \quad t \in [t_k, t_{k+1}), \quad \forall k \in \mathbb{N}, \\ v_i(t_k) = \psi_i(\xi_i(t_k), Y_i(t_{k+1}^-)), \end{cases} \tag{1.9}$$

in which $\xi_i \in \mathbb{R}^{q_i}$ represents the internal state of the ith agent's decision updating process, $v_i(t) \in \mathbb{R}^{l_i}$ are its *communicated variables*, $(v_j(t_k))_{j \in \mathscr{N}_C(i)}$ denotes an ordered set of all communicated variables received by agent i, and

$$Y_i(t_{k+1}^-) = \begin{bmatrix} y_i(t_{k_1}) \\ \vdots \\ y_i(t_{k_{s_i}}) \end{bmatrix} \tag{1.10}$$

represents a collection of measurements made by agent i within the time interval $[t_k, t_{k+1})$.

Then, the *discrete-time, dynamic decentralized coordination control problem* (DT-DDCCP) is to design, if possible, for all $i \in \mathscr{V}$, the relations $\phi_i(\cdot)$, $\psi_i(\cdot)$ and $\eta_i(\cdot)$ such that corresponding to X^*, there is a nonempty, compact set $\Xi^* \subset \mathbb{R}^q$ (with $q = \sum_{i \in \mathscr{V}} q_i$), and $X^* \times \Xi^*$ is *semiglobally, practically, asymptotically stable* for the closed-loop system (1.8)-(1.9). ◊

Figure 1.1 depicts a networked multiagent system and the relevant properties of dynamic DCCPs such as Problem 1.2.2.

Remark 1.2.5: The variables v_i collect all the information that agent i communicates to its neighbors on $\mathscr{G}_C$. The option to design the relation $\psi_i(\cdot)$ is meant to reflect the desirability of communicating sparingly, the importance of choosing what is to be communicated, or the possibility that some information may be deemed private. ◊

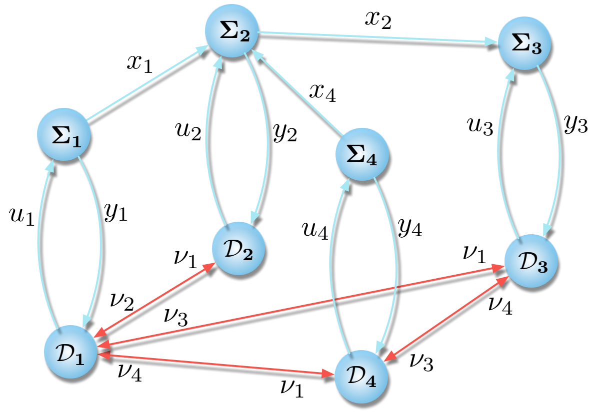

Figure 1.1: Dynamic decentralized coordination control problem scenarios. The connections between the decision updating processes $\mathscr{D}_i$ represent edges in the communication graph $\mathscr{G}_C$, while connections among the dynamic subsystems Σ_i signify the presence of general dynamic coupling.

Remark 1.2.6: The notion of semiglobal, practical, asymptotic stability (SPAS) is defined in the upcoming §2.5. For now, it suffices to note that the SPAS concept is more general than the more familiar concept of global asymptotic stability (GAS). In the case of Problem 1.2.2 for example, SPAS contrasts with GAS in not requiring $X^* \times \Xi^*$ to be identical with the set of equilibria associated with (1.8)-(1.9). Roughly speaking, SPAS implies that for an arbitrarily large set of initial conditions, trajectories of (1.8)-(1.9) can be made to asymptotically approach an arbitrarily small set containing $X^* \times \Xi^*$. $\diamondsuit$

To solidify the notions thus far introduced, we consider the following basic example.

Example 1.2.2 (DT-DDCCP for Force-Actuated Point Masses)**:** As an extension of Example 1.2.1, consider a multiagent system in which $\mathscr{V} = \{1,2\}$, and agent i is modelled according to

$$\Sigma_i : \begin{cases} \dot{x}_{i,1} = x_{i,2} \\ \dot{x}_{i,2} = -x_{i,1} - k_i x_{i,2} + u_i \\ y_i = h_i(J_i(x; d_i(t))). \end{cases} \tag{1.11}$$

The collective state here is given by $x = [x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}]^T \in \mathbb{R}^4$, and the collective decision by $u = [u_1, u_2]^T \in \mathbb{R}^2$. System Σ_i might represent the motion of a force-actuated point of unity mass, free to move along a line. In that case, $x_{i,1}$ represents particle i's position, and $x_{i,2}$ its velocity. Suppose that the collective objective is to adjust the distance between the particles in some optimal way in response to certain environmental conditions characterized by four numbers, say $d_i(t) = [a_i(t), b_i(t)]^T \in \mathbb{R}^2$, $i \in \mathscr{V}$. Specifically, suppose that the collective cost is given by $J(x;d) = J_1(x;d_1) + J_2(x;d_2)$, where

$$J_i(x;d) = \tfrac{1}{2} a_i (x_{1,1} - x_{2,1} - b_i)^2, \tag{1.12}$$

and we have dropped the time-dependence notation.

From Example 1.2.1 and the discussion that followed, we know that the equilibrium map pertaining to Σ_i is given by $l_i(u_i) = [u_i, 0]^T$. Therefore, the equilibrium map $l : \mathbb{R}^2 \to \mathbb{R}^4$ for the collective dynamics Σ is given by

$$l(u) = \begin{bmatrix} u_1 \\ 0 \\ u_2 \\ 0 \end{bmatrix}. \tag{1.13}$$

Based on (1.3) then,

$$\begin{aligned} U^* &= \underset{u\in\mathbb{R}^2}{\arg\min}\ J(l(u);d) \\ &= \underset{u\in\mathbb{R}^2}{\arg\min}\ \left(\tfrac{1}{2}a_1(u_1-u_2-b_1)^2 + \tfrac{1}{2}a_2(u_1-u_2-b_2)^2\right) \\ &= \{u \in \mathbb{R}^2 \,|\, u_1 - u_2 = \tfrac{a_1b_1+a_2b_2}{a_1+a_2}\}. \end{aligned} \tag{1.14}$$

The corresponding goal set X^* is given by (1.2) as

$$X^* = l(U^*) = \{x \in \mathbb{R}^4 \,|\, x_{1,1} - x_{2,1} = \tfrac{a_1b_1+a_2b_2}{a_1+a_2},\ x_{1,2} = x_{2,2} = 0\}. \tag{1.15}$$

In words, the DT-DDCCP in this example is to design the decision updating processes $\mathscr{D}_i$ such that the distance between the particles is adjusted until it reaches a weighted average of the numbers b_1 and b_2, and the particles' velocities are zeroed. $\diamondsuit$

Static Decentralized Coordination Control Problems

In practice, one also often encounters *static* coordination control problems. In such problems, individual agents have access to measurements related to their individual performance, but these measurements are not affected by any dynamic quantities. One possible formulation of a static DCCP is given in the following statement, and may be regarded as a special case of Problem 1.2.2.

Problem 1.2.3 (Discrete-Time, Static Decentralized Coordination Control Problem (DT-SDCCP))**:** Consider a set of mulitagent DCCP scenarios characterized by features P1 to P7, in which a given networked multiagent system is modelled according to

$$\Sigma_i : \begin{cases} x_i \equiv u_i, & \forall i \in \mathscr{V} \\ y_i = h_i(J_i(x;d_i(t)). \end{cases} \tag{1.16}$$

Also consider a candidate coordination control algorithm $\mathscr{D}_i$ of the form (1.9), in which ξ_i, $v_i(t)$, $(v_j(t_k))_{j\in\mathscr{N}_C(i)}$ and $Y_i(t_{k+1}^-)$ are as described in Problem 1.2.2.

Then, the *discrete-time, static decentralized coordination control problem* (DT-SDCCP) is to design, if possible, for all $i \in \mathscr{V}$, the relations $\phi_i(\cdot)$, $\psi_i(\cdot)$ and $\eta_i(\cdot)$ such that for some nonempty, compact set $\Xi^* \subset \mathbb{R}^q$ (with $q = \sum_{i\in\mathscr{V}} q_i$), Ξ^* is *semiglobally, practically, asymptotically stable* for the dynamics (1.9), and Ξ^* relates to X^* in (1.5) as follows: for each $\xi^* = [\xi_1^{*T}, \dots, \xi_N^{*T}]^T \in \Xi^*$,

$$u^* \equiv x^* = \begin{bmatrix} \eta_1(\xi_1^*) \\ \vdots \\ \eta_N(\xi_N^*) \end{bmatrix} \in X^*. \tag{1.17}$$

$\diamondsuit$

Example 1.2.3: Consider a static version of the DCCP described in Example 1.2.2. This problem involves two abstract agents, each endowed with a private cost function of the form (1.12) (with $x_i \equiv u_i$). Agent i is capable of manipulating only its own decision variable u_i, and has knowledge of its private cost function by means of its measured variable $y_i = h_i(J_i(u; d_i(t)))$. Agent i has no direct knowledge of the other agent's private cost, or its decision at any time. The agents' collective objective is to coordinate their decisions until the difference between them coincides with a weighted average of the numbers b_1 and b_2 (i.e., $u \in U^*$, where U^* is as in (1.14)). The DT-SDCCP in this case is to design a set of decision updating processes $\mathscr{D}_i$ that (practically) achieve this objective. $\diamondsuit$

In some applications it may be more natural to consider decision updating processes that evolve in continuous-time. In that case, one may formulate the static coordination control problem as follows.

Problem 1.2.4 (Continous-Time, Static Decentralized Coordination Control Problem (CT-SDCCP))**:** Consider a set of mulitagent DCCP scenarios characterized by features P1 to P7, in which a given networked multiagent system is modelled according to (1.16). Also consider a candidate coordination control algorithm of the form

$$\mathscr{D}_i : \begin{cases} \dot{\xi}_i = \phi_i(\xi_i, (v_j)_{j \in \mathscr{N}_C(i)}, y_i), \quad \forall i \in \mathscr{V}, \\ u_i = \eta_i(\xi_i), \\ \nu_i = \psi_i(\xi_i, y_i). \end{cases} \tag{1.18}$$

Then, the *continuous-time, static decentralized coordination control problem* (CT-SDCCP) is to design, if possible, for all $i \in \mathscr{V}$, the relations $\phi_i(\cdot)$, $\psi_i(\cdot)$ and $\eta_i(\cdot)$ such that for some nonempty, compact set $\Xi^* \subset \mathbb{R}^q$ (with $q = \sum_{i \in \mathscr{V}} q_i$), Ξ^* is *semiglobally, practically, asymptotically stable* for the dynamics (1.18), and Ξ^* relates to X^* in (1.5) as follows: for each $\xi^* = [\xi_1^{*T}, \ldots, \xi_N^{*T}]^T \in \Xi^*$,

$$u^* \equiv x^* = \begin{bmatrix} \eta_1(\xi_1^*) \\ \vdots \\ \eta_N(\xi_N^*) \end{bmatrix} \in X^*. \tag{1.19}$$

$\diamondsuit$

In a similar way it is straightforward to formulate a continuous-time version of the dynamic coordination control problem specified in Problem 1.2.2.

Remark 1.2.7 (On notation.)**:** Taking x_i to be identical to the decision u_i in Problems 1.2.3 and 1.2.4 is a notational convenience that allows us to maintain consistency with notation typically found in the literature. In those chapters that address static coordination control problems we omit reference to Σ or u, and we express the discrete-time decision updating processes $\mathscr{D}_i$ as

$$\mathscr{D}_i : \begin{cases} y_i(t) = h_i(x(t); d_i(t))), \quad \forall i \in \mathscr{V}, \\ \xi_i(t_{k+1}) = \phi_i(\xi_i(t_k), (v_j(t_k))_{j \in \mathscr{N}_C(i)}, Y_i(t_k)), \\ x_i(t) = \eta_i(\xi_i(t_k)), \quad t \in [t_k, t_{k+1}), \quad \forall k \in \mathbb{N}, \\ \nu_i(t_k) = \psi_i(\xi_i(t_k), Y_i(t_k)), \end{cases} \tag{1.20}$$

where all variables are as previously defined. More compactly still, we omit reference to the time index, and express (1.20) as

$$\mathscr{D}_i : \begin{cases} y_i = h_i(x;d_i)), \quad \forall i \in \mathscr{V}, \\ \xi_i^+ = \phi_i(\xi_i,(v_j)_{j\in\mathscr{N}_C(i)},Y_i), \\ x_i = \eta_i(\xi_i), \\ v_i = \psi_i(\xi_i,Y_i). \end{cases} \tag{1.21}$$

The notation for continuous-time decision updating processes involved in static coordination control problems is similarly adapted. ◇

One approach to the synthesis problem for decentralized coordination control could be based on existing methods for decentralized optimization.

1.3 Decentralized Optimization

Let us consider a static optimization problem of the form

$$\min_{u\in\mathbb{R}^m} \sum_{i=1}^{N} J_i(u), \tag{1.22}$$

where $u = [u_1^T,\dots,u_N^T]^T$, and $u_i \in \mathbb{R}^{m_i}$ for each i.There are a number of methods available to solve problems of this form in a decentralized way, aimed at dividing the workload among a number of networked processors. Recent interest in the development of such optimization methods is driven primarily by emerging technologies such as cloud computing and distributed sensor networks, and the corresponding necessity to distributively process very large datasets [103], [21], [127], [153], [145].

Though we are interested in coordination control more so than parallel processing, we submit that such methods have several attributes that are desirable in the context of DCCPs. Since such methods play an important role in our approach to decentralized coordination control design, we proceed to review two prominent classes thereof – decomposition-based methods, and consensus-based methods.

Decomposition-Based Methods

The idea of decomposing an optimization problem into a number of smaller subproblems is more than fifty years old [39]. Two outstanding techniques for doing so are *primal decomposition*, which leverages the fact that one can optimize first over a subset of the optimization variables, and then optimize the outcome over the remaining variables, and *dual decomposition*, which leverages strong duality and relies on Lagrangian relaxation methods. In addition to the usual "trickery" (i.e., introduction of auxiliary variables, changes of coordinates, transformation of objective and constraint functions, etc. [22]), these two techniques can be combined in a number of different ways to yield all sorts of different problem decompositions and associated distributed optimization algorithms [121]. Quite often, especially if primal decomposition is used, one ends up with a hierarchical problem structure involving a *master problem*, which is typically solved on a slower time-scale, and a number of smaller *auxiliary*, or *slave problems*, which are solved on a faster time-scale.[9]

Decomposition-based methods are famous for having been successfully applied to the so-called *network utility maximization* (NUM) problems [31], [99], [64], [48], starting with the seminal work of Kelly [79]. A canonical

[9]It is interesting to note that decomposition methods in the context of optimization appear to have heavily inspired some of the prominent work on hierarchical control theory [102].

example of a NUM problem is congestion control in TCP/IP networks by means of dual decomposition, though the techniques applied are relevant to a much broader class of resource allocation problems. A characteristic feature of such problems is that the costs $J_i(\cdot)$ are *separable* – that is, $J_i(\cdot)$ is a function only of the variable u_i. However, there is typically a *coupling constraint* that prevents the problem from being directly solved in parallel by N separate processors. For this reason, numerical solutions are often applied to the dual problem, which can be separated. The congestion control problem in its most basic form is discussed in detail in §4.5 of this thesis.

In solving problems such as (1.22), in which the costs $J_i(\cdot)$ are *coupled*, one may proceed to decompose the problem by introducing auxiliary variables $\xi_i \in \mathbb{R}^m$, which are partitioned exactly like u – i.e.,

$$\xi_i = [\xi_{i,1}^T, \ldots, \xi_{i,N}^T]^T, \tag{1.23}$$

where $\xi_{i,j} \in \mathbb{R}^{m_j}$. Then, letting $\xi = [\xi_1^T, \ldots, \xi_N^T]^T$, problem (1.22) can be equivalently written as

$$\min_{\xi \in \mathbb{R}^{Nm}} \sum_{i=1}^{N} J_i(\xi_i) \tag{1.24}$$

$$\text{s.t. } \xi_1 = \cdots = \xi_N. \tag{1.25}$$

Components u_i of the vector u that appear in any function $J_j(\cdot)$, $j \neq i$ are called *complicating variables*, and (1.25) are referred to as *consistency constraints* [21]. By re-writing problem (1.22) as (1.24)-(1.25), we have effectively decoupled the costs $J_i(\cdot)$ since we may regard ξ_i as agent i's private copy of u. We then relax the consistency constraints by incorporating them into the Lagrangian function $\mathscr{L}(\xi, \mu)$, where μ is the dual variable associated with (1.25).

One way to implement a *distributed* solution to (1.24)-(1.25) is to decompose the Lagrangian into additive components $\mathscr{L}_i(\xi_i, \mu)$ and run N parallel gradient (or subgradient) descent algorithms; agent i minimizes the function $\mathscr{L}_i(\cdot, \mu)$ for a fixed μ. Once each agent completes this process, it passes its solution $\xi_i^*(\mu)$ to a central *collector* station that uses this information to update the dual variable μ according to a gradient (or subgradient) ascent on $\mathscr{L}(\xi^*(\mu), \mu)$.

A more elegant approach is proposed in [163], where a fully decentralized solution is proposed by exploiting the observation that there is more than one way to express the consistency constraints (1.25). In particular, suppose that the N agents may exchange information over a graph $\mathscr{G}_C = (\mathscr{V}, \mathscr{E}_C)$. Then, (1.25) is satisfied whenever

$$(L \otimes I_n)\xi = \mathbf{0}, \tag{1.26}$$

where L is the graph Laplacian associated with $\mathscr{G}_C$. This fact is quite obvious to those who have previously worked with continuous-time consensus algorithms [136], because it is well known that the so-called "agreement subspace" coincides with the null space of L when $\mathscr{G}_C$ is connected. It turns out that a first-order primal-dual algorithm[10]

$$\xi^+ = \xi - \alpha \nabla_\xi \mathscr{L}(\xi, \mu) \tag{1.27}$$

$$\mu^+ = \mu + \alpha \nabla_\mu \mathscr{L}(\xi, \mu) \tag{1.28}$$

[10] The stability and convergence properties of such algorithms – whether in continuous-time [8], [119] or discrete-time [115] have been well established. Originally, Lagrange saddle-point dynamics evolving in continuous-time were studied by means of Lyapunov techniques, as early as 1958 [8].

on the Lagrangian

$$\mathscr{L}(\xi,\mu) = \sum_{i\in\mathscr{V}} J_i(\xi_i) + \mu^T (L\otimes I_n)\xi \tag{1.29}$$

can be rearranged in such a way that for all $i \in \mathscr{V}$, agent i updates exactly those primal and dual variables whose update equations depend exclusively on its own set of updated variables, and those being updated by its one-hop neighbors on $\mathscr{G}_C$. In other words, the updates are fully decentralized and the need for a central collector, or coordinator that sets “prices”, is bypassed.

In §4.5, we propose an alternative approach to decentralizing the coordinating role of a central collector. The approach we investigate in §4.5 is based on the application of a consensus-optimization method, which we describe next.

Consensus Optimization Methods

Consensus optimization (CO) methods originate in the early 1980s, in the doctoral work of Tsitsiklis [154], [155]. Tsitsiklis considers the problem of distributing the unconstrained numerical minimization of an objective function $J : \mathbb{R}^m \to \mathbb{R}$ among N processors, who are able to exchange information asynchronously over an undirected graph $\mathscr{G}_C$, which is required to be “connected in the long-run”. Though each processor has knowledge of the analytic structure of $J(\cdot)$, processors collaborate in estimating a minimizer u^* in order to mitigate the effects of various stochastic uncertainties afflicting the optimization process. In the deterministic version of the algorithm, agent i updates its estimate of an optimizer u^* according to

$$\xi_i^+ = \sum_{j=1}^{N} [A]_{i,j}\xi_j - \alpha\nabla J(\xi_i), \tag{1.30}$$

where α is a diminishing step-size and $A \in \mathbb{R}^{N\times N}$ is a specially weighted, and possibly time-varying adjacency matrix associated with $\mathscr{G}_C$. In contrast with the basic gradient descent algorithm, each processor implementing (1.30) takes into account its neighbors’ current estimates of u^* by combining them with its own estimate prior to taking a step in a direction minimizing $J(\cdot)$. Indeed, the *consensus term* $\sum_{j=1}^{N}[A]_{i,j}\xi_j$ represents a convex combination (i.e., a weighted average) of the ith processor’s estimate with those of its neighbors on $\mathscr{G}_C$. Under appropriate assumptions on the matrix A, the process of repeatedly averaging processors’ individual estimates in this way acts to bind them together, eventually leading to an agreement on u^*.

Two decades passed before the work of Tsitsiklis would be extended in several important ways that are of particular relevance to the DCCP [113], [114], [116]. In [113] and [114], Nedić et al generalize the problem setting in [154] such that each agent has private knowledge of its individual objective function $J_i(\cdot)$, and the collective objective is comprised as the sum of these component functions, much as in (1.4) and (1.6). In addition, the agents’ individual costs need not be differentiable, and the optimization problem may involve general convex constraints [116]. Specifically, the agents aim to cooperatively solve a problem of the form

$$\begin{aligned} \min_{u\in\mathbb{R}^m} \quad & \sum_{i\in\mathscr{V}} J_i(u) \\ \text{s.t.} \quad & u \in X \end{aligned} \tag{1.31}$$

where

$$X = \bigcap_{i\in\mathscr{V}} X_i, \tag{1.32}$$

is nonempty, and $\forall i \in \mathscr{V}$, $X_i \subseteq \mathbb{R}^m$ is a closed, convex set known only to agent i. In this new setting, agent i updates its estimate of a minimizer u^* for (1.31) according to the algorithm

$$\xi_i^+ = \mathbf{P}_{X_i}\Big[\sum_{j=1}^{N}[A]_{i,j}\xi_j - \alpha s_i(\xi_i)\Big], \tag{1.33}$$

where $\mathbf{P}_{X_i}(\cdot)$ is the orthogonal projection onto the set X_i, α is a diminishing step-size, and $s_i(\xi_i)$ is any subgradient of $J_i(\cdot)$ at ξ_i.

The matrix A in (1.33) is required to satisfy conditions very similar to those stipulated for (1.30), and plays exactly the same role – to bind the agent's individual estimates of u^* toward a consensus. Because of this, the problem being solved by (1.33) is more clearly expressed as

$$\begin{aligned} \min_{\xi \in \mathbb{R}^{Nm}} \quad & \sum_{i \in \mathscr{V}} J_i(\xi_i) \\ \text{s.t.} \quad & \xi_i \in X_i \\ & \xi_1 = \cdots = \xi_N, \end{aligned} \tag{1.34}$$

which is equivalent to problem (1.31). In other words, for the unconstrained case in which $X = \mathbb{R}^m$, algorithm (1.33) and algorithm (1.27)-(1.28) solve exactly the same problem.

Coordination Control vs. Parameter Estimation

A cannonical application problem for consensus optimization methods is decentralized parameter estimation (DPE) in sensor networks. A parameter-fitting problem can often be expressed as a convex program whose solution corresponds to the "best" parameter estimate. In the multiagent setting, a large quantity of parameter-fitting data is divided among N processors, the ith block of data corresponds to $J_i(\cdot)$, and the overall parameter-fitting problem corresponds to the minimization of $\sum_{i \in \mathscr{V}} J_i(\cdot)$.

On first glance it appears that consensus optimization aglorithms like (1.33) are ideally suited to solving static coordination control problems such as the DT-SDCCP (q.v. Problem 1.2.3). However, there is a subtle distinction between the class of problems that consensus optimization is intended to solve, and coordination control problems whose goal sets are given by (1.5). The distinction hinges on the interpretation of the optimization variables in question and a careful consideration of what is known and unknown to an agent in the two problem settings. This distinction is made apparent by attempting to recast a canonical consensus-optimization problem such as the DPE, as a static coordination control problem.

Example 1.3.1 (Recasting the DPE as a DT-SDCCP.)**:** Suppose we are given a parameter estimation problem that can be expressed as a strictly convex optimization problem, whose unique solution corresponds to the best parameter estimate:

$$\theta^* = \arg\min_{\theta \in \mathbb{R}^m} \sum_{i \in \mathscr{V}} J_i(\theta). \tag{1.35}$$

Suppose further that $J_i(\theta)$ is differentiable at each θ. This DPE can be cast as a static DCCP by identifying agent i's decision variable u_i with its private estimate of the sought parameter θ^* – i.e., $\forall i \in \mathscr{V}$, $u_i \in \mathbb{R}^m$, and the collective decision u is an element in $\mathbb{R}^{Nm}$.

In this case (1.5) takes the form

$$U^* = \{\theta^*\} = \arg\min_{u \in \mathscr{U}} \sum_{i \in \mathscr{V}} J_i(u_i), \tag{1.36}$$

where $\mathscr{U} = \{u \,|\, u_1 = \cdots = u_N\}$ (c.f. (1.31) and (1.34), with $X \equiv \mathbb{R}^m$).

Since the analytic structure of $J_i(\cdot)$ is typically known in parameter estimation applications, agent i is able to evaluate the quantity $\nabla_\theta J_i(\theta)|_{\theta=u_i}$, and thus implement an unconstrained version of algorithm (1.33) – i.e.,

$$u_i^+ = \sum_{i\in\mathscr{V}} [A]_{i,j} u_j - \alpha \nabla_\theta J_i(\theta)|_{\theta=u_i}. \tag{1.37}$$

The decision updating rule (1.37) corresponds to $\mathscr{D}_i$ in Problem 1.2.3 (q.v. expression (1.9)), with

$$\mathscr{D}_i : \begin{cases} y_i = \nabla_\theta J_i(\theta)|_{\theta=\xi_i} \\ \xi_i^+ = \sum_{i\in\mathscr{V}} [A]_{i,j}\xi_j - \alpha y_i \\ u_i = \xi_i, \\ v_i = \xi_i. \end{cases} \tag{1.38}$$

According to [114] then, under appropriate assumptions on A and α, for each $i \in \mathscr{V}$, $u_i \to \theta^*$. $\diamondsuit$

Remark 1.3.1: Decentralized parameter estimation is a canonical representative of the class of problems that CO methods are designed to solve *directly*, according to the update rule (1.37). In Example 1.3.1, we have shown how the DPE problem can be cast as a DT-SDCCP, and therefore, that the DPE corresponds to a class of DT-DCCPs involving goal sets that happen to be specified as minimizers of a sum of *separable* agent cost functions, and for which $\mathscr{U} = \{u \,|\, u_1 = \cdots = u_N\}$. From this, we conclude that the consensus-optimization algorithm (1.33) can be directly applied to solve all DT-SDCCPs involving goal sets that happen to be specified as minimizers of a sum of *separable* agent cost functions, and for which $\mathscr{U} = \{u \,|\, u_1 = \cdots = u_N\}$. However, in consideration of our formulation of Problem 1.2.3, these features identify a rather narrow (though nonempty[11]) class of DCCPs. $\diamondsuit$

1.4 Consensus Optimization: A Foundation for the Design of Decentralized Coordination Control

General static DCCPs represented by Problem 1.2.3 can be solved by a slightly modified version of the standard consensus optimization algorithm (1.33).

Example 1.4.1 (Recasting the DT-SDCCPs in terms of Consensus Optimization.)**:** Consider a class of DT-SDCCPs represented by Problem 1.2.3, in which the collective cost is given by $J(u) = \sum_{i\in\mathscr{V}} J_i(u)$, and in which each $J_i(\cdot)$ is strictly convex and differentiable at each collective decision $u \in \mathbb{R}^m$. Interpret ξ_i as agent i's "opinion" of what constitutes the optimal collective decision u^*, and $\xi_{i,j} \in \mathbb{R}^{m_j}$ as agent i's "suggestion" to agent j on how to behave. If agent i's measurement y_i (q.v. P7) is given by

$$y_i = \nabla_{\hat{u}} J_i(\hat{u})\big|_{\hat{u}=\xi_i}, \tag{1.39}$$

then the unconstrained version of the consensus optimization algorithm (1.33) can be directly employed to solve

[11] In Chapter 4 we consider an application of a continuous-time variant of (1.33) to a class of cooperative resource allocation problems in which agents must reach an agreement on an optimal resource "pricing" parameter. This problem can be regarded as an example of a CT-SDCCP.

the said DCCP – i.e., we may take

$$\mathscr{D}_i : \begin{cases} \xi_i^+ = \sum_{i\in\mathscr{V}} [A]_{i,j} \nu_j - \alpha y_i \\ \nu_i = \xi_i, \\ u_i = \xi_{i,i}, \end{cases} \tag{1.40}$$

and according to [114], under appropriate assumptions on A and α, for each $i \in \mathscr{V}$, $\xi_i \to u^*$. Clearly then, we also have that $u \to u^*$. $\diamondsuit$

Remark 1.4.1 (Comparison to Game Theoretic Methods)**:** With property P5, the problem settings characterizing the two static DCCPs specified in this chapter coincide with the specification of a continuous-kernel game, which is typically given as the triplet $(\mathscr{V}, \mathscr{U}, \{J_1(u), \ldots, J_N(u)\})$ (q.v. Remark 1.2.2). In contrast with game-theoretic methods such as *fictitious play* [134], the method (1.40) converges to the set of socially optimal collective decisions U^*, given by (1.5). Moreover, with the interpretation proposed in Example 1.4.1, the decision updating process $\mathscr{D}_i$ in (1.40) does not require agent i to have any knowledge of other agents' actions. $\diamondsuit$

Remark 1.4.2: In contrast with the setting of the DPE where $\nabla_{\hat{u}} J_i(\hat{u})\big|_{\hat{u}=\xi_i}$ is assumed to be measured (q.v. (1.38)), in some coordination control applications the quantity $\nabla_{\hat{u}} J_i(\hat{u})\big|_{\hat{u}=\xi_i}$ may not be accessible. For example, consider the case in which

$$y_i = \nabla_{\hat{u}} J_i(\hat{u})\big|_{\hat{u}=u}, \tag{1.41}$$

where $u = [u_1^T, \ldots, u_N^T]^T = [\xi_{1,1}^T, \ldots, \xi_{N,N}^T]^T$ is the collective decision, and neither the analytic structure of $\nabla J_i(\cdot)$, nor the value of $u_j,\ j \neq i$ are known to agent i – i.e., only the values of $\nabla_{\hat{u}} J_i(\hat{u})\big|_{\hat{u}=u}$ are being directly measured, or estimated from measurements of $J_i(u)$. Such a scenario might arise in CDMA (code division multiple access) power control problems, for example. In such problems, $J_i(u)$ relates to the signal-to-interference ratio experienced by user i, and u_i represents the uplink power applied by user i. [47]. In such a case, one may still employ the decision update rule (1.40) with y_i as in (1.41), and it can be shown that this slightly modified version of the standard consensus optimization algorithm (1.33) also converges (q.v. Chapter 5 in this thesis). $\diamondsuit$

Example 1.4.2 (CO-based $\mathscr{D}_i$ for Force-Actuated Point Masses)**:** Continuing with Example 1.2.2, we consider a CO-based coordination control design for the system of two force-actuated point masses on a line, modelled according to (1.11). First, suppose that the two agents may communicate with one another, and that the measurements available to agent i are given as

$$\begin{aligned} y_i &= h_i(J_i(x; d_i) \\ &= a_i(x_{1,1} - x_{2,1} - b_i), \end{aligned} \tag{1.42}$$

so that

$$\nabla_u J_i(l(u); d)\big|_{l(u)=x} = \begin{bmatrix} y_i \\ -y_i \end{bmatrix}, \tag{1.43}$$

where the equilibrium map $l(u)$ is given in (1.13) and $J_i(x; d)$ in (1.12). In that case, consider the following potential solution to the DT-DDCCP (q.v. Problem 1.2.2) for this example:

$$\mathscr{D}_i : \begin{cases} \xi_i(t_{k+1}) = \sum_{i\in\mathscr{V}} [A]_{i,j} \xi_j(t_k) - \alpha \hat{s}_i(y_i(t_{k+1}^-)) \\ u_i(t) = \xi_{i,i}(t_k), \qquad t \in [t_k, t_{k+1}),\ k \in \mathbb{N} \\ \nu_i(t_k) = \xi_i(t_k), \end{cases} \tag{1.44}$$

where $\xi_i(t_k) \in \mathbb{R}^2$, the weighted adjacency matrix is taken as

$$A = \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix}, \tag{1.45}$$

and the approximate search direction as

$$\hat{s}_i(y_i(t_{k+1}^-)) = \begin{bmatrix} y_i(t_{k+1}^-) \\ -y_i(t_{k+1}^-) \end{bmatrix} = \begin{bmatrix} a_i(x_{1,1}(t_{k+1}^-) - x_{2,1}(t_{k+1}^-) - b_i) \\ -a_i(x_{1,1}(t_{k+1}^-) - x_{2,1}(t_{k+1}^-) - b_i) \end{bmatrix}. \tag{1.46}$$

In this decision updating process, ξ_i may be interpreted as agent i's "opinion" of what constitutes the optimal control input $u^* = [u_1^*, u_2^*]^T$ for both agents. However, agent i only implements the ith component of ξ_i – i.e., $u_i = \xi_{i,i}$.

We refer to $\hat{s}_i(\cdot)$ as the "approximate" search direction, because it does not coincide with the the search direction $s_i(\cdot)$ in the standard CO algorithm (1.33). For this example, the search direction $s_i(\cdot)$ used in algorithm (1.33) would be given by

$$s_i(\xi_i(t_k)) = \nabla_u J_i(l(u);d)\big|_{u=\xi_i(t_k)} = \begin{bmatrix} a_i(u_1 - u_2 - b_i) \\ -a_i(u_1 - u_2 - b_i) \end{bmatrix}\Big|_{u_1=\xi_{i,1}(t_k),\, u_2=\xi_{i,2}(t_k)}, \tag{1.47}$$

and the direct knowledge thereof is not accessible to agent i for two reasons. First, the presence of the dynamics Σ_i (along with the assumed stability of $l(u)$ as in P4) implies that

$$\hat{s}_i(y_i(t_{k+1}^-)) \to \begin{bmatrix} a_i(u_1(t_k) - u_2(t_k) - b_i) \\ -a_i(u_1(t_k) - u_2(t_k) - b_i) \end{bmatrix} \tag{1.48}$$

only as $T = t_{k+1} - t_k$ tends to infinity. Second, even if the quantity

$$\begin{bmatrix} a_i(u_1(t_k) - u_2(t_k) - b_i) \\ -a_i(u_1(t_k) - u_2(t_k) - b_i) \end{bmatrix} \tag{1.49}$$

were directly available for measurement, agent i would not be able to evaluate this quantity at $u = \xi_i$ as required in (1.47) since u_j, $j \neq i$, is not a variable that agent i can manipulate. In Chapter 5 we show how one may nevertheless derive conditions guaranteeing the convergence of sampled-data systems such as (1.11)-(1.44). $\diamondsuit$

Example 1.4.3 (A Static DCCP with Unknown Gradients)**:** Continuing with Example 1.2.3, we consider a CO-based coordination control design for the system of two static agents whose private costs are given by (1.12). Suppose that the two agents may communicate with one another, and that the measurements available to agent i are given as

$$\begin{aligned} y_i &= h_i(J_i(u;d_i)) \\ &= J_i(u;d_i). \end{aligned} \tag{1.50}$$

If the agents are to implement a CO-based decision updating process, they must estimate the gradients of their private costs by means of some sequence of measurements. Consider for example the following potential solution to this DT-SDCCP (q.v. Problem 1.2.3):

$$\mathscr{D}_i : \begin{cases} \xi_i(t_{k+1}) = \sum_{i \in \mathscr{V}} [A]_{i,j} \xi_j(t_k) - \alpha \hat{s}_i(Y_i(t_{k+1}^-)) \\ u_i(t) = \xi_{i,i}(t_k), \qquad t \in [t_k, t_{k+1}), \ k \in \mathbb{N} \\ v_i(t_k) = \xi_i(t_k), \end{cases} \tag{1.51}$$

where $\xi_i(t_k) \in \mathbb{R}^2$, the weighted adjacency matrix is taken as in (1.45),

$$Y_i(t_{k+1}^-) = \begin{bmatrix} y_i(t_k) \\ y_i(t_k + \delta_t) \\ y_i(t_k + 2\delta_t) \end{bmatrix} \tag{1.52}$$

and the approximate search direction is given by

$$\hat{s}_i(Y_i(t_{k+1}^-)) = \begin{bmatrix} \frac{1}{\delta_u}(y_i(t_k + \delta_t) - y_i(t_k)) \\ \frac{1}{\delta_u}(y_i(t_k + 2\delta_t) - y_i(t_k)) \end{bmatrix}, \tag{1.53}$$

where $t_k < t_k + \delta_t < t_k + 2\delta_t < t_{k+1}$, δ_u and δ_t are some appropriately selected constants known to both agents, and

$$u(t_k + \delta_t) = \begin{bmatrix} u_1(t_k) + \delta_u \\ u_2(t_k) \end{bmatrix}, \quad \text{and} \quad u(t_k + 2\delta_t) = \begin{bmatrix} u_1(t_k) \\ u_2(t_k) + \delta_u \end{bmatrix}. \tag{1.54}$$

In words, inside every iteration of the CO-like scheme (1.51), the agents coordinate their actions in a prearranged way in order to perturb their coupled costs and thus obtain a forward-difference approximation of the required gradients. $\diamondsuit$

Decomposition vs. Consensus-Based Optimization Methods

Because both consensus and decomposition-based algorithms are capable of solving coupled problems of the form (1.22), both are amenable to the reinterpretation considered in Example 1.4.1. As such, either could form an acceptable foundation on which to build solutions to the various DCCPs formulated in the prequel.

In contrast with decomposition-based methods that result in algorithms of the form (1.27)-(1.28), algorithm (1.33) operates exclusively on the primal variables. Consequently, its dynamics are always simpler and of a lower dimension. On the other hand, combining decomposition techniques such as the ones described in §1.3 with techniques for augmenting the Lagrangian with appropriate penalty terms [21], [163] may result in algorithms with improved convergence properties.

Another distinction between decomposition and consensus-based algorithms is that under standard constraint qualifications, the former have a well-defined set of equilibria that can be characterized by the KKT conditions [22]. Consequently, the saddle-point dynamics (1.27)-(1.28) (at least in continuous-time) can be analyzed using Lyapunov techniques proposed nearly sixty years ago [8]. On the other hand, for the case in which the minimizers of agents' individual costs $J_i(\cdot)$ do not coincide, it is easy to verify that the set of fixed points associated with (1.33) would not generally include the point ξ^*, where ξ^* is a solution to the collective problem (1.34). In fact, even for the case in which $J_i(\cdot)$ is strictly convex and differentiable, an analytic expression for the actual fixed point of (1.33) may in general be tricky to obtain. This, combined with the (somewhat dated) view that the "Lyapunov approach is limited by the assumption that the equilibrium is known or that the isolated subsystem equilibria coincide with that of the interconnected system equilibrium" ([76], § III-C, p. 116) means for this author that consensus-based methods are simply begging to be analyzed from the point of view of *practical asymptotic stability*, and precisely by means of the Lyapunov approach!

Being unable to resist such a challenge (and also owing to reasons of a completely arbitrary nature), in this thesis we choose to study consensus optimization schemes – by means of the Lyapunov approach, of course.

1.5 A Summary of Contributions

The main proposition in this thesis is that consensus optimization methods constitute a viable class of decision updating processes that are able to solve large classes of both static and dynamic DCCPs. Specifically, we are proposing that one solution (1.9) to the DT-DDCCP defined in Problem 1.2.2 can be given by

$$\mathscr{D}_i : \begin{cases} \xi_i(t_{k+1}) = \phi_i(\xi_i(t_k), (v_j(t_k))_{j \in \mathscr{N}_C(i)}, Y_i(t_{k+1}^-)), \quad \forall i \in \mathscr{V}, \\ u_i(t) = \eta_i(\xi_i(t_k)), \quad t \in [t_k, t_{k+1}), \quad \forall k \in \mathbb{N}, \\ \nu_i(t_k) = \psi_i(\xi_i(t_k), Y_i(t_{k+1}^-)), \end{cases} \tag{1.55}$$

where

$$\phi_i(\xi_i(t_k), (v_j(t_k))_{j \in \mathscr{N}_C(i)}, Y_i(t_{k+1}^-)) = \sum_{i \in \mathscr{V}} [A]_{i,j} \nu_j(t_k) - \alpha \hat{s}_i(Y_i(t_{k+1}^-)) \tag{1.56}$$

$$\eta_i(\xi_i(t_k)) = \xi_{i,i}(t_k), \tag{1.57}$$

$$\psi_i(\xi_i(t_k), Y_i(t_{k+1}^-)) = \xi_i(t_k), \tag{1.58}$$

and $\hat{s}(Y_i(t_{k+1}^-))$ represents an estimate of the search direction $s_i(\cdot)$ in (1.33), which is constructed on the basis of the measurements Y_i made within the interval $[t_k, t_{k+1})$.

All of our separate contributions throughout this thesis are tied together by an effort aimed at assessing and improving the efficacy of such a solution.

With an ultimate focus on discrete-time, dynamic DCCPs represented by Problem 1.2.2, the majority of our efforts are aimed at developing new analysis techniques that allow us to study the interaction of numerical consensus optimization methods with continuous-time dynamical systems. These techniques are developed in Chapters 2 and 3, and they are applied to the setting of Problem 1.2.2 in Chapter 5. Chapter 4 demonstrates that the analytic approach proposed in Chapter 3 also facilitates the study of continuous-time consensus-optimization schemes, where such studies may be difficult by means of existing techniques. In Chapters 6 and 7, we test the applicability of our proposed coordination control philosophy to the problem of decentralized content caching in information-centric networks. In Chapter 7, we formulate this problem as a static, discrete-time DCCP, and we solve it using the *reduced consensus optimization* (RCO) algorithm. RCO is a streamlined version of algorithm 1.33 that we develop in Chapter 6. An extensive simulation study is presented in Chapter 7, in which we attempt to assess the utility of RCO in a setting involving 440 search variables.

To conclude this chapter, we provide a summary of the contributions made in this thesis. Most chapters contain a substantive literature review which is specialized to the topic being addressed. The introductory sections to the individual chapters thereby provide additional perspective on the contributions that we list here.

Chapter 3: An Analytic Framework for Consensus Optimization Methods: The Interconnected Systems Approach. As an initial step toward addressing the DT-DDCCP (q.v. Problem 1.2.2) by means of consensus optimization, we develop a new framework for the analysis of algorithms such as (1.33). In the setting of this chapter we are only concerned with static optimization problems of the form (1.31).

The essential idea of this chapter is based on [91], wherein it is proposed that consensus optimization schemes such as (1.33) can be studied from a system theoretic point of view. The novelty in our approach is to treat the evolution of the agents' mean estimate and the vector of deviations from this mean, as the feedback interconnection of two nonlinear, dynamical systems. Then, small-gain techniques are employed to derive conditions ensuring the internal stability (i.e. stability in the sense of Lyapunov) of the interconnection (q.v. Theorems 3.6.1 and 3.6.2).

We emphasize internal stability because it is the most fundamental property of relevance in control applications, and this emphasis distinguishes our analysis from that in most existing literature on decentralized optimization. One of the analytic tools we contribute in connection to this chapter is a theorem that characterizes the semiglobal, practical, asymptotic stability of a set of fixed points associated with a nonlinear discrete-time dynamical system, in terms of certain properties of its associated Lyapunov function (q.v. Theorem 2.5.1). We refer to this theorem as the SPAS theorem, and we provide its proof in Chapter 2.

Together, the interconnected systems viewpoint of consensus optimization along with the SPAS theorem constitute an analytic framework that allows us to study the dynamic sampled-data interaction of numerical consensus optimization methods with continuous-time dynamical systems, as in Problem 1.2.2. However, the application of this analytic framework has precipitated several observations that may be considered of independent interest to the area of decentralized optimization itself. These observations include the following.

- In the literature on consensus optimization, the effects of the dynamic coupling between the mean and deviation variables are usually suppressed by a combination of two persisting assumptions: the boundedness of agents' individual constraint sets (or the subgradients used to form their search directions), and the use of diminishing step sizes. Instead of suppressing the effects of dynamic coupling among subsystems, the literature on interconnected systems emphasizes techniques that seek to exploit this coupling in order to avoid conservatism [104]. Indeed, in our case a careful examination of the interconnection structure reveals that the destabilizing effects of the projection-related terms arising in one subsystem negate related effects arising in the other (q.v. Lemma 3.6.3). The conclusion to be drawn is that the presence of the projection operation in (1.33) need not add any conservatism to the final convergence condition.

- The individual constraint sets X_i in (1.32) need not be bounded or identical. Their intersection X also need not be bounded. This is true regardless of whether $s_i(\cdot)$ in (1.33) represents a gradient or a subgradient.

- The step-size α may remain fixed throughout the execution of algorithm (1.33), regardless of whether $s_i(\cdot)$ represents a gradient or a subgradient, even when the constraint sets X_i are not bounded.

- In the literature on decentralized optimization, it is typically assumed that agents' individual cost functions $J_i(\cdot)$ are convex, whether or not they are differentiable. For the case in which each agent's search direction $s_i(\cdot)$ is locally Lipschitz and based on a gradient of the agent's private cost function, our convergence conditions indicate that the convexity of all agents' individual costs is not necessary (q.v. Lemma 3.8.1).

- In contrast with existing Lyapunov-based analyses of related decentralized optimization schemes, the application of our SPAS theorem does not require a precise characterization of the algorithm's actual set of fixed points, and allows for the study of iterative methods directly in discrete-time.

Aside from preparing us to address the DT-DDCCP, the basic analysis philosophy presented in this chapter also turns out to be useful in the study of continuous-time variants of (1.33).

Chapter 4: Continuous-Time Consensus Optimization with Positivity Constraints. The setting of this chapter is most closely related to the formulation of the CT-SDCCP (q.v. Problem 1.2.4), though just as in Chapter 3, coordination control is not our main focus here. We consider static optimization problems of the form (1.31), in which X is a positivity constraint, and we are concerned with deriving the convergence properties of a continuous-time variant of algorithm 1.33. In contrast with many of the analytic techniques one finds in the literature on discrete-time consensus optimization, the techniques developed in Chapter 3 easily accommodate the continuous-time setting. In our main theorem (q.v. Theorem 4.4.1) we derive explicit bounds on the algorithm's convergence rate and ultimate estimation error, in terms of relevant problem parameters. In the last section of the chapter, we consider an application of the proposed scheme to a class of cooperative resource allocation problems, which are a special subclass of CT-SDCCPs represented by Problem 1.2.4.

As in Chapter 3, our analytic approach leads to several potentially interesting observations.

- The direct continuous-time counterpart of the standard consensus optimization algorithm (1.33) has a velocity vector field which is comprised of a linear graph Laplacian term and a gradient term which is weighted by a tunable step-size (i.e., optimization gain) parameter. We consider a variation of consensus optimization in which we introduce a consensus gain as a second tunable parameter. Several interesting observations can be made concerning the utility of this parameter, which can adjust the strength of the consensus term relative to that of the optimization term. We show that this tuning parameter can be used to improve the ultimate accuracy of the algorithm without affecting the convergence rate of the mean of agents' estimates. More importantly, this parameter can be used to relax the upper bound imposed on the step-size in order to guarantee the stability of the collective optimum (q.v. Theorem 4.4.1). These observations are made possible by the interconnected-systems viewpoint, and appear to have not been made elsewhere in the literature.

- The analysis is made challenging by the presence of a logical projection operation used to enforce a positivity constraint on the evolution of the dynamics. Remarkably, as in the discrete-time case, a careful examination of the interconnection terms in the composite Lyapunov argument leads to an elegant result concerning the effect of the logical projection operation on the evolution of the continuous-time consensus optimization dynamics (q.v. Lemma 4.4.3, part (d)).

Chapter 5: Decentralized Extremum-Seeking Control Based on Consensus Optimization. With the development of the SPAS theorem given in §2.5 and the analytic techniques described in Chapter 3, we are prepared to address the discrete-time, dynamic DCCP specified in the statement of Problem 1.2.2. Our main objective in this chapter is to derive a set of conditions under which the set $X^* \times \Xi^*$ in Problem 1.2.2 is semiglobally, practically, asymptotically stable (SPAS) for the feedback interconnection of the collective dynamics Σ given by (1.8), and the collective decision processes $\mathscr{D}$ given by (1.55). We consider the case in which $J_i(\cdot)$ is a function of the collective state x only, and the approximate search direction $\hat{s}_i(\cdot)$ is such that

$$\lim_{T\to\infty} \hat{s}_i(y_i(t_{k+1}^-)) = \nabla_u J_i(l(u))|_{u=u(t_k)}, \tag{1.59}$$

where $l(u)$ is the equilibrium map associated to the collective dynamics Σ (q.v. §1.2.1).

The main motivation for appealing to the stability theory for interconnected systems in studying consensus optimization in Chapter 3, is that the resulting analytic framework easily accommodates the presence of additional dynamics. In particular, the collective agent dynamics Σ can be simply regarded as yet another subsystem that dynamically couples to the mean and deviation subsystem dynamics studied in Chapter 3. In effect, the results of

Chapter 3 establish the convergence properties of the isolated subsystem dynamics $\mathscr{D}$, while the results of Chapter 5 establish the stability properties of the feedback interconnection of $\mathscr{D}$ and Σ.

Our results are summarized as follows.

- In Theorem 5.5.1, we derive a set of small-gain conditions on the parameters α and T, which are tunable in the scheme involving Σ in (1.8) and $\mathscr{D}$ in (1.55).
- In Theorem 5.5.3 we show that these conditions suffice to guarantee that the conditions of the SPAS Theorem 2.5.1 are satisfied.
- Lemmas 5.4.1 to 5.4.5 are preliminary to Theorem 5.5.1, and they embody our interconnected systems perspective.

Together with the interpretation of the optimization variables discussed in Example 1.4.1, these results demonstrate that consensus optimization methods constitute a viable solution to Problem 1.2.2, when the collective objective is given by (1.2).

Under a minor extension, the results presented in this chapter can be viewed as a proposal for a novel decentralized extremum-seeking scheme. On the other hand, relative to the literature on distributed optimization, our contribution here is to generalize the problem setting from that involving optimization of static maps, to that involving optimization of dynamic maps.

Chapter 6: Reduced Consensus Optimization. In this chapter we turn our attention to general, discrete-time, static DCCPs represented by Problem 1.2.3, and we consider the consensus optimization-based solution proposed in Example 1.4.1.

An obvious drawback of the decision updating process (1.40) is that multiagent systems employing it do not embody the desirable feature F4 described in §1.1. In particular, though agent i need not be aware of the collective objective or the actions taken by agents other than those in its graphical neighborhood, it must know how many other agents are participating in the optimization process since it must maintain an estimate of the entire optimal collective decision u^*. In large networks, it may happen that the actions of some agent j have a negligible effect on the performance of some other agent i. One may wonder why in such a case agent i ought to have any "opinion" as to how agent j ought to behave, and why he ought to expend any effort updating the variable $\xi_{i,j}$.

In many engineered mulitagent system applications, it is particularly cumbersome to require each node (i.e. agnet) to maintain an estimate of the optimal configuration u^* of the entire agent network; each time the network undergoes a structural modification such as the addition of a node, each preexisting node must modify its update rule to include additional variables representing an estimate of the added node's optimal action. Motivated by these considerations, in Chapter 6 we develop a streamlined version of algorithm 1.33 that we call *reduced consensus optimization* (RCO). Its primary strengths are the following:

- In general, agents implementing RCO need not be aware of the number of other agents participating in the optimization process.
- In large networks characterized by a sparse *interference structure* (i.e. a graph that indicates which agents' actions interfere with which agent's private costs), the total number of real-valued variables that need to be updated and exchanged by all agents at each iteration could be drastically smaller for RCO in comparison to standard consensus optimization (1.33).

The concept of RCO is particularly relevant to either static or dynamic DCCPs in which the interference structure may be known, or subject to design. In Chapter 7, we examine an application involving a coordination control problem in which the agents' private costs are subject to design.

Chapter 7: Decentralized Content Caching Strategies for Content-Centric Networks. In this chapter we study the problem of decentralized content caching in content-centric networks, which belongs to the class of discrete-time, static DCCPs represented by Problem 1.2.3. We propose a general methodology for the design of decentralized content-caching strategies (CCSs) on the basis of the reduced consensus optimization (RCO) algorithm proposed in Chapter 6. Our contributions in this chapter can be summarized as follows.

- The content caching problem is typically formulated as an integer program. We propose a formulation of the decentralized content caching problem as a constrained convex program of the form (1.31) (q.v. §7.3).
- We consider a set of performance criteria related to the efficient use of the network's transport and caching resources, and we propose a general, systematic method for the design of agents' private cost functions in reflection of these performance criteria (q.v. §7.4.1 and §7.4.2).
- We provide a basic, though extensive pedagogical example demonstrating the application of our proposed design methodology (q.v. §7.5).
- We evaluate the performance of a specific CCS derived using the proposed methodology, on an 11-node network (the European optical backbone network, COST 239). We compare the performance of our RCO-based CCS against the least frequently used (LFU) cache eviction policy, along several network-wide efficiency-related performance metrics (q.v. §7.6).

By comparison to several content caching strategies proposed in the literature, RCO-based CCSs have the following strengths:

- Many CCSs proposed in the literature are designed to optimize the performance of individual caches or network substructures such as trees or paths from sources to consumers. The RCO-based CCS is designed to manage the network as a whole.
- Many CCS designs proposed in the literature depend on extensive network models or simulations which are typically carried out offline. Sometimes these strategies are tuned for a best response to simplified traffic pattern predictions, which may not reflect actual traffic patterns once the CCS is deployed. The RCO-based CCS is designed to respond in real-time to network-wide changes in content demand patterns. In particular, the set of exogenous signals $d_i(t)$ representing the set of environmental conditions affecting agent i in the statement of Problem 1.2.3 are in this setting interpreted as time-varying content demand rates measured at node i.
- Some CCS design approaches guarantee a quantifiably suboptimal caching performance only in the case that certain symmetry assumptions are satisfied (such as all caches having the same size, and all content demand rates being equal at each node). Others require nodes to know or estimate the operational parameters (such as cache size) of other nodes in the network, and the efficacy with which the caching resources are utilized is known to be affected by the accuracy of such estimates. The RCO-based CCS explicitly accommodates node heterogeneity with respect to all relevant characteristics, including cache size, transport efficiency parameters, caching efficiency parameters, and content demand rates.

- Many CCSs proposed in the literature are purely heuristic. RCO-based CCSs are based on provably-convergent consensus optimization methods.

The content-caching problem is an excellent example of a static DCCP in which an RCO variant of the solution (1.40) demonstrates the main proposition of this thesis.

Chapter 2

Background and Preliminaries

2.1 Graphs and some of their Spectral Properties

A *graph* $\mathscr{G}$ is a pair $(\mathscr{V},\mathscr{E})$, where $\mathscr{V}$ is a set of *vertices* (or *nodes*), and $\mathscr{E} \subset \mathscr{V} \times \mathscr{V}$ is the set of *directed edges* between them. A graph $\mathscr{G}$ is *undirected* if $(i,j) \in \mathscr{E} \implies (j,i) \in \mathscr{E}$, and it is *directed* otherwise. We refer to a directed graph as a *digraph*, while using the term "graph" to refer to an undirected graph. The *adjacency matrix* $A \in \mathbb{R}^{|\mathscr{V}|\times|\mathscr{V}|}$ associated with $\mathscr{G}$ is given by[1]

$$[A]_{i,j} = \begin{cases} 1, & (j,i) \in \mathscr{E} \\ 0, & \text{otherwise,} \end{cases} \tag{2.1}$$

where $[A]_{i,j}$ is the element of A in the ith row and jth column (q.v. Appendix A). Thus, $\mathscr{G}$ is undirected if $A = A^T$. A *weighted adjacency matrix* $A \in \mathbb{R}^{|\mathscr{V}|\times|\mathscr{V}|}$ associated with $\mathscr{G}$ is given by

$$[A]_{i,j} = \begin{cases} a_{i,j}, & (j,i) \in \mathscr{E} \\ 0, & \text{otherwise,} \end{cases} \tag{2.2}$$

where $a_{i,j}$ is some positive, real number. Two vertices on a graph are *adjacent* if there exists an edge between them. A graph is *complete* if every pair of vertices in $\mathscr{V}$ is adjacent. The set $\mathscr{N}(i) \subset \mathscr{V}$ is the set of node i's *one-hop neighbors* (or just *neighbors*) on $\mathscr{G}$, and is equal to the set of all nodes adjacent to node i on the graph $\mathscr{G}$. A *subgraph* of a graph $\mathscr{G}$ is a graph $\mathscr{G}_o = (\mathscr{V}_o,\mathscr{E}_o)$, with $\mathscr{V}_o \subset \mathscr{V}$, and $\mathscr{E}_o \subset (\mathscr{V}_o \times \mathscr{V}_o) \cap \mathscr{E}$. The *degree* of a node i is equal to the number of its neighbors, and is denoted by d_i (i.e., $d_i = |\mathscr{N}(i)|$). The *degree matrix* of a graph $\mathscr{G}$ is a diagonal matrix $D \in \mathbb{R}^{|\mathscr{V}|\times|\mathscr{V}|}$, with $[D]_{i,i} = d_i$. The *graph Laplacian* associated to $\mathscr{G}$ is the matrix $L = D - A$. Equivalently, L is given by

$$[L]_{i,j} = \begin{cases} d_i, & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } (j,i) \in \mathscr{E} \\ 0, & \text{otherwise.} \end{cases} \tag{2.3}$$

From this definition of L, we observe that $L\mathbf{1} = 0$ – i.e., $0 \in \sigma(L)$, with eigenvector $\mathbf{1}$ (q.v. Appendix A for a list of notation). For undirected graphs, $L = L^T$, and therefore $\sigma(L) \subset \mathbb{R}$. From the definition (2.3) we can then also

[1] Some authors use the opposite convention: $[A]_{i,j} = 1$ if $(i,j) \in \mathscr{E}$, and $[A]_{i,j} = 0$ otherwise.

conclude that $\sigma(L) \subset \mathbb{R}_+$. One way to see this is to apply the Geršgorin disc theorem (q.v. Theorem 6.1.1 in [67], for example), which states that for any matrix $L \in \mathbb{R}^{n\times n}$, $\sigma(L) \subset \cup_{i=1}^{n} G_i(L)$, where $G_i(L)$ is a *Geršgorin disc*, defined as

$$G_i(L) = \{z \in \mathbb{C} \mid |z - [L]_{i,i}| \leq R_i(L)\}, \tag{2.4}$$

where

$$R_i(L) = \sum_{j\in\{1,\ldots,n\}\setminus\{i\}} |[L]_{i,j}|. \tag{2.5}$$

Clearly then, for the graph Laplacian (2.3), $\cup_{i=1}^{n} G_i(L)$ corresponds to a union of n intervals, the ith interval being centred at d_i and having radius d_i – i.e., $\sigma(L) \subset [0, 2\max_i d_i]$. We denote the spectrum of L as follows: $\sigma(L) = \{\lambda_1(L), \ldots, \lambda_n(L)\}$, with

$$0 = \lambda_1(L) \leq \lambda_2(L) \leq \cdots \leq \lambda_n(L). \tag{2.6}$$

A *path* from a vertex i to a vertex j in $\mathscr{V}$ is a sequence of edges leading from i to j. A path may alternatively be defined as a sequence of vertices such that the first vertex in the sequence is i, the last vertex in the sequence is j, and there exists an edge between any two consecutive vertices within the sequence. A graph is *connected* if for any pair of its vertices i and j, there exists a path from i to j. A *connected component* of a graph $\mathscr{G}$ is a subgraph $\mathscr{G}_o = (\mathscr{V}_o, \mathscr{E}_o)$ of $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ which is connected and is such that there do not exist a pair of nodes i and j, with $i \in \mathscr{V}\setminus\mathscr{V}_o$ and $j \in \mathscr{V}_o$, having a path between them on $\mathscr{G}$. Clearly, a graph is connected if it is comprised of only one connected component – i.e., the graph itself.

The multiplicity of the zero eigenvalue of L is equal to the number of connected components of $\mathscr{G}$ (q.v. Proposition 2.3 in [107], for example). We therefore conclude that if a graph is connected, the eigenvalue $\lambda_1(L) = 0$ is isolated, and therefore $\lambda_2(L) > \lambda_1(L)$. The second smallest eigenvalue $\lambda_2(L)$ plays a special role in continuous-time consensus algorithms, and is often referred to as the *algebraic connectivity* of $\mathscr{G}$. In the literature, $\lambda_2(L)$ is also sometimes called the *Fiedler eigenvalue* of L, in honour of M. Fiedler, who first studied its properties [49].

2.2 Nonnegative Matrices

For a matrix $A \in \mathbb{R}^{N\times N}$, $\sigma(A) \subset \mathbb{C}$ denotes its spectrum. Its *spectral radius* is given by

$$\rho(A) = \max_{\lambda\in\sigma(A)} |\lambda|. \tag{2.7}$$

A matrix $A \in \mathbb{R}^{N\times N}$ is *nonnegative* if each of its entries is nonnegative, in which case we write $A \geq 0$. From its definition in the prequel, we see that a weighted adjacency matrix is nonnegative. By Theorem 1.1 in Chapter 2 of [13], every nonnegative matrix has an eigenvalue equal to its spectral radius – i.e., $A \geq 0 \Longrightarrow \rho(A) \in \sigma(A)$. A matrix $A \in \mathbb{R}^{N\times N}$, with $N \geq 2$ is *reducible* if there exists a permutation matrix P such that

$$P^T A P = \begin{bmatrix} B & C \\ \mathbf{0} & D \end{bmatrix}, \tag{2.8}$$

where B, C, D and $\mathbf{0}$ are any dimensionally compatible matrices (q.v. Definition 6.2.1 in [67]). A matrix $A \in \mathbb{R}^{N\times N}$ is *irreducible* if it is not reducible. A weighted adjacency matrix $A \in \mathbb{R}^{N\times N}$ associated to a graph $\mathscr{G}$ is irreducible if, and only if $\mathscr{G}$ is connected (q.v. Theorem 2.7 in Chapter 2 of [13], for example). According to Perron-Frobenius theory, if a nonnegative matrix A is irreducible, then $\rho(A)$ is a simple eigenvalue of A, and any other eigenvalue of the same modulus is also simple (q.v. Theorem 1.4 (b) in Chapter 2 of [13], for example). A nonnegative matrix

A is *primitive* if it is irreducible and for all $\lambda \in \sigma(A)\backslash\{\rho(A)\}$, $\rho(A) > |\lambda|$ (i.e., $\rho(A)$ is the only eigenvalue of maximum modulus). An irreducible matrix is primitive if its trace is positive (q.v. Corollary 2.28 in Chapter 2 of [13]). Therefore, if A is a weighted adjacency matrix associated with a connected graph $\mathscr{G}$, A is primitive if $\mathscr{G}$ contains at least one positively-weighted self-loop. A matrix $A \in \mathbb{R}^{N\times N}$ is *stochastic* if $A \geq 0$ and each row sums to one – i.e., for each $i \in \{1,\ldots,N\}$, $\sum_{j=1}^{N}[A]_{i,j} = 1$. A matrix A is *doubly stochastic* if it is stochastic and its transpose is stochastic. Let $\mathscr{V}$ denote the set $\{1,\ldots,N\}$. According to Theorem 2.35 in Chapter 2 of [13], for any nonnegative, irreducible matrix $A \in \mathbb{R}^{N\times N}$, it holds that

$$\min_{i\in\mathscr{V}} \sum_{j\in\mathscr{V}} [A]_{i,j} \leq \rho(A) \leq \max_{i\in\mathscr{V}} \sum_{j\in\mathscr{V}} [A]_{i,j}. \tag{2.9}$$

We therefore conclude that for a stochastic, weighted adjacency matrix $A \in \mathbb{R}^{N\times N}$ associated to a connected graph $\mathscr{G}$ with at least one positively-weighted self-loop, $\rho(A) = 1$ is a simple eigenvalue of A, and every other eigenvalue of A has a smaller modulus. In other words, the spectrum of such a matrix A can be arranged as

$$1 = \rho(A) = \lambda_1(A) > |\lambda_2(A)| \geq \cdots \geq |\lambda_N(A)| \geq 0. \tag{2.10}$$

2.3 Consensus Algorithms

Consensus algorithms are a fundamental building block of the cooperative multiagent decision-updating processes studied in this thesis. There is an extensive literature on the subject within control theory, and reference [136] provides a good overview of its applications and origins. Here we briefly review two basic variants that play a role in this thesis.

2.3.1 Continuous-Time Consensus Algorithms

Consider a system of N agents in which the state of the ith agent evolves according to

$$\dot{x}_i = -[L]_i x, \tag{2.11}$$

where $x = [x_1,\ldots,x_N]^T \in \mathbb{R}^N$, and L is the Laplacian matrix of a graph $\mathscr{G}$ whose nodes represent the agents, and whose edges specify those pairs of agents that can exchange state information at any time. Written compactly, (2.11) becomes

$$\dot{x} = -Lx, \qquad x(0) = x_0 \in \mathbb{R}^N. \tag{2.12}$$

We have the following.

Lemma 2.3.1: If $\mathscr{G}$ is undirected and connected, the point $\bar{x} = \frac{1}{N}\mathbf{1}\mathbf{1}^T x_0$ is asymptotically stable for (2.12).

Proof. Introduce two auxiliary variables: define $y = \frac{1}{N}\mathbf{1}^T x$ and $z = Mx$, with

$$M = I_N - \tfrac{1}{N}\mathbf{1}\mathbf{1}^T. \tag{2.13}$$

The variable y clearly represents the mean of agent states at any time, and z quantifies the agent states' deviations from that mean – i.e.,

$$z = \begin{bmatrix} x_1 - y \\ \vdots \\ x_N - y \end{bmatrix}. \tag{2.14}$$

From the definition of L in (2.3), we see that $L\mathbf{1} = 0$, and we therefore find that $\dot{y} = \frac{1}{N}\mathbf{1}^T\dot{x} = \frac{1}{N}\mathbf{1}^T Lx = 0$. In other words, the mean of agent states remains constant forever, and is given by $y(t) = y(0) = \frac{1}{N}\mathbf{1}^T x(0)$. Since the mean $y(t)$ of agent states remains fixed throughout the evolution of (2.12), showing that $\bar{x} = \mathbf{1}y(0)$ is asymptotically stable for (2.12) amounts to showing that the vector of deviations (2.14) converges to the origin.

To examine the evolution of the deviation variable, first note that $ML = L = LM$. Therefore,

$$\dot{z} = M\dot{x} = -MLx = -LMx = -Lz, \qquad z(0) = Mx_0. \tag{2.15}$$

The difference between the dynamics (2.15) of the deviation variable and the agent dynamics (2.12), is that z is constrained to evolve on the orthogonal complement of the subspace spanned by the vector $\mathbf{1}$. To see this, note that for any $x_0 \in \mathbb{R}^N$ and for any $t \in \mathbb{R}_+$, $\mathbf{1}^T z(t) = \mathbf{1}^T Mx(t) = 0$, owing to the definition of M in (2.13). In fact, M is the matrix realization of the orthogonal projection onto the set $\mathrm{span}\{\mathbf{1}\}^\perp$.

The asymptotic stability of $z = 0$ for (2.15) can be concluded by considering the Lyapunov function $V_{\mathscr{C}}(z) = \frac{1}{2}\|z\|^2$, whose time-derivative along the trajectories of (2.15) is given by $\dot{V}_{\mathscr{C}}(z) = -z^T Lz$. We aim to show that the term $-z^T Lz$ is negative definite on $\mathrm{span}\{\mathbf{1}\}^\perp$.

By the spectral mapping theorem, $\sigma(-L) = -\sigma(L)$, and therefore $\sigma(L) \subset [0, 2\max_i d_i]$ implies that $\sigma(-L) \subset [-2\max_i d_i, 0]$. Moreover, the smallest eigenvalue of L is $\lambda_1(L) = 0$, corresponding to the eigenvector $\mathbf{1}$. Since $\mathscr{G}$ is connected, $\lambda_1(L)$ is isolated, and therefore $\lambda_2(-L) = -\lambda_2(L) < 0$. Then, by Theorem 3 in [136], we have that for all $z \in \mathrm{span}\{\mathbf{1}\}^\perp$ – i.e., the orthogonal complement of the subspace spanned by the eigenvector corresponding to the largest eigenvalue of $(-L)$ – the following bound holds

$$\dot{V}_{\mathscr{C}}(z) = z^T(-L)z \leq -\lambda_2(L)\|z\|^2. \tag{2.16}$$

Standard Lyapunov arguments (q.v. Theorem 4.1 in [80], for example) then allow us to conclude that $z = 0$ is asymptotically stable for (2.15). Equivalently, we have that for all $i \in \mathscr{V}$, $x_i \to y \equiv \frac{1}{N}\mathbf{1}^T x_0$, asymptotically. $\diamondsuit$

We refer to system (2.12) as the *continuous-time consensus dynamics*, because the agents executing it reach a consensus equilibrium state whenever $\mathscr{G}$ is connected. Our "deviation variable" z is sometimes referred to as the "disagreement vector" [136], and the subspace $\mathrm{span}\{\mathbf{1}\}$ is often called the "agreement subspace".

2.3.2 Discrete-Time Consensus Algorithms

Discrete-time consensus algorithms take a form similar to that of (2.11), except that the graph Laplacian is replaced by a specially weighted adjacency matrix. Once again, consider a system of N agents in which the state of the ith agent evolves according to

$$x_i(t+1) = [A]_i x(t), \qquad t \in \mathbb{N} \tag{2.17}$$

where $x = [x_1, \ldots, x_N]^T \in \mathbb{R}^N$ and A is a weighted adjacency matrix associated to $\mathscr{G}$, whose nodes represent the agents, and whose edges specify those pairs of agents that can exchange state information at any time. Written compactly, (2.17) becomes

$$x^+ = Ax, \qquad x(0) = x_0 \in \mathbb{R}^N. \tag{2.18}$$

We have the following.

Lemma 2.3.2: If A is stochastic and $\mathscr{G}$ is connected, undirected, and has at least one positively-weighted self-loop, then $x \to \bar{x} = \frac{1}{N}\mathbf{1}\mathbf{1}^T x_0$ at a geometric rate.

Proof. We define the mean and deviation variables exactly as in the proof of Lemma 2.3.1, and we examine their evolution. Since $\mathscr{G}$ is undirected, $A = A^T$, and the stochasticity of A therefore implies that $\mathbf{1}^T A = \mathbf{1}^T$. Consequently, the mean of the agents' states evolves according to $y^+ = \frac{1}{N}\mathbf{1}^T x^+ = y$; in other words, the mean state remains constant forever, and is given by $y(t) = y(0) = \frac{1}{N}\mathbf{1}^T x(0)$. Since the mean state remains fixed throughout the evolution of (2.18), showing that $\bar{x} = \mathbf{1}y(0)$ is asymptotically stable for (2.18) amounts to showing that the vector of deviations (2.14) converges to the origin.

To examine the evolution of the deviation variable, first note that $MA = A - \frac{1}{N}\mathbf{1}\mathbf{1}^T = AM$. Therefore,

$$z^+ = MAx^+ = MAx = AMx = Az, \qquad z(0) = Mx_0. \tag{2.19}$$

As in the continuous-time case, the deviation variable $z = Mx$ is constrained to evolve on the orthogonal complement of the subspace spanned by the vector $\mathbf{1}$, which follows from the definition of M in (2.13).

The geometric convergence of z to the origin can be concluded by considering the evolution of the norm $\|z\|$. From (2.19), we have that $\|z^+\|^2 = z^T A^2 z$. By the spectral mapping theorem, $\lambda \in \sigma(A) \Longrightarrow \lambda^2 \in \sigma(A^2)$. Therefore, our assumptions and the discussion leading to (2.10) implies that the spectrum of A^2 can be arranged as follows:

$$1 = \lambda_1(A^2) = \lambda_1(A)^2 > \lambda_2(A^2) \geq \cdots \geq \lambda_N(A^2) \geq 0. \tag{2.20}$$

In particular, (2.10) implies that $\lambda_2(A^2) = \max\{\lambda_2(A)^2, \lambda_N(A)^2\} < 1$. As in the proof of Lemma 2.3.1, we apply Theorem 3 from [136] to conclude that for all $z \in \operatorname{span}\{\mathbf{1}\}^\perp$, $z^T A^2 z \leq \lambda_2(A^2)\|z\|^2$, and therefore that

$$\|z^+\|^2 \leq \lambda_2(A^2)\|z\|^2. \tag{2.21}$$

The contraction mapping principle (q.v. §2.3.1 in [124], for example), together with the fact that $\lambda_2(A^2) < 1$ allows us to conclude that $(\|z(t)\|)_{t=0}^\infty \to 0$ at a geometric rate. Consequently, $(z(t))_{t=0}^\infty \to \mathbf{0}$, and therefore $(x(t))_{t=0}^\infty \to \bar{x}$, both at a geometric rate. ♢

Remark 2.3.1: If the graph topology is fixed, as is the case in many cluster computing or parallel processing applications, then the network nodes may implement the following decentralized algorithm (also described in [114] and [167]) in order to generate a set of link weights that yield the desired spectral properties of A:

Algorithm 2.3.1:

1. $\forall i \in \mathscr{V}$, agent i sets $p_i = \frac{1}{|\mathscr{N}_i|+1}$, where $\mathscr{N}_i$ denotes the number of i's one-hop neighbors on $\mathscr{G}_C$.

2. $\forall i \in \mathscr{V}$, agent i sends to each agent $j \in \mathscr{N}_i$ the number p_i, and receives the number p_j in return.

3. $\forall (i, j) \in \mathscr{E}_C$, agents i and j select the weight for edge (i, j) as $[A]_{i,j} = [A]_{j,i} = \min\{p_i, p_j\}$.

4. $\forall i \in \mathscr{V}$, agent i sets $[A]_{i,i} = 1 - \sum_{j \in \mathscr{V}\setminus\{i\}}[A]_{i,j}$.

♢

2.4 Convex Analysis

There are many excellent references on convex analysis, including [22], [15], [14] and [124]. We list some basic notions here.

A *convex combination* of a set of vectors $\{x_1,\dots,x_N\}$ in $\mathbb{R}^n$ is a vector $x = a_1x_1 + \cdots + a_Nx_N$, where for each $i \in \{1,\dots,N\}$, the coefficient a_i is nonnegative, and $\sum_{i=1}^N a_i = 1$. A set $C \subset \mathbb{R}^n$ is *convex* if any convex combination of its elements is also an element in C. If S is a subset of $\mathbb{R}_n$, then its *convex hull* is a set C formed by all possible convex combinations of the elements in S; in that case we write $C = \mathrm{co}(S)$. The intersection of an arbitrary collection of convex sets is convex (q.v. Proposition 1.2.1 [14]). A function $J : \mathbb{R}^n \to \mathbb{R}$ is *convex* if for any $a \in [0,1]$ and for any x_1 and x_2 in $\mathbb{R}^n$, it holds that

$$J(ax_1 + (1-a)x_2) \le aJ(x_1) + (1-a)J(x_2). \tag{2.22}$$

The same function is *strictly convex* if inequality (2.22) holds strictly whenever $x_1 \neq x_2$. The same function is *strongly convex* with constant $\beta \in \mathbb{R}_{++}$ if

$$J(ax_1 + (1-a)x_2) \le aJ(x_1) + (1-a)J(x_2) - \beta\tfrac{a(1-a)}{2}\|x_1 - x_2\| \tag{2.23}$$

(q.v. §1.1 in [124]). For a convex function $J : \mathbb{R}^n \to \mathbb{R}$, (2.22) generalizes to *Jensen's inequality*, which states that for any set of vectors $\{x_1,\dots,x_N\}$ in $\mathbb{R}^n$ and for any set of nonnegative coefficients $\{a_1,\dots,a_N\}$ that sum to one,

$$J(a_1x_1 + \cdots + a_Nx_N) \le a_1J(x_1) + \cdots + a_NJ(x_N). \tag{2.24}$$

A differentiable function $J : \mathbb{R}^n \to \mathbb{R}$ is convex if, and only if for any x_1 and x_2 in $\mathbb{R}^n$,

$$J(x_1) \ge J(x_2) + \nabla J(x_2)^T(x_1 - x_2). \tag{2.25}$$

The same function $J(\cdot)$ is strictly convex if, and only if the inequality (2.25) holds strictly whenever $x_1 \neq x_2$. The same function is strongly convex with constant $\beta \in \mathbb{R}_{++}$ if, and only if

$$J(x_1) \ge J(x_2) + \nabla J(x_2)^T(x_1 - x_2) + \tfrac{\beta}{2}\|x_1 - x_2\| \tag{2.26}$$

(q.v. Lemma 3 in §1.1 of [124]). A convex function is continuous (q.v. Proposition 1.4.6 [14], or Lemma 3 in §5.1 of [124]).

A point $x^* \in \mathbb{R}^n$ is a *local minimum* for $J : \mathbb{R}^n \to \mathbb{R}$ if there exists a neighbourhood of x^* in which $J(x) \ge J(x^*)$, for every x in that neighbourhood. The point x^* is a *global minimum* if $J(x) \ge J(x^*)$ for all $x \in \mathbb{R}^n$. If x^* is a minimum for $J(\cdot)$ and $J(\cdot)$ is differentiable, then the *first-order necessary condition for optimality* states that $\nabla J(x^*) = 0$. If $J(\cdot)$ is convex and differentiable, then $\nabla J(x^*) = 0$ if, and only if x^* is a global minimum for $J(\cdot)$.

All sublevel sets of a convex function are closed and convex (q.v. Corollary to Lemma 3 in §5.1 of [124]). If some sublevel set of a convex function is nonempty and bounded, then there exists a minimum for that function (q.v. Theorem 3 in §5.2 of [124]). If some sublevel set of a convex function is nonempty and bounded, then all sublevel sets of that function are bounded (q.v. Lemma 1 in §5.3 of [124]). For a strictly convex function there exists a unique minimum (q.v. Theorem 3 in §1.3 of [124]).

A vector $s_{x_o} \in \mathbb{R}^n$ is a *subgradient* of a convex function $J : \mathbb{R}^n \to \mathbb{R}$ at $x_o \in \mathbb{R}^n$ if for all $x \in \mathbb{R}^n$,

$$J(x) \geq J(x_o) + s_{x_o}^T (x - x_o). \tag{2.27}$$

Clearly, every gradient is a subgradient – i.e., if a convex function $J(\cdot)$ is differentiable at x, then $\partial J(x) = \{\nabla J(x)\}$. The *subdifferential* of $J(\cdot)$ at x_o is the set of all subgradients of $J(\cdot)$ at x_o, and it is denoted by $\partial J(x_o)$. The subdifferential of a convex function is nonempty, convex and compact at every point in the function's domain (q.v. Proposition 4.2.1 in [14]). For a nondifferentiable convex function $J(\cdot)$, the condition $0 \in \partial J(x_o)$ is necessary and sufficient for x_o to be a minimum of $J(\cdot)$ (q.v. Theorem 1 in §5.2 of [124]).

An *orthogonal projection* of a point $x \in \mathbb{R}^n$ onto a closed set C is defined as

$$\mathbf{P}_C(x) = \arg\min_{x_o \in C} \|x - x_o\|. \tag{2.28}$$

If C is convex, then $\mathbf{P}_C(x)$ is unique. The projection operator is *nonexpansive* – i.e., for any x_1 and x_2 in $\mathbb{R}^n$,

$$\|\mathbf{P}_C(x_1) - \mathbf{P}_C(x_2)\| \leq \|x_1 - x_2\| \tag{2.29}$$

(q.v. Proposition 2.2.1 in [14]). The following lemma is used in the proof of Lemma 3.6.3. An alternative proof can be found in Lemma 1 (a), [116].

Lemma 2.4.1: Consider a closed, convex set $S \subset \mathbb{R}^n$ and let $w \in \mathbb{R}^n$ be arbitrary. Then, for any $w^* \in S$,

$$(w - w^*)^T (\mathbf{P}_S(w) - w) \leq -\|w - \mathbf{P}_S(w)\|^2. \tag{2.30}$$

Proof. We have

$$\begin{aligned} \|\mathbf{P}_S(w) - w\|^2 &= (\mathbf{P}_S(w) - w)^T (\mathbf{P}_S(w) + w^* - w^* - w) \\ &= (\mathbf{P}_S(w) - w)^T (\mathbf{P}_S(w) - w^*) \\ &\quad + (\mathbf{P}_S(w) - w)^T (w^* - w). \end{aligned}$$

Inequality (2.30) follows by rearranging this expression and noting that the inner product $(\mathbf{P}_S(w) - w)^T (\mathbf{P}_S(w) - w^*)$ is negative semidefinite, owing to the geometry of convex sets. □

2.5 Semiglobal, Practical, Asymptotic Stability

All results presented in this section are new and have not been published elsewhere.

Let $\Xi \subset \mathbb{R}^n$ be a closed, convex set, and consider the system

$$\xi^+ = \mathbf{P}_\Xi\big(f(\xi;\alpha)\big), \quad \xi \in \mathbb{R}^n, \tag{2.31}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is parameterized by $\alpha \in \mathbb{R}$.

Definition 2.5.1: For a given $\alpha \in \mathbb{R}_{++}$, a compact set $S \subset \Xi$ is said to be *stable* for (2.31) on Ξ if for every $\varepsilon \in \mathbb{R}_{++}$ there exists a $\delta \in \mathbb{R}_{++}$ such that for all $t \in \mathbb{N}$, $\xi(t) \in \bar{B}_\varepsilon(S) \cap \Xi$ whenever $\xi(0) \in \bar{B}_\delta(S) \cap \Xi$.

Definition 2.5.2: Let σ be a number in $\mathbb{R}_{++} \cup \{\infty\}$. For a given $\alpha \in \mathbb{R}_{++}$, a compact set $S \subset \Xi$ is said to be *attractive* for (2.31) on $\bar{B}_\sigma(S) \cap \Xi$ if for every $\varepsilon \in (0, \sigma)$ there exists a $T \in \mathbb{N}$ such that for all $t \geq T$, $\xi(t) \in$

$\bar{B}_\varepsilon(S) \cap \Xi$, whenever $\xi(0) \in \bar{B}_\sigma(S) \cap \Xi$.

Definition 2.5.3: A set $\Gamma_0 \subset \Xi$ is *semiglobally practically asymptotically stable* (SPAS) for (2.31) if for any positive, real numbers σ and ρ, with $\infty > \sigma > \rho$, there exists a number $\alpha^* \in \mathbb{R}_{++}$ such that whenever $\alpha \in (0, \alpha^*)$, $\bar{B}_\rho(\Gamma_0) \cap \Xi$ is stable and attractive for (2.31) on $\bar{B}_\sigma(\Gamma_0) \cap \Xi$.

We also define the related convergence concept of asymptotic stability, which is stronger than SPAS.

Definition 2.5.4: Let σ be a number in $\mathbb{R}_{++} \cup \{\infty\}$. A set $\Gamma_0 \subset \Xi$ is *asymptotically stable* for (2.31) on $\bar{B}_\sigma(\Gamma_0) \cap \Xi$ if Γ_0 is stable and attractive for (2.31) on $\bar{B}_\sigma(\Gamma_0) \cap \Xi$. $\diamondsuit$

Theorem 2.5.1: Consider the system (2.31), and suppose there exists a compact set $\Gamma_0 \subset \Xi$ and a function $V \in C^0[\mathbb{R}^n, \mathbb{R}_+]$ which is radially unbounded and positive definite with respect to Γ_0 on $\mathbb{R}^n$. Suppose that for every $\sigma \in \mathbb{R}_{++}$, there exists a number $\bar{\alpha} \in \mathbb{R}_{++}$ such that whenever $\alpha \in (0, \bar{\alpha})$,

$$V(\xi^+) - V(\xi) \leq W(\xi; \alpha), \quad \forall \xi \in \bar{B}_\sigma(\Gamma_0) \cap \Xi, \tag{2.32}$$

where the function $W \in C^0[\mathbb{R}^n, \mathbb{R}]$, parameterized by α, has the following properties:

should make this only R_+

- P1: There exists $b_W(\alpha) \in \mathbb{R}_{++}$ such that

  $$W(\xi; \alpha) \leq b_W(\alpha), \quad \forall \xi \in \bar{B}_\sigma(\Gamma_0) \cap \Xi,$$

  and $\lim_{\alpha \downarrow 0} b_W(\alpha) = 0$.

- P2: The set

  $$Z(\alpha) = \{\xi \in \Xi \mid W(\xi; \alpha) \geq 0\} \tag{2.33}$$

  contains Γ_0.

- P3: For every $\delta \in (0, \sigma)$, there exists an $\alpha_Z \in \mathbb{R}_{++}$ such that whenever $\alpha \in (0, \min\{\alpha_Z, \bar{\alpha}\})$, $Z(\alpha) \subseteq \bar{B}_\delta(\Gamma_0)$.

Then, Γ_0 is semiglobally practically asymptotically stable for (2.31), with Lyapunov function $V(\cdot)$.

Should give an intiutitive explanation of properties. And change order: P2: W is negative except on a compact set containing Gamma_0, and P1:, it is upper bounded by a quantity that can be made arbitrarily small by adjusting the tuning parameter. Moreover, P3: the set where W is non-negative can also be made arbitrarily small by appropriately adjusting the tuning parameter. .i.e., V decreases for all xi outside some set Z containing Gamma_0, and that set can be made to fit inside an arbitrarily small ball centered at Gamma_0.

Proof. The proof is given in §2.5.2 □

Remark 2.5.1: The proof of Theorem 2.5.1 is an adaptation of the proofs of Theorem 5.14.2 and Corollary 5.14.3 in [3] to the preceding definition of semiglobal practical asymptotic stability of compact sets. In particular, Definition 2.5.3 is stronger than the definition of practical stability employed in [3] since the latter does not entail any notion of parametrization of the size of X^*'s basin of attraction, or the ultimate upper bound on the sequences' distance from X^*. In contrast, such parameterizations are customary in definitions of SPAS found in the control theoretic literature [149]. The proof of our theorem may be of independent interest as it represents an extension of the results in [149] to systems that evolve in discrete time and on subsets of their state space, and have a compact set of equilibria. Moreover, the development of the analytic tools introduced in [149] is motivated primarily by applications involving periodically perturbed systems, whereas our setting involves a more general class of non-vanishing perturbations. $\diamondsuit$

Prior to presenting the proof of Theorem 2.5.1, we provide two lemmas used therein.

2.5.1 Technical Lemmata

The following are two technical lemmas used in the proof of the SPAS theorem, and referenced on several occasions throughout Chapter 3.

Lemma 2.5.1: Consider a function $\psi \in C^0[\mathbb{R}^n, \mathbb{R}_+]$ which is radially unbounded and positive definite with respect to a compact set $\Psi_0 \subset \mathbb{R}^n$ on $\mathbb{R}^n$. For any given $\hat{\rho} \in \mathbb{R}_{++}$, there exists a number $l \in \mathbb{R}_{++}$ such that the set $\Psi_l = \{\xi \in \mathbb{R}^n \mid \psi(\xi) \leq l\}$ is strictly contained inside the set $\bar{B}_{\hat{\rho}}(\Psi_0)$.

should be Bar.

Proof. Let $\bar{\bar{l}}$ be the minimum value attained by $\psi(\cdot)$ on the set $\partial B_{\hat{\rho}}(\Psi_0)$. The existence of $\bar{\bar{l}}$ is guaranteed by the continuity of $\psi(\cdot)$ and the compactness of $\partial B_{\hat{\rho}}(\Psi_0)$. For the same reasons, the number

$$\bar{l} = \min_{\xi \in \Psi_{\bar{\bar{l}}} \backslash B_{\hat{\rho}}(\Psi_0)} \psi(\xi)$$

Q: where does radial unboundedness enter in this proof?
compactness of sublevel sets ensures that the second min actually exists.
But need to fix sentence: "For the same reasons..." It's not for the same reasons. Psi_l_barBar is compact exactly because radial unboundedness is assumed. That needs to be made explicit.

is also guaranteed to exist[2].

Since $\psi(\cdot)$ is positive definite with respect to Ψ_0 on $\mathbb{R}^n$, $\bar{l}$ is positive and there exists a number $l \in (0, \bar{l})$. It can be seen that for any such l, Ψ_l is strictly contained inside $\bar{B}_{\hat{\rho}}(\Psi_0)$, for, assuming that there exists a point $\xi_o \in \Psi_{\bar{\bar{l}}} \backslash B_{\hat{\rho}}(\Psi_0)$ leads to the absurd conclusion that both $\psi(\xi_o) \leq l$ and $\psi(\xi_o) \geq \bar{l}$ hold. □

It is by means of this lemma that we remove the strong convexity assumption used in [91].

Lemma 2.5.2: Consider a function $\phi \in C^0[\mathbb{R}^n, \mathbb{R}_+]$ which is positive definite with respect to a compact set $X^* \subset \mathbb{R}^n$. Given any three positive, real numbers K_ϕ, σ and $\hat{\delta}$, with $\infty > \sigma > \hat{\delta}$, there exists a number $\alpha_\phi \in \mathbb{R}_{++}$ such that whenever $\alpha \in (0, \alpha_\phi)$,

$$\phi(y) \geq \alpha K_\phi \|y - \mathbf{P}_{X^*}(y)\|^2, \quad \forall y \in \bar{B}_\sigma(X^*) \backslash B_{\hat{\delta}}(X^*). \tag{2.34}$$

Proof. By Lemma 2.5.1 (with $\hat{\rho} = \hat{\delta}$), there exists a number $c \in \mathbb{R}_{++}$ such that the set

$$\Phi_c = \{y \in \mathbb{R}^n \mid \phi(y) \leq c\} \tag{2.35}$$

is strictly contained inside $\bar{B}_{\hat{\delta}}(X^*)$. We note that whenever $y \in \bar{B}_\sigma(X^*) \backslash \Phi_c$, $\|y - \mathbf{P}_{X^*}(y)\| \leq \sigma$. Therefore,

$$\alpha K_\phi \|y - \mathbf{P}_{X^*}(y)\|^2 \leq \alpha K_\phi \sigma^2, \quad \forall y \in \bar{B}_\sigma(X^*) \backslash \Phi_c, \tag{2.36}$$

for any positive, real α and K_ϕ. On the same set, $\phi(\cdot)$ is strictly larger than c. Taking

$$\alpha \in (0, \alpha_\phi), \quad \alpha_\phi = \frac{c}{K_\phi \sigma^2}, \tag{2.37}$$

implies that

$$\phi(y) > c \geq \alpha K_\phi \sigma^2, \tag{2.38}$$

whenever $y \in \bar{B}_\sigma(X^*) \backslash \Phi_c$. Together with (2.36), (2.38) implies (2.34), since $\bar{B}_{\hat{\delta}}(X^*) \supset \Phi_c$. □

[2] The reason for performing the second minimization of $\psi(\cdot)$ is the following. The radial unboundedness and positive definiteness of $\psi(\cdot)$ does not preclude the possibility that some of its sublevel sets are not connected (since $\psi(\cdot)$ need not be monotonically increasing in all directions away from Ψ_0), and it can therefore not be claimed that $\Psi_l \subset \bar{B}_{\hat{\rho}}(\Psi_0)$ for some $l \in (0, \bar{\bar{l}})$. If all sublevel sets of $\psi(\cdot)$ are connected, then $\bar{\bar{l}} = \bar{l}$.

2.5.2 Proof of Theorem 2.5.1

Should put rho geq 0

As in Definition 2.5.3, let σ and ρ be arbitrary positive real numbers with $\infty > \sigma > \rho$. Our first task is to show that under the theorem's assumptions, $\bar{B}_\rho(\Gamma_0) \cap \Xi$ is stable for (2.31) for a sufficiently small gain α.

We begin by constructing several sets. For any $r \in \mathbb{R}_{++}$, we let Γ_r denote the set $\{\xi \in \mathbb{R}^n \mid V(\xi) \leq r\}$ while $\partial\Gamma_r$ denotes its boundary $\{\xi \in \mathbb{R}^n \mid V(\xi) = r\}$. We note that since $V(\cdot)$ is assumed to be radially unbounded with respect to Γ_0, and Γ_0 is assumed to be compact, all the sublevel sets Γ_r are compact. Moreover, for any $r \in \mathbb{R}_{++}$, the intersection $\Gamma_r \cap \Xi$ is nonempty since Γ_0 is a subset of both Γ_r and Ξ.

By Lemma 2.5.1, there exists a number $l \in \mathbb{R}_{++}$ such that Γ_l is strictly contained inside $\bar{B}_\rho(\Gamma_0)$. Then, we have that $\bar{B}_\rho(\Gamma_0) \supset \Gamma_l \supset \Gamma_{l/2}$. Let

$$\delta = \min_{\xi \in \partial\Gamma_{l/2}} \|\xi - \mathbf{P}_{\Gamma_0}(\xi)\|, \tag{2.39}$$

so that $\bar{B}_\delta(\Gamma_0)$ is the largest "ball" contained in $\Gamma_{l/2}$ and "centered" at Γ_0. By assumption, there exists an $\alpha_Z \in \mathbb{R}_{++}$ such that $Z(\alpha) \subseteq \bar{B}_\delta(\Gamma_0)$, whenever $\alpha \in (0, \alpha_Z)$. To summarize our construction so far, we have

$$\bar{B}_\sigma(\Gamma_0) \supset \bar{B}_\rho(\Gamma_0) \supset \Gamma_l \supset \Gamma_{l/2} \supseteq \bar{B}_\delta(\Gamma_0) \supseteq Z(\alpha), \tag{2.40}$$

whenever $\alpha \in (0, \min\{\bar{\alpha}, \alpha_Z\})$.

Finally, we note that by assumption, there exists a number $\alpha_W \in \mathbb{R}_{++}$ such that $b_W(\alpha) \leq \frac{l}{2}$, whenever $\alpha \in (0, \alpha_W)$.

Claim 2.5.1: For any $r \in [l, \infty)$, the set $\Gamma_r \cap \Xi$ is positively invariant for (2.31), whenever $\alpha \in (0, \min\{\bar{\alpha}, \alpha_Z, \alpha_W\})$.

Proof. We will show that if for some $t \in \mathbb{N}$, $\xi \in \Gamma_r \cap \Xi$, then necessarily $\xi^+ \in \Gamma_r \cap \Xi$. Suppose that $\xi \in \Gamma_r \cap \Xi$. Then, since

$$\Gamma_r \supseteq \Gamma_l \supset \Gamma_{l/2} \supset Z(\alpha)$$

whenever $\alpha \in (0, \min\{\bar{\alpha}, \alpha_Z\})$ (q.v. (2.40)), we see that either $\xi \in Z(\alpha)$, or $\xi \in (\Gamma_r \cap \Xi) \backslash Z(\alpha)$.

If $\xi \in Z(\alpha)$, then $\xi \in \Gamma_{l/2} \cap \Xi$, and therefore $V(\xi) \leq \frac{l}{2}$. Also, since $\alpha \leq \alpha_W$, $W(\xi;\alpha) \leq b_W(\alpha) \leq \frac{l}{2}$. Therefore,

$$V(\xi^+) \leq V(\xi) + W(\xi;\alpha) \leq \tfrac{l}{2} + \tfrac{l}{2},$$

implying that $\xi^+ \in \Gamma_l \cap \Xi$, and therefore $\xi^+ \in \Gamma_r \cap \Xi$.

On the other hand if $\xi \in (\Gamma_r \cap \Xi) \backslash Z(\alpha)$, then $V(\xi) \leq r$, while $W(\xi;\alpha) < 0$ (q.v (2.33)). Therefore, $V(\xi^+) \leq r$, implying that $\xi^+ \in \Gamma_r \cap \Xi$. The conclusion then follows from the principle of induction. $\diamondsuit$

Let $\varepsilon \in \mathbb{R}_{++}$ be arbitrary. By Claim 2.5.1, $\Gamma_l \cap \Xi$ is invariant for (2.31). The invariance of $\Gamma_l \cap \Xi$ and the fact that $\bar{B}_\delta(\Gamma_0) \subset \Gamma_l \subset \bar{B}_\rho(\Gamma_0)$ (q.v. (2.40)) imply that whenever (2.31) is initialized inside $\bar{B}_\delta(\Gamma_0)$, the sequence $(\xi(t))_{t=1}^\infty$ remains inside $\bar{B}_\rho(\Gamma_0) \cap \Xi$ (and hence inside $\bar{B}_{\rho+\varepsilon}(\Gamma_0) \cap \Xi$) forever, provided that $\alpha \in (0, \min\{\bar{\alpha}, \alpha_Z, \alpha_W\})$. We therefore conclude that $\bar{B}_\rho(\Gamma_0)$ is stable for (2.31).

Our second task is to show that under the assumptions of the theorem, $\bar{B}_\rho(\Gamma_0)$ is attractive for (2.31) on $\bar{B}_\sigma(\Gamma_0)$. We begin with the following claim.

Claim 2.5.2: For every $\sigma \in \mathbb{R}_{++}$, there exist positive, real numbers $\hat{\sigma}$ and α^* such that $\forall t \in \mathbb{N}$, $\xi(t) \in \bar{B}_{\hat{\sigma}}(\Gamma_0) \cap \Xi$ whenever $\xi(0) \in \bar{B}_\sigma(\Gamma_0) \cap \Xi$ and $\alpha \in (0, \alpha^*)$.

Proof. The continuity of $V(\cdot)$ and the compactness of Γ_0 imply the existence of the number

$$\hat{l} = \max_{\xi \in \bar{B}_\sigma(\Gamma_0)} V(\xi). \tag{2.41}$$

Since $\Gamma_{\hat{l}}$ is compact, there exists a number $\hat{\sigma}$ such that $\bar{B}_{\hat{\sigma}}(\Gamma_0) \supseteq \Gamma_{\hat{l}}$; for example, one may take

$$\hat{\sigma} = \max_{\xi \in \partial \Gamma_{\hat{l}}} \|\xi - \mathbf{P}_{\Gamma_0}(\xi)\|, \tag{2.42}$$

which is well defined since $d(\cdot, \Gamma_0)$ is continuous (q.v. Remark 3.3.5). By the assumptions of the theorem, there exists a number $\hat{\alpha} \in \mathbb{R}_{++}$ such that (2.32) holds on $\bar{B}_{\hat{\sigma}}(\Gamma_0) \cap \Xi$. Then, by Claim 2.5.1, $\Gamma_{\hat{l}} \cap \Xi$ is positively invariant for (2.31) whenever $\alpha \in (0, \alpha^*)$, where

$$\alpha^* = \min\{\hat{\alpha}, \bar{\alpha}, \alpha_Z, \alpha_W\}, \tag{2.43}$$

and $\bar{\alpha}$, α_Z and α_W are as in Claim 2.5.1. Since

$$\bar{B}_{\hat{\sigma}}(X^*) \supseteq \Gamma_{\hat{l}} \supseteq \bar{B}_\sigma(X^*), \tag{2.44}$$

the invariance of $\Gamma_{\hat{l}} \cap \Xi$ proves the claim. $\diamondsuit$

The attractivity of $\bar{B}_\rho(\Gamma_0)$ for (2.31) on $\bar{B}_\sigma(\Gamma_0)$ follows from the following claim.

Claim 2.5.3: Let δ be as in (2.39). Then, for any $r \in [\delta, \sigma)$, the set $\bar{B}_r(\Gamma_0)$ is attractive for (2.31) on $\bar{B}_\sigma(\Gamma_0)$.

Proof. Let $\varepsilon \in (0, \sigma - r]$ be arbitrary, and let $\alpha \in (0, \alpha^*)$, with α^* as in Claim 2.5.2. The continuity of $W(\cdot;\alpha)$ implies that there exists a real number γ such that

$$-\gamma = \max_{\xi \in (\bar{B}_{\hat{\sigma}}(\Gamma_0) \setminus B_{r+\varepsilon}(\Gamma_0)) \cap \Xi} W(\xi;\alpha), \tag{2.45}$$

where $\hat{\sigma}$ is as in Claim 2.5.2. Since $W(\xi;\alpha) < 0$ for all $\xi \notin Z(\alpha)$, and $\bar{B}_{r+\varepsilon}(\Gamma_0) \supset \bar{B}_\delta(\Gamma_0) \supseteq Z(\alpha)$, we see that $\gamma > 0$.

From (2.32) and (2.45), we observe that whenever $\xi(0) \in (\bar{B}_\sigma(\Gamma_0) \setminus \bar{B}_{r+\varepsilon}(\Gamma_0)) \cap \Xi$, the inequality

$$V(\xi^+) \le V(\xi) - \gamma \tag{2.46}$$

holds for all $t \in \mathbb{N}$, so long as $\xi(t) \in (\bar{B}_{\hat{\sigma}} \setminus \bar{B}_{r+\varepsilon}(\Gamma_0) \cap \Xi$. Solving (2.46), we obtain that

$$V(\xi(t)) \le V(\xi(0)) - t\gamma, \tag{2.47}$$

which, together with the positive definiteness of $V(\cdot)$, shows that no sequence $(\xi(t))_{t=0}^\infty$, generated by (2.31) and initialized inside $\bar{B}_\sigma(\Gamma_0) \cap \Xi$ can remain in $(\bar{B}_{\hat{\sigma}}(\Gamma_0) \setminus B_{r+\varepsilon}(\Gamma_0)) \cap \Xi$ forever. Claim 2.5.2 implies that no such sequence can leave $\bar{B}_{\hat{\sigma}}(\Gamma_0) \cap \Xi$, and we therefore conclude that $(\xi(t))_{t=0}^\infty$ enters $\bar{B}_{r+\varepsilon}(\Gamma_0) \cap \Xi$ in finitely many iterations, proving the claim. $\diamondsuit$

The above claim implies that $\bar{B}_r(\Gamma_0)$, with $r = \rho$ is attractive for (2.31) on $\bar{B}_\sigma(\Gamma_0)$, and the theorem is proved. $\square$

2.5.3 Global Asymptotic Stability

The following characterization of global asymptotic stability may be regarded as a corollary of Theorem 2.5.1.

Theorem 2.5.2 (c.f. Corollary 5.9.9, [3])**:** Consider the system (2.31), and suppose there exists a function $V \in C^0[\mathbb{R}^n, \mathbb{R}_+]$ which is radially unbounded and positive definite with respect to $\Gamma_0 = \{\mathbf{0}\} \subset \Xi$ on $\mathbb{R}^n$. If there exists a function $W \in C^0[\mathbb{R}^n, \mathbb{R}]$ such that

$$V(\xi^+) - V(\xi) \leq W(\xi), \quad \forall \xi \in \Xi, \tag{2.48}$$

and $-W(\cdot) \in \mathscr{K}$, then Γ_0 is globally asymptotically stable for (2.31) on Ξ, with Lyapunov function $V(\cdot)$. $\diamondsuit$

Chapter 3

An Analytic Framework for Consensus Optimization Methods: The Interconnected Systems Approach

3.1 Synopsis

As a preliminary step toward addressing the class of DT-DDCCPs represented by Problem 1.2.2, we propose an analytic framework for the derivation of convergence conditions for a class of consensus optimization algorithms. This framework is founded on concepts from the theory of large-scale interconnected systems. As such, its development is intended to accommodate the analysis of multiagent coordination control designs involving the interaction of consensus optimization algorithms with dynamical systems. The convergence concept we emphasize for such applications is that of *semiglobal, practical, asymptotic stability* (SPAS). A key tool that we developed for our analysis is Theorem 2.5.1, which characterizes the SPAS of the set of fixed points associated with a general, nonlinear, discrete-time system, in terms of a Lyapunov function. We apply this theorem in deriving convergence conditions for two subclasses of consensus optimization algorithms: those whose search directions are bounded but possibly multivalued, and those whose search directions are locally-Lipschitz continuous. For the latter case, our conditions indicate that all agents' individual cost functions need not be convex. In both cases, we study generally constrained problems in which agents' individual constraint sets are assumed to be neither bounded nor identical. The interconnected systems point of view allows us to show that the projection operation used in enforcing the constraints need not add any conservatism to the convergence condition. In contrast with existing Lyapunov-based analyses of related decentralized optimization schemes, the application of our SPAS theorem does not require a precise characterization of the algorithm's actual set of fixed points, and allows for the study of iterative methods directly in discrete-time.

3.2 Introduction

Much of the current research effort in decentralized optimization (DOpt) is motivated primarily by sensor network and machine learning applications. As such, much of this effort is directed toward quantifying and improving the convergence rates of various specific DOpt algorithms [165], [29], [72], [127], [116], [44]. Although undoubtedly important, convergence rate analyses usually rely on techniques that may not be suitable for studying the efficacy of such algorithms in the context of coordination control, where the internal stability of the coupled system involving the agents and the DOpt algorithm is the primary concern.

By contrast, the application area that motivates the present work, and thereby prescribes our focus, concerns the design of decentralized coordination control strategies for networked multiagent systems. As detailed in §1.2.3, the essential problem is to design rules by which agents individually operate and interact in order to produce a desired collective behaviour, or accomplish some non-trivial task in a way that leverages the synergism of the collective. In multiagent coordination control scenarios for which the coordination objective can be expressed within the formalism of convex optimization, the solution to the collective optimization problem may vary with time [27], [89], or according to the environmental conditions faced by the agents. More importantly, the agents themselves may be non-static entities, and their dynamic behaviour may affect the performance of the DOpt algorithm. Though undoubtedly important, convergence rate analyses typically rely on techniques that are not necessarily suitable for studying the interaction of DOpt algorithms with dynamic systems.

For example, the convergence concept typically emphasized in the optimization literature is the *attractivity* of fixed points. Although the *stability* of an algorithm's fixed points can sometimes be inferred indirectly from the ensuing convergence rate analysis, stability is almost never explicitly addressed [173], [114], [116], [27], [44]. The stability of fixed points is especially unclear when the convergence rate analysis is carried out exclusively by means of auxiliary sequences such as Cesàro (i.e. running) averages of the algorithm's iterates [44], [165]; a sequence of Cesàro means may converge even if the original sequence is not well behaved. Another example concerns the common use of diminishing step-size rules [173], [116], [72], [27], [16], [44], which, as noted in [30], may not be appropriate in online applications in which the optimization process is ongoing.

What is needed then, is a set of tools and techniques that are more innately suited to the analysis and design of DOpt algorithms in the context of coordination control for dynamic multiagent systems.

3.2.1 Objectives, Method and Contributions

Our main objective is to develop a broadly applicable analytic framework for the derivation of convergence conditions for a large class of consensus optimization (CO) algorithms. In contrast with existing analytic techniques, we want this framework to enable both the stability and convergence rate analysis of general CO methods, and to accommodate settings in which the agents may themselves be dynamic entities, whose dynamics couple to those of the CO algorithm and affect its performance.

To achieve this, we base our method on concepts drawn from the literature on interconnected dynamical systems [161], [104]. As is commonly done in the literature, our analysis begins by examining the evolution of the mean estimate across agents, and the vector of deviations from that mean [114], [116], [44], [173], [127]. In contrast to existing approaches, we interpret the evolution of these variables as a feedback interconnection of two nonlinear, discrete-time, dynamical systems. When the agents executing the CO algorithm are dynamic entities, their individual dynamics may then be regarded as additional subsystems coupled to this interconnection, and thereby be addressed within the same framework. This observation is used to derive the results of Chapter 5.

Although dynamic multiagent coordination control problems remain our primary motivation, in this chapter we restrict our attention to the scenario in which the agents are static entities, and we investigate whether the said framework allows us to learn anything new about CO algorithms themselves. In doing so, our secondary objective is to broaden the set of tools available for the design and analysis of specific variants of CO algorithms, and to promote the utility of certain system-theoretic analysis techniques that are largely absent from the optimization literature.

We study generally constrained problems in which each agent projects its estimate updates onto its individual, privately known constraint set. We find that these constraint sets need not be bounded or identical; they are only assumed to be closed and convex, and to have a non-empty intersection.

In the literature, the effects of the dynamic coupling between the mean and deviation variables are usually suppressed by a combination of two persisting assumptions: the boundedness of agents' individual constraint sets (or the subgradients used to form their search directions), and the use of diminishing step sizes. Instead of suppressing the effects of dynamic coupling among subsystems, the literature on interconnected systems emphasizes techniques that seek to exploit this coupling in order to avoid conservatism [104]. Indeed, in our case a careful examination of the interconnection structure reveals that the destabilizing effects of the constraint-related projection terms arising in one subsystem negate related effects arising in the other (q.v. Lemma 3.6.3).

The destabilizing effects of all remaining interconnection terms can be dominated by means of a "small-gain" argument in the composite Lyapunov analysis by exploiting the stability properties of the idealized (i.e. isolated) subsystem dynamics. The final convergence condition is an upper bound on the algorithm step-size, which may remain fixed throughout the execution of the algorithm.

The convergence concept of interest to us is that of *semiglobal, practical, asymptotic stability* (SPAS) [149]. One of the analytic tools that we contribute is a theorem characterizing the SPAS of a constrained, nonlinear, discrete-time dynamical system, in terms of certain properties of a Lyapunov function (q.v. Theorem 2.5.1).

We apply this theorem in deriving convergence conditions for two sub-classes of CO algorithms: those in which each agent's search direction is bounded but possibly multi-valued, and those in which it is a locally Lipschitz function. An important outcome for the case in which each agent's search direction is locally Lipschitz and based on a gradient of the agent's private cost function, is that the convexity of all agents' individual costs is not necessary (q.v. Lemma 3.8.1).

Generally speaking, Lyapunov techniques are favoured in the analysis of dynamical systems because they facilitate studies of robustness to various disturbances and uncertainties. As early as 1958, Lyapunov stability and LaSalle invariance concepts were used to establish the asymptotic stability of continuous-time saddle point dynamics arising in certain classes of separable, linearly constrained resource allocation problems [8]. These techniques have later been adapted to the study of similar dynamics arising in network utility maximization problems [79], [99], [48] and separable problems with consensus constraints [163], [54]. Lyapunov techniques also appear in [127], [140] and [16].

In contrast with this literature, the Lyapunov tools we provide here enable the study of iterative algorithms directly in discrete time, though the same general framework is useful in the study of continuous-time CO schemes as well (q.v. Chapter 4). Secondly, the concept of SPAS is more general than that of asymptotic stability. Specifically, the application of our SPAS theorem does not require a precise characterization of an algorithm's *actual* set of fixed points. This feature is relevant to the study of CO algorithms since their fixed points may not generally coincide with the set of points satisfying the KKT conditions associated with the collective optimization problem; CO algorithms achieve perfect consensus if, and only if the sets of optima pertaining to agents' individual objective functions have a nonempty intersection [140], and their fixed points may otherwise be difficult to characterize. Finally, the constraints considered in this literature are usually specially structured (i.e., often linear), and are approximately enforced throughout the execution of the algorithm. We instead consider generally constrained problems in which the constraints are enforced exactly at each iteration by means of a projection operation, and we provide an elegant way of studying its effects on the evolution of CO algorithms.

The remainder of this chapter is organized as follows. We specify our problem setting and the class of CO algorithms being considered, and state our main assumptions in §3.3. In §3.4, we provide preliminary analysis results. We state our main results in §3.6 and we demonstrate their application to a class of weighted-gradient CO algorithms in §3.8. Several lemmas whose proofs are omitted from the body of this chapter are provided in §3.9. The chapter is concluded in §3.10.

3.3 Problem Setting

The problem setting described in this section is in essence the same as that considered in [114] and [116], and we strive to use some of the same notation.

We consider a system of N agents networked over a graph $\mathscr{G}_C = (\mathscr{V}, \mathscr{E}_C(t))$, where $\mathscr{V} = \{1, \ldots, N\}$ is a set that indexes the agents, and $\mathscr{E}_C(t) \subseteq \mathscr{V} \times \mathscr{V}$ specifies the pairs of agents that communicate at time t.

The agents' objective is to cooperatively locate any solution to the collective optimization problem

$$\min_{x \in X} J(x), \tag{3.1}$$

where $J : \mathbb{R}^n \to \mathbb{R}$, and $X \subseteq \mathbb{R}^n$ is some nonempty, closed, convex set which is not necessarily bounded. We assume that $J(\cdot)$ is such that the set of its minimizers

$$X^* = \arg\min_{x \in X} J(x) \tag{3.2}$$

is nonempty, convex and compact.

The function $J(\cdot)$ is comprised of N additive components, and X is given as the intersection of N closed, convex sets – namely,

$$J(x) = \sum_{i=1}^{N} J_i(x), \tag{3.3}$$

and

$$X = \bigcap_{i \in \mathscr{V}} X_i. \tag{3.4}$$

For all $i \in \mathscr{V}$, $J_i(\cdot)$ and X_i are known only to agent i, while the set X and the collective cost $J(\cdot)$ are not known to any of the agents.

In what follows, we develop an analytic framework for the study of a class of decentralized optimization algorithms in which the i'th agent iteratively updates its estimate of some $x^* \in X^*$, according to the rule

$$\begin{aligned} x_i(t+1) &= \mathbf{P}_{X_i}\left[v_i(t) - \alpha s_i(v_i(t))\right], \quad \forall t \in \mathbb{N}, \\ v_i(t) &= \sum_{i=1}^{N} [A]_{i,j}(t) x_j(t), \end{aligned} \tag{3.5}$$

where $x_i \in \mathbb{R}^n$, $\alpha \in \mathbb{R}_{++}$ is the "step size", $A(t) \in \mathbb{R}^{N \times N}$ is a weighted adjacency matrix associated to $\mathscr{G}_C$, and

$$s_i(v_i(t)) \in (\Theta \circ J_i)(v_i(t)) \tag{3.6}$$

is agent i's search direction at iteration t, generated by the operator $\Theta : C^0[\mathbb{R}^n, \mathbb{R}] \rightrightarrows C^0[\mathbb{R}^n, \mathbb{R}^n]$, whose properties we specify in the upcoming §3.3.2. The gain α and the operator $\Theta(\cdot)$ are assumed to be known by all agents.

Remark 3.3.1: In this chapter we are not focused on addressing DCCPs per se. However, in relation to the discrete-time, *static* DCCP formulated in Chapter 1, the set X^* may be regarded as corresponding to the goal set, and x_i as corresponding to agent i's decision (or action) variable (q.v. Remark 1.2.7 regarding notation). The agents' objective is to reach an agreement on some $x^* \in X^*$; specifically, the agents aim to update their "decisions" x_i such that

$$\lim_{t\to\infty} x_1(t) = \cdots = \lim_{t\to\infty} x_N(t) = x^* \in X^*. \tag{3.7}$$

We note that in (3.5) it is implicitly assumed that agent i is capable of evaluating the quantity $s_i(\cdot) = (\Theta \circ J_i)(\cdot)$ in (3.6) anywhere within $\mathbb{R}^n$, and in particular at $v_i(t)$. For these reasons, the problem setting in this chapter may be regarded as corresponding to a subclass of DT-SDCCPs represented by Problem 1.2.2 (q.v. Example 1.3.1, Remark 1.3.1, Example 1.4.1 and Remark 1.4.2). $\diamondsuit$

3.3.1 Properties of the Communication Graph $\mathscr{G}_C$

We assume that the weighted adjacency matrix $A(t)$ has the following properties.

A3.3.1 (Properties of $A(t)$)**:** The entry $[A]_{i,j}(t) \neq 0$ if, and only if $(j,i) \in \mathscr{E}_C(t)$. For all $t \in \mathbb{N}$, $A(t)$ is stochastic and symmetric, with eigenvalues

$$1 = \lambda_1(A(t)) > \lambda_2(A(t)) \geq \cdots \geq \lambda_N(A(t)) > -1. \tag{3.8}$$

Moreover, $\forall t \in \mathbb{N}$, $\lambda_2(A^2(t)) = \max\{\lambda_2^2(A(t)), \lambda_N^2(A(t))\} \in [0,\mu]$, for some $\mu \in [0,1)$. $\diamondsuit$

According to the discussion in §2.2, the weighted adjacency matrix $A(t)$ in A3.3.1 would have the desired spectrum (3.8) if $\mathscr{G}_C$ is connected and has at least one positively-weighted self-loop – i.e., $[A]_{i,i} > 0$, for some $i \in \mathscr{V}$. Remark 2.3.1 describes how agents may choose the link weights in a decentralized way in order to produce a weighted adjacency matrix satisfying A3.3.1.

Remark 3.3.2: Consider a matrix $A(t) \otimes I_n$, where $A(t)$ satisfies A3.3.1, and let e_i denote the unit vector along the ith coordinate of $\mathbb{R}^n$. By the properties of the Kronecker product, $\lambda \in \sigma(A(t))$ implies that $\lambda \in \sigma(A(t) \otimes I_n)$, with multiplicity n. Therefore, the null space of $A(t) \otimes I_n - \lambda_1(A(t)) I_{Nn}$, which is sometimes referred to as the "agreement subspace", is spanned by $\{\mathbf{1}_N \otimes e_1, \ldots, \mathbf{1}_N \otimes e_n\}$, and for any vector $z \in \mathbb{R}^{Nn}$ inside the orthogonal complement of the agreement subspace, it holds that

$$\|(A(t) \otimes I_n) z\|^2 \leq \mu \|z\|^2 < \|z\|^2. \tag{3.9}$$

$\diamondsuit$

Remark 3.3.3: An important outstanding challenge in the development of decentralized optimization algorithms is to relax the assumptions typically made on the agents' interaction model [152]. In particular, several recent research efforts are focused on the analysis of CO schemes involving directed or asynchronous communication among agents [112], [16], [53]. Since this is not the focus of the present work, we adopt the stronger assumption A3.3.1 in order to simplify our presentation. Assumptions similar to A3.3.1 are common in the literature [75], [153]. $\diamondsuit$

3.3.2 Properties of the Search Directions $s_i(\cdot)$

One of the rare monographs in which Lyapunov techniques are explicitly used in the analysis of optimization methods is [124]. Therein one finds notions of "pseudogradient" and "strongly pseudogradient", which are properties ascribed to the search directions employed by a generic class of iterative (centralized) optimization methods (q.v. §2.2, [124]). We generalize these notions in the following definition.

Definition 3.3.1: Consider a multi-valued vector field $\Psi : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ and a differentiable function $V : \mathbb{R}^n \to \mathbb{R}$, which is positive definite with respect to a compact set $\Gamma_0 \subset \mathbb{R}^n$. We say that $\Psi(\cdot)$ is *strictly pseudogradient* with respect to $V(\cdot)$ on a set $\Xi \supset \Gamma_0$, if there exists a function $\phi \in C^0[\mathbb{R}^n, \mathbb{R}_+]$ which is radially unbounded and positive definite with respect to Γ_0 on Ξ, and for each $s \in \Psi(\xi)$ it holds that

$$\nabla V(\xi)^T s \geq \phi(\xi), \quad \forall \xi \in \Xi. \tag{3.10}$$

$\diamondsuit$

Remark 3.3.4: Although the strict pseudogradient property is weaker than the *strong pseudogradient property*, which requires that

$$\nabla V(\xi)^T s \geq \tau V(\xi), \quad \forall s \in \Psi(\xi), \forall \xi \in \mathbb{R}^n, \tag{3.11}$$

and some $\tau > 0$, it is stronger than the *pseudogradient property* (q.v. §2.2, [124]), which requires that

$$\nabla V(\xi)^T s \geq 0, \quad \forall s \in \Psi(\xi), \forall \xi \in \mathbb{R}^n. \tag{3.12}$$

$\diamondsuit$

We use Definition 3.3.1 to state our main assumption on the set of search directions generated by $(\Theta \circ J)(\cdot)$.

A3.3.2 (Strictly Pseudogradient Search Directions)**:** The (possibly multi-valued) vector field $(\Theta \circ J)(\cdot)$ is strictly pseudogradient with respect to the function $y \mapsto \|y - \mathbf{P}_{X^*}(y)\|^2$, on the set

$$\hat{X} = \mathrm{co}\left(\cup_{i \in \mathscr{V}} X_i\right). \tag{3.13}$$

$\diamondsuit$

Remark 3.3.5: A well known fact, attributable to Motzkin [109], is that the distance function

$$d(y, X^*) = \min_{x_o \in X^*} \|y - x_o\| = \|y - \mathbf{P}_{X^*}(y)\| \tag{3.14}$$

is differentiable at each $x \in \mathbb{R}^n \backslash X^*$, if, and only if X^* is closed, nonempty and convex. Using the definition of Fréchet differentiability and the properties of the projection operator, one can show that $2(x - \mathbf{P}_{X^*}(x))$ is the gradient of the function $d(y, X^*)^2$. Consequently, A3.3.2 is equivalent to having

$$2(y - \mathbf{P}_{X^*}(y))^T s \geq \phi(y), \quad \forall s \in \Theta(J(y)), \forall y \in \hat{X}, \tag{3.15}$$

for some function $\phi(\cdot)$ having the properties in Definition 3.3.1. $\diamondsuit$

Remark 3.3.6: For the case in which $J(\cdot)$ is convex and $\Theta(J(y)) = \partial J(y)$, inequality (3.15) is satisfied, for example, by taking $\phi(y) = 2(J(y) - J^*)$, where J^* denotes the value of $J(\cdot)$ on X^*. $\diamondsuit$

We also assume that $\Theta(\cdot)$ is a linear operator.

A3.3.3 (Linearity of Θ)**:** For any $y \in \mathbb{R}^n$, and for each $i \in \mathscr{V}$, let $s_i(y)$ be an arbitrary element of $\Theta(J_i(y))$ and c_i be an arbitrary real number. Then,

$$\sum_{i \in \mathscr{V}} c_i s_i(y) \in \Theta\Big(\sum_{i \in \mathscr{V}} c_i J_i(y)\Big), \quad \forall y \in \mathbb{R}^n. \tag{3.16}$$

$\diamondsuit$

Remark 3.3.7: By Lemma 10 in §5 [124], A3.3.3 is satisfied for the case in which $\Theta(J(y)) = \partial J(y)$, and $J(\cdot)$ is convex. $\diamondsuit$

In the analyses that follow, it is possible to draw meaningful conclusions from either of the following two alternative assumptions on the search directions $s_i(\cdot)$.

A3.3.4 (Locally Lipschitz Search Directions)**:** For any $y \in \mathbb{R}^n$, and for each $i \in \mathscr{V}$, $\Theta(J_i(y)) = \{s_i(y)\}$, and the functions $s_i(y)$ are locally Lipschitz – i.e., for any compact $\Omega \subset \mathbb{R}^n$, there exists an $L_i \in \mathbb{R}_{++}$ such that for all v and y in Ω,

$$\|s_i(v) - s_i(y)\| \leq L_i \|v - y\|. \tag{3.17}$$

$\diamondsuit$

A3.3.4′ (Locally Bounded Search Directions)**:** For any compact $\Omega \subset \mathbb{R}^n$, there exists a $B \in \mathbb{R}_{++}$ such that for all $y \in \Omega$ and $i \in \mathscr{V}$, each $s_i(y) \in \Theta(J_i(y))$ satisfies

$$\|s_i(y)\| \leq B. \tag{3.18}$$

$\diamondsuit$

Remark 3.3.8: The case studied in [116] corresponds to $\Theta(J_i(v_i)) = \partial J_i(v_i)$, and the analysis is carried out under the assumption that the subgradients $s_i(v_i)$ are bounded. This is a reasonable assumption for problems involving polyhedral $J_i(\cdot)$, which may arise as dual functions in certain integer programming problems (q.v. §8.2, [14]). $\diamondsuit$

For the case in which the search directions $s_i(\cdot)$ are locally bounded but not necessarily locally Lipschitz continuous, we find it useful to strengthen assumption A3.3.2 to the following:

A3.3.2′: For each $i \in \mathscr{V}$, there exists a locally Lipschitz function $\phi_i : \mathbb{R}^n \to \mathbb{R}$, such that for any $\xi \in \mathbb{R}^n$ and any $s_i(\xi) \in (\Theta \circ J_i)(\xi)$,

$$2(\xi - x^*)^T s_i(\xi) \geq \phi_i(\xi), \quad \forall x^* \in X^*. \tag{3.19}$$

Moreover, there exists a function $\phi \in C^0[\mathbb{R}^n, \mathbb{R}_+]$ which is radially unbounded and positive definite with respect to X^* on $\hat{X}$, such that

$$\phi(\xi) \leq \sum_{i \in \mathscr{V}} \phi_i(\xi), \quad \forall \xi \in \hat{X}, \tag{3.20}$$

where $\hat{X}$ is as in (3.13).

Remark 3.3.9: The case in which $(\Theta \circ J_i)(\cdot) = \partial J_i$ and $J_i(\cdot)$ is assumed to be convex for each $i \in \mathscr{V}$ is studied in [116]. The assumption that $J_i(\cdot)$ is convex can be seen to imply A3.3.2′ by taking $\phi_i(\xi) = 2J_i(\xi) - 2J_i(\mathbf{P}_{X^*}(\xi))$, and noting that every convex function $J_i(\cdot) : \mathbb{R}^n \to \mathbb{R}$ is locally Lipschitz over its domain of definition [133]. In that case the sum $\sum_{i \in \mathscr{V}} \phi_i(\cdot)$ is equal to $\phi(\cdot) = 2(J(\cdot) - J^*)$, which is radially unbounded and positive definite with respect to X^* on $\hat{X}$. In other words, assumptions A3.3.4′ and A3.3.2′ taken together are no stronger than those assumptions adopted in [116]. $\diamondsuit$

Remark 3.3.10 (The Rationale behind A3.3.2 and A3.3.2′)**:** Many recent contributions in decentralized optimization present convergence analyses of specific algorithm structures designed to perform well on specific, often restricted classes of cost functions. For example, in addition to the usual requirement that each $J_i(\cdot)$ be convex, [29], [145], and [127] derive results that exploit the special form of cost functions arising in certain classes of parameter fitting problems, and [72] requires that agents' individual cost functions are all convex and differentiable, with gradients that are Lipschitz continuous and bounded.

Instead of directly restricting the class of objective functions under consideration, assumptions such as A3.3.2 and A3.3.4, or A3.3.4′ and A3.3.2′ place the focus on the structure of the algorithm itself. As such, they help align our viewpoint more closely with those typical to control system design and analysis. Secondly, although the search directions $s_i(\cdot)$ relate to the individual and collective cost functions by means of assumptions A3.3.2 (or A3.3.2′) and A3.3.3, the relationship implied by these assumptions is quite generic. It is hoped that the generality of this alternative perspective may serve to inform proposals for new designs of specific CO algorithms and associated objective function classes. ♢

3.4 The Interconnected Systems Point of View

Building on the ideas originally proposed in [91], we present the essence of our proposed analytic framework in this section (A detailed description is given in §3.4.2). The analysis begins by examining the evolution of the mean estimate of a collective optimizer $x^* \in X^*$ among agents, and the vector of deviations of agents' individual estimates from this mean (q.v. §3.4.1). The idea proposed in [91] is to interpret the evolution of these two variables as a feedback interconnection of two coupled dynamical systems. In contrast with the present contribution, the work in [91] considers a narrow class of CO algorithms and unconstrained optimization problems in which the collective cost is assumed to be strongly convex.

3.4.1 The Mean and Deviation Subsystems

To begin, we write (3.5) more compactly as

$$\begin{aligned} x^+ &= (A \otimes I_n)x - \alpha s(v) + \eta \\ &= (A \otimes I_n)x - \alpha s((A \otimes I_n)x) + \eta, \end{aligned} \tag{3.21}$$

where $x = [x_1(t)^T, \dots, x_N(t)^T]^T \in \mathbb{R}^{Nn}$, $v = [v_1(t)^T, \dots, v_N(t)^T]^T \in \mathbb{R}^{Nn}$, $s(v) = [s_1(v_1(t))^T, \dots, s_N(v_N(t))^T]^T$ and $\eta = [\eta_1^T, \dots, \eta_N^T]^T$, with

$$\eta_i = \mathbf{P}_{X_i}\big[v_i(t) - \alpha s_i(v_i(t))\big] - \big[v_i(t) - \alpha s_i(v_i(t))\big], \tag{3.22}$$

and $s_i(v_i(t))$ and $v_i(t)$ as in (3.6) and (3.5).

As in [116], we consider the agents' mean estimate

$$y = \tfrac{1}{N}(\mathbf{1}_N^T \otimes I_n)x, \tag{3.23}$$

which, based on (3.21), evolves according to

$$y^+ = \tfrac{1}{N}(\mathbf{1}_N^T \otimes I_n)(A \otimes I_n)x - \tfrac{\alpha}{N}(\mathbf{1}_N^T \otimes I_n)s(v) + \tfrac{1}{N}(\mathbf{1}_N^T \otimes I_n)\eta. \tag{3.24}$$

Since A is assumed to be stochastic and symmetric, it is doubly stochastic – i.e., $\mathbf{1}_N^T A = \mathbf{1}_N^T$. The properties of the Kronecker product therefore imply that $(\mathbf{1}_N^T \otimes I_n)(A \otimes I_n) = (\mathbf{1}_N^T A \otimes I_n) = (\mathbf{1}_N^T \otimes I_n)$, which, together with (3.23), yields

$$y^+ = y - \tfrac{\alpha}{N}(\mathbf{1}_N^T \otimes I_n)s(v) + \tfrac{1}{N}(\mathbf{1}_N^T \otimes I_n)\eta. \tag{3.25}$$

Next, we introduce the deviation variable $z = [z_1^T, \ldots, z_N^T]^T \in \mathbb{R}^{Nn}$, with $z_i = x_i - y$. Equivalently,

$$z = (M \otimes I_n)x, \tag{3.26}$$

where

$$M = I_N - \tfrac{1}{N}\mathbf{1}_N\mathbf{1}_N^T. \tag{3.27}$$

In the literature on consensus algorithms, the deviation variable z is often termed the "disagreement vector" [136], and it is useful to note that for all $x \in \mathbb{R}^{Nn}$, z belongs to the orthogonal complement of the agreement subspace (q.v. Remark 3.3.2).

Based on (3.21), the deviation variable evolves according to

$$z^+ = (M \otimes I_n)(A \otimes I_n)x - \alpha(M \otimes I_n)s(v) + (M \otimes I_n)\eta. \tag{3.28}$$

From (3.27) and the double stochasticity of A, we see that $MA = AM$, so that by the properties of the Kronecker product, we have $(M \otimes I_n)(A \otimes I_n)x = (A \otimes I_n)(M \otimes I_n)x$. Then, from the definition of z in (3.26), we obtain

$$z^+ = (A \otimes I_n)z - \alpha(M \otimes I_n)s(v) + (M \otimes I_n)\eta. \tag{3.29}$$

Equations (3.25) and (3.29) can be expressed exclusively in terms of the mean and deviation variables by noting that

$$v = (A \otimes I_n)z + \mathbf{1}_N \otimes y. \tag{3.30}$$

We therefore observe that (3.25) and (3.29) represent two coupled nonlinear difference equations that can be interpreted as a feedback interconnection of two dynamical systems, as shown in Figure 3.1.

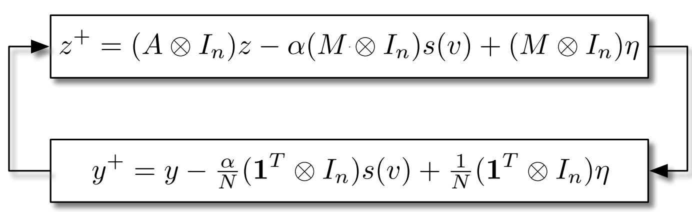

Figure 3.1: The dynamic coupling between the mean and deviation subsystems arises from the gradient terms, and the terms related to the projection operation.

3.4.2 Small-Gain Techniques, Semiglobal, Practical, Asymptotic Stability, and Interconnected Systems Techniques

There is a variety of powerful control theoretic tools available for the analysis of feedback interconnected systems such as those in Figure 3.1. We choose to appeal to those developed for the study of large-scale, or interconnected nonlinear systems (q.v. [104], [161], for example), though other approaches based on concepts such as passivity or input-to-state stability may well lead to the derivation of a different set of insights and convergence conditions.

The essential strategy in our approach is to identify the *idealized* subsystem dynamics associated with the *actual* mean and deviation subsystems depicted in Figure 3.1. We refer to these dynamics as "idealized" because they posses desirable stability properties that can be exploited in deriving convergence conditions for the overall interconnection. We then aim to express the actual mean and deviation dynamics as consisting of these idealized dynamics, additively perturbed by various interconnection terms. The main challenge then is to apply the well-known "art" of *small-gain* argumentation in order to derive conditions under which the destabilizing effects of the interconnecting terms, viewed as perturbations of the idealized dynamics, are guaranteed to be overcome by the stability properties of the idealized subsystem dynamics.

The convergence concept of interest to us is that of *semiglobal, practical, asymptotic stability* (SPAS). This concept is defined in §2.5, where we also prove an important theorem that characterizes the SPAS of fixed points associated with a general, nonlinear, discrete-time dynamical system, in terms of a Lyapunov function with certain properties (q.v. Theorem 2.5.1). This theorem allows us to be precise about specifying the conditions under which the stability properties of the idealized dynamics dominate the destabilizing effects of the interconnection terms.

An important aspect of the "art" of small gain argumentation that is easily and often overlooked, is the possibility that dynamically interacting subsystems may posses interconnection terms that are not necessarily antagonistic to the stability of the overall feedback loop[1] If one is careful in examining the interconnection structure of a system such as that shown in Figure 3.1, one may sometimes find that certain interconnection terms arising in one subsystem actually help stabilize the other subsystem. When this happens, these "helpful" terms need not be treated as perturbations in the small-gain argument. As a consequence, such terms ultimately need not add any conservatism to the final small-gain conditions. Indeed, in our case we find that all projection-related interconnection terms arising in one subsystem negate the effects of analogous terms arising in the other subsystem. These observations are made explicit in Lemma 3.6.3.

The combination of small gain techniques and the theory of SPAS, together with interconnected systems techniques constitutes the essence of our proposed analytic framework.

3.5 The Idealized Subsystem Dynamics and their Stability Properties

Our aim here is to identify the "idealized" dynamics associated to the mean and deviation subsystem dynamics given by (3.25) and (3.29), and to specify their stability properties.

3.5.1 The Idealized Mean Subsystem Dynamics

We write expression (3.25) as

$$y^+ = y - \tfrac{\alpha}{N} s_o(y) + p_1(y,z) + p_2(y,z), \tag{3.31}$$

where

$$p_1(y,z) = \tfrac{\alpha}{N}(\mathbf{1}_N^T \otimes I_n)\big(s(\mathbf{1}_N \otimes y) - s(v)\big), \tag{3.32}$$

$$p_2(y,z) = \tfrac{1}{N}(\mathbf{1}_N^T \otimes I_n)\eta, \tag{3.33}$$

and

$$s_o(y) = (\mathbf{1}_N^T \otimes I_n) s(\mathbf{1}_N \otimes y) = \sum_{i \in \mathscr{V}} s_i(y), \tag{3.34}$$

with $s_i(y) \in \Theta(J_i(y))$.

[1] For example, the ability to exploit this observation is often promoted as one of the key strengths of a popular constructive nonlinear control design technique known as *backstepping* (q.v. "Avoiding Cancellations", §2.2, [85]). These observations are also emphasized throughout [104].

Then, from our definition of the mean estimate y in (3.23), we see that for each $t \in \mathbb{N}$, $y(t)$ belongs to $\hat{X}$, the convex hull of $\bigcup_{i\in\mathscr{V}} X_i$ – i.e., $y(t) = \frac{1}{N}x_1(t) + \cdots + \frac{1}{N}x_N(t)$, and $x_i(t) \in X_i$, for all $i \in \mathscr{V}$ and $t \in \mathbb{N}$. Consequently, the update law

$$y^+ = \mathbf{P}_{\hat{X}}\left[y - \tfrac{\alpha}{N}s_o(y) + p_1(y,z) + p_2(y,z)\right] \tag{3.35}$$

generates the same set of iterates as (3.31), provided the two are initialized at the same place.

We observe that by A3.3.3, the definition of $s_o(\cdot)$ in (3.34) implies that $s_o(y) \in \Theta(J(y))$, and that A3.3.2 therefore applies to $s_o(\cdot)$. This observation can be used to show that the function $V_{\mathscr{O}}(y) = N\|y - \mathbf{P}_{X^*}(y)\|^2$ is a Lyapunov function for the idealized mean dynamics

$$y^+ = \mathbf{P}_{\hat{X}}\left[y - \tfrac{\alpha}{N}s_o(y)\right], \tag{3.36}$$

under either assumption A3.3.4 (q.v. Lemma 3.9.3) or A3.3.4′ (q.v. Lemma 3.9.5). Specifically, we have the following lemma, whose proof is postponed until §3.9.

Lemma 3.5.1: Consider the algorithm (3.36), where $s_o(y) \in \Theta(J(y))$. Suppose that A3.3.3 and A3.3.2 hold in addition to either A3.3.4 or A3.3.4′. Then, by Theorem 2.5.1, X^* is SPAS for (3.36), with Lyapunov function $V_{\mathscr{O}}(y) = N\|y - \mathbf{P}_{X^*}(y)\|^2$.

Proof. For the set of assumptions A3.3.3, A3.3.2 and A3.3.4, see the combined proofs of Lemmas 3.9.1, 3.9.2, and 3.9.3. For the set of assumptions A3.3.3, A3.3.2 and A3.3.4′, see the combined proofs of Lemmas 3.9.1, 3.9.4, and 3.9.5. ♢

Given that $s_o(\cdot)$ is assumed to satisfy the relationship (3.15) on $\hat{X}$, the idealized mean dynamics (3.36) can be regarded as a *centralized* optimization algorithm on $J(\cdot)$.

3.5.2 The Idealized Deviation Subsystem Dynamics

We express (3.29) as

$$z^+ = (A \otimes I_n)z + p_3(y,z), \tag{3.37}$$

where

$$p_3(y,z) = (M \otimes I_n)\big(\eta - \alpha s(v)\big). \tag{3.38}$$

The interconnection term $p_3(\cdot,\cdot)$ perturbs the evolution of the idealized deviation dynamics, which we regard to be given by

$$z^+ = (A \otimes I_n)z, \quad z(0) \in \mathscr{A}^\perp, \tag{3.39}$$

where

$$\mathscr{A}^\perp = \{\mathbf{1}_N \otimes e_1, \ldots, \mathbf{1}_N \otimes e_n\}^\perp, \tag{3.40}$$

and e_i is the unit vector along the ith coordinate in $\mathbb{R}^n$. The constraint on the initial condition in (3.39) follows from the definition of z in (3.26); the matrix $(M \otimes I_n)$ is the realization of an orthogonal projection onto the set $\mathscr{A}^\perp$. In particular, $z(0) = (M \otimes I_n)x(0)$ implies that $z(0) \in \mathscr{A}^\perp$ for any $x(0) \in \mathbb{R}^n$.

The stability properties of (3.39) are summarized in the following lemma.

Lemma 3.5.2: Consider the algorithm (3.39), and suppose that A3.3.1 holds. Then, by Theorem 2.5.2, $\{\mathbf{0}\}$ is globally asymptotically stable for (3.39) on $\mathscr{A}^\perp$, with Lyapunov function $V_{\mathscr{C}} = \|z\|^2$.

Proof. The proof is given in §3.9.2. ♢

In the absence of the perturbation $p_3(\cdot,\cdot)$, the deviation subsystem converges to $\mathbf{0}$, implying that the agents reach a perfect consensus from any initial condition in $\mathscr{A}^\perp$.

In the sequel, we use the results of Lemmas 3.5.1 and 3.5.2 to show how Theorem 2.5.1 can be applied to establish the SPAS of $X^* \times \{\mathbf{0}\}$ for the feedback interconnected system (3.25)-(3.29) shown in Figure 3.1.

3.6 Composite Lyapunov Analysis

In the prequel, the functions $V_{\mathscr{O}}(y) = N\|y - \mathbf{P}_{X^*}(y)\|^2$ and $V_{\mathscr{C}}(z) = \|z\|^2$ are shown to be Lyapunov functions for the idealized mean and deviation subsystems, respectively. In the next two lemmas, we examine how the values of these functions evolve along the trajectories of the actual dynamics (3.31)–(3.37). In Lemma 3.6.3, we combine these results to derive a difference inequality bounding the evolution of a composite Lyapunov function $V(\cdot,\cdot)$, comprised as the sum of $V_{\mathscr{O}}(\cdot)$ and $V_{\mathscr{C}}(\cdot)$.

Lemma 3.6.3 constitutes the main result in this section, its proof highlighting the utility of the interconnected systems point of view. The interesting outcome of this lemma is that the projection-related terms arising in the upper bound on $\Delta V_{\mathscr{O}}(\cdot)$ can be shown to negate those arising in the upper bound on $\Delta V_{\mathscr{C}}(\cdot)$. In other words, the projection-related perturbations to the idealized mean dynamics help to stabilize the deviation subsystem, and vice versa. The ultimate implication of this observation, is that the presence of projections in the CO algorithm (3.5) need not add any conservatism to its convergence conditions.

Before presenting Lemma 3.6.3, we examine the evolution of $V_{\mathscr{O}}(\cdot)$ along the sequences generated by (3.25) (q.v. Lemma 3.6.1), and the evolution of $V_{\mathscr{C}}(\cdot)$ along the sequences generated by (3.29) (q.v. Lemma 3.6.2).

Lemma 3.6.1: Consider the function $V_{\mathscr{O}}(y) = N\|y - \mathbf{P}_{X^*}(y)\|^2$ and the system (3.31). For all y generated by (3.31),

$$\Delta V_{\mathscr{O}}(y) \leq -2\alpha(y - \mathbf{P}_{X^*}(y))^T s_o(y) + \tau_1 + \tau_2 + \tau_3,$$

where

$$\begin{aligned}
\tau_1 &= 2\alpha(y - \mathbf{P}_{X^*}(y))^T(\mathbf{1}_N^T \otimes I_n)\big(s(\mathbf{1}_N \otimes y) - s(v)\big), \\
\tau_2 &= -2(v - \mathbf{1}_N \otimes \mathbf{P}_{X^*}(y))^T(M \otimes I_n)\eta + 2(v - \mathbf{1}_N \otimes \mathbf{P}_{X^*}(y))^T\eta, \\
\tau_3 &= \|\eta\|^2 + \alpha^2\|s(v)\|^2 + 2\alpha\eta^T(M \otimes I_n)s(v) - 2\alpha\eta^T s(v),
\end{aligned} \tag{3.41}$$

and $\Delta V_{\mathscr{O}}(y) = V_{\mathscr{O}}(y^+) - V_{\mathscr{O}}(y)$.

Proof. For brevity, we drop the arguments for $p_1(\cdot,\cdot)$ and $p_2(\cdot,\cdot)$. The definition of the projection operator and the convexity of X^* give that

$$\begin{aligned}
\Delta V_{\mathscr{O}}(y) &= N\|y^+ - \mathbf{P}_{X^*}(y^+)\|^2 - N\|y - \mathbf{P}_{X^*}(y)\|^2 \\
&\leq N\|y^+ - \mathbf{P}_{X^*}(y)\|^2 - N\|y - \mathbf{P}_{X^*}(y)\|^2.
\end{aligned}$$

Then, based on (3.31),

$$\begin{aligned}\Delta V_{\mathscr{O}}(y) \leq{}& N\|y-\tfrac{\alpha}{N}s_o(y)+p_1+p_2-\mathbf{P}_{X^*}(y)\|^2\\ &-N\|y-\mathbf{P}_{X^*}(y)\|^2\\ ={}& N\|p_1+p_2-\tfrac{\alpha}{N}s_o(y)\|^2+2N(y-\mathbf{P}_{X^*}(y))^T(p_1+p_2)\\ &-2\alpha(y-\mathbf{P}_{X^*}(y))^T s_o(y).\end{aligned}$$

Recalling (3.32), we see that $2N(y-\mathbf{P}_{X^*}(y))^T p_1 = \tau_1$ Next, we show that $2N(y-\mathbf{P}_{X^*}(y))^T p_2 = \tau_2$. We have

$$\begin{aligned}N(y-\mathbf{P}_{X^*}(y))^T p_2 &= (y-\mathbf{P}_{X^*}(y))^T(\mathbf{1}_N^T\otimes I_n)\eta\\ &= \left[\tfrac{1}{N}(\mathbf{1}_N^T\otimes I_n)(v-\mathbf{1}_N\otimes\mathbf{P}_{X^*}(y))\right]^T(\mathbf{1}_N^T\otimes I_n)\eta,\end{aligned}\tag{3.42}$$

where the second equality is obtained using the fact that $\frac{1}{N}(\mathbf{1}_N^T\otimes I_n)(\mathbf{1}_N\otimes\mathbf{P}_{X^*}(y)) = \mathbf{P}_{X^*}(y)$ and the fact that

$$\tfrac{1}{N}(\mathbf{1}_N^T\otimes I_n)v = \tfrac{1}{N}(\mathbf{1}_N^T\otimes I_n)(A\otimes I_n)x = \tfrac{1}{N}(\mathbf{1}_N^T\otimes I_n)x = y,$$

which follows from the definition of v in (3.5), the definition of y in (3.23) and the fact that A is column-stochastic. Rearranging (3.42) and using the definition of M in (3.27), we obtain

$$\begin{aligned}N(y-\mathbf{P}_{X^*}(y))^T p_2 &= (v-\mathbf{1}_N\otimes\mathbf{P}_{X^*}(y))^T\left(\tfrac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\otimes I_n\right)\eta\\ &= (v-\mathbf{1}_N\otimes\mathbf{P}_{X^*}(y))^T\left((-M+I_N)\otimes I_n\right)\eta,\end{aligned}$$

from which it is seen that $2N(y-\mathbf{P}_{X^*}(y))^T p_2 = \tau_2$.

Finally, we show that $N\|p_1+p_2-\frac{\alpha}{N}s_o(y)\|^2 = \tau_3$. Recalling (3.32) and (3.33), we have

$$\begin{aligned}&N\|p_1+p_2-\tfrac{\alpha}{N}s_o(y)\|^2\\ &\quad= N\|\tfrac{1}{N}(\mathbf{1}_N^T\otimes I_n)[\eta-\alpha s(v)]\|^2\\ &\quad= N\|\tfrac{1}{N}(\mathbf{1}_N^T\otimes I_n)\eta\|^2+\alpha^2 N\|\tfrac{1}{N}(\mathbf{1}_N^T\otimes I_n)s(v)\|^2\\ &\qquad\quad -2\tfrac{\alpha}{N}\left((\mathbf{1}_N^T\otimes I_n)\eta\right)^T\left((\mathbf{1}_N^T\otimes I_n)s(v)\right)\\ &\quad= N\|\tfrac{1}{N}\sum_{i\in\mathscr{V}}\eta_i\|^2+\alpha^2 N\|\tfrac{1}{N}\sum_{i\in\mathscr{V}}s_i(v_i)\|^2\\ &\qquad\quad -2\alpha\eta^T(\tfrac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\otimes I_n)s(v).\end{aligned}\tag{3.43}$$

Recognizing that $\|\cdot\|$ is a convex function, we apply Jensen's inequality to the first two terms in (3.43) to obtain

$$\begin{aligned}&N\|p_1+p_2-\tfrac{\alpha}{N}s_o(y)\|^2\\ &\quad= \|\eta\|^2+\alpha^2\|s(v)\|^2-2\alpha\eta^T(\tfrac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\otimes I_n)s(v).\end{aligned}\tag{3.44}$$

Using the fact that $\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T = -M+I_N$ (q.v. (3.27)), we see that the right-hand side of (3.44) is equal to τ_3. □

Lemma 3.6.2: Consider the function $V_{\mathscr{C}}(z) = \|z\|^2$ and the system (3.37), where A satisfies A3.3.1. For all z generated by (3.37),

$$\Delta V_{\mathscr{C}}(z) \leq -(1-\mu)\|z\|^2+\tau_4+\tau_5,$$

where

$$\tau_4 = 2z^T(A\otimes I_n)\eta - 2\alpha z^T(A\otimes I_n)s(v),$$
$$\tau_5 = \|\eta\|^2 + \alpha^2\|s(v)\|^2 - 2\alpha\eta^T(M\otimes I_n)s(v),$$

and $\Delta V_{\mathscr{C}}(z) = V_{\mathscr{C}}(z^+) - V_{\mathscr{C}}(z)$.

Proof. Based on (3.37) and A3.3.1,

$$\begin{aligned}\Delta V_{\mathscr{C}}(z) &= \|(A\otimes I_n)z + p_3\|^2 - \|z\|^2\\ &= \|(A\otimes I_n)z\|^2 - \|z\|^2 + 2z^T(A\otimes I_n)p_3 + \|p_3\|^2\\ &\leq -(1-\mu)\|z\|^2 + 2z^T(A\otimes I_n)p_3 + \|p_3\|^2.\end{aligned}$$

Recalling (3.38), we have

$$2z^T(A\otimes I_n)p_3 = 2z^T(A\otimes I_n)(M\otimes I_n)(\eta - \alpha s(v)). \tag{3.45}$$

From the definition of M in (3.27) and the properties of A, we see $MA = AM$, and therefore $(A\otimes I_n)(M\otimes I_n) = (M\otimes I_n)(A\otimes I_n)$. Moreover, since $M^2 = M$ and $z^T = x^T(M\otimes I_n)$ (q.v. (3.26)), we have that $z^T(M\otimes I_n) = z^T$. Using these two facts and considering (3.45) allows us to conclude that $2z^T(A\otimes I_n)p_3 = \tau_4$.

Next, we show that $\|p_3\|^2 \leq \tau_5$. We have

$$\begin{aligned}\|p_3\|^2 = \|(M\otimes I_n)\eta\|^2 &+ \alpha^2\|(M\otimes I_n)s(v)\|^2\\ &- 2\alpha\eta^T(M\otimes I_n)^T(M\otimes I_n)s(v).\end{aligned}$$

It can be shown that the spectral radius of M is $\rho(M) = 1$. Since M is symmetric, $\rho(M)$ coincides with $\|M\|$, and the conclusion follows from the compatibility property of induced matrix norms. □

In the following lemma, we examine the evolution of the composite Lyapunov function candidate $V(y,z) = N\|y - \mathbf{P}_{X^*}(y)\|^2 + \|z\|^2$ for the feedback interconnection of the mean and deviation subsystems, shown in Figure 3.1. We find that the projection-related perturbations to the evolution of the idealized mean dynamics effectively negate the destabilizing effects of analogous perturbations to the idealized deviation dynamics, and vice versa.

Lemma 3.6.3: Consider the interconnected system (3.31)-(3.37), and suppose that all of the conditions of Lemmas 3.6.1 and 3.6.2 are satisfied. Then, the function $V(y,z) = N\|y - \mathbf{P}_{X^*}(y)\|^2 + \|z\|^2$ is such that for all y^+ and z^+ generated by (3.31)-(3.37),

$$\begin{aligned}\Delta V(y,z) \leq -2\alpha(y - \mathbf{P}_{X^*}(y))^T s_o(y) &- (1-\mu)\|z\|^2\\ &+ \tau_1 + 2\alpha^2\|s(v)\|^2 - 2\alpha z^T(A\otimes I_n)s(v),\end{aligned} \tag{3.46}$$

where τ_1 is as in (3.41), and $\Delta V(y,z) = V(y^+,z^+) - V(y,z)$.

Proof. From Lemmas 3.6.1 and 3.6.2 we have that

$$\Delta V(y,z) \leq -2\alpha(y - \mathbf{P}_{X^*}(y))^T s_o(y) - (1-\mu)\|z\|^2 + \sum_{i=1}^{5}\tau_i.$$

Therefore, our task is to show that

$$\sum_{i=2}^{5} \tau_i \leq 2\alpha^2 \|s(v)\|^2 - 2\alpha z^T (A \otimes I_n) s(v), \tag{3.47}$$

where the expressions for τ_2, τ_3, τ_4 and τ_5 are given in the statements of Lemmas 3.6.1 and 3.6.2.

First, we observe the cancellation of cross term involving η and $s(v)$ in τ_3 and τ_5, yielding

$$\tau_3 + \tau_5 = 2\|\eta\|^2 + 2\alpha^2 \|s(v)\|^2 - 2\alpha s(v)^T \eta.$$

Adding τ_2 we obtain

$$\begin{aligned}
\tau_2 + \tau_3 + \tau_5 &= 2\|\eta\|^2 + 2\alpha^2 \|s(v)\|^2 - 2\alpha s(v)^T \eta \\
&\quad - 2(v - \mathbf{1}_N \otimes \mathbf{P}_{X^*}(y))^T (M \otimes I_n)\eta \\
&\quad + 2(v - \mathbf{1}_N \otimes \mathbf{P}_{X^*}(y))^T \eta \\
&= 2\|\eta\|^2 + 2\alpha^2 \|s(v)\|^2 \\
&\quad - 2(v - \mathbf{1}_N \otimes \mathbf{P}_{X^*}(y))^T (M \otimes I_n)\eta \\
&\quad + 2\big((v - \alpha s(v)) - \mathbf{1}_N \otimes \mathbf{P}_{X^*}(y)\big)^T \eta.
\end{aligned} \tag{3.48}$$

Recalling the definitions of η, v and $s(v)$ in (3.21), we write the last term in (3.48) as

$$\begin{aligned}
&2\big((v - \alpha s(v)) - \mathbf{1}_N \otimes \mathbf{P}_{X^*}(y)\big)^T \eta \\
&\quad = 2 \sum_{i \in \mathscr{V}} \big((v_i - \alpha s_i(v_i)) - \mathbf{P}_{X^*}(y)\big)^T \eta_i,
\end{aligned} \tag{3.49}$$

where η_i is as in (3.22). Since $X^* \subseteq X_i$, $\forall i \in \mathscr{V}$ (q.v. (3.2)), we see that for all $y \in \mathbb{R}^n$, $\mathbf{P}_{X^*}(y) \in X_i$, for every $i \in \mathscr{V}$. Consequently, we may apply Lemma 2.4.1 to obtain that

$$\begin{aligned}
&2\big((v - \alpha s(v)) - \mathbf{1}_N \otimes \mathbf{P}_{X^*}(y)\big)^T \eta \\
&\quad \leq -2 \sum_{i \in \mathscr{V}} \|\eta_i\|^2 = -2\|\eta\|^2,
\end{aligned} \tag{3.50}$$

and therefore

$$\tau_2 + \tau_3 + \tau_5 \leq 2\alpha^2 \|s(v)\|^2 - 2(v - \mathbf{1}_N \otimes \mathbf{P}_{X^*}(y))^T (M \otimes I_n)\eta.$$

Next, we note that the second term above can be written as

$$\begin{aligned}
&2(v - \mathbf{1}_N \otimes \mathbf{P}_{X^*}(y))^T (M \otimes I_n)\eta \\
&\quad = 2\big[(M \otimes I_n)v - (M \otimes I_n)(\mathbf{1}_N \otimes \mathbf{P}_{X^*}(y))\big]^T \eta \\
&\quad = 2((M \otimes I_n)v)^T \eta,
\end{aligned}$$

since $M\mathbf{1}_N = \mathbf{0}$. Then, recalling that $v = (A \otimes I_n)x$ and that $MA = AM$, we have that

$$\begin{aligned} 2(v - \mathbf{1}_N \otimes \mathbf{P}_{X^*}(y))^T (M \otimes I_n)\eta &= 2((A \otimes I_n)(M \otimes I_n)x)^T \eta \\ &= 2((A \otimes I_n)z)^T \eta, \end{aligned}$$

and therefore

$$\tau_2 + \tau_3 + \tau_5 \leq 2\alpha^2 \|s(v)\|^2 - 2z^T (A \otimes I_n)\eta.$$

Finally, we observe that by adding τ_4, the terms involving the projection errors η cancel, resulting in the desired inequality (3.47). □

The conclusions of Lemma 3.6.3 are independent of whether the search directions $s_i(\cdot)$ are locally bounded or locally Lipschitz continuous. In the following two subsections, we study these two cases separately. We apply Theorem 2.5.1 to derive conditions on the algorithm gain α which guarantee the SPAS of $X^* \times \{\mathbf{0}\}$ for the feedback interconnected system (3.31)-(3.37) in each case.

Since projection-related terms do not appear in the upper bound on $\Delta V(\cdot)$ derived in Lemma 3.6.3, the projection operation plays no role in the derivation of the final convergence condition.

3.6.1 Locally Lipschitz Search Directions

In the following lemma we refine the form for $\Delta V(\cdot,\cdot)$ derived in Lemma 3.6.3 by assuming the local Lipschitz continuity of the search directions $s_i(\cdot)$ and the strict pseudogradient property of $s_o(\cdot)$.

Lemma 3.6.4: Consider the interconnected system (3.31)-(3.37) and the composite Lyapunov function candidate $V(y,z) = N\|y - \mathbf{P}_{X^*}(y)\|^2 + \|z\|^2$. Suppose that A3.3.2 and A3.3.4 are satisfied in addition to the conditions of Lemma 3.6.3. Then, given an arbitrarily large compact set $\Omega \subset \mathbb{R}^n$ containing X^*, it holds that for all $(y^T, z^T)^T \in (\Omega \cap \hat{X}) \times \mathscr{A}^\perp$, with $\mathscr{A}^\perp$ as in (3.40),

$$\Delta V(y,z) \leq -\alpha\phi(y) + \alpha^2 K_y \|y - \mathbf{P}_{X^*}(y)\|^2 - K_z\|z\|^2 + \alpha^2 K, \tag{3.51}$$

where y^+ and z^+ are generated by (3.31)-(3.37),

$$K_y = \frac{4L^2N\mu}{1-\mu} + (8L^2N)\frac{1+\mu}{1-\mu}, \tag{3.52}$$

$$K_z = \tfrac{1}{2}(1-\mu) - \alpha^2(4L^2\mu)\frac{1+\mu}{1-\mu}, \tag{3.53}$$

$$K = 8\frac{1+\mu}{1-\mu}s^*, \tag{3.54}$$

$L = \max_i L_i$, and

$$s^* = \max_{x^* \in X^*} \|s(\mathbf{1}_N \otimes x^*)\|^2. \tag{3.55}$$

Proof. Since $x_i \in X_i$ for each $t \in \mathbb{N}$, from the definition of y in (3.23) we observe that $y \in \hat{X}$ for each $t \in \mathbb{N}$, and we may therefore apply A3.3.2 to the first term in (3.46) to obtain

$$\begin{aligned}\Delta V(y,z) \leq &-\alpha\phi(y) - (1-\mu)\|z\|^2 \\ &+ \tau_1 + 2\alpha^2\|s(v)\|^2 - 2\alpha z^T(A\otimes I_n)s(v).\end{aligned} \tag{3.56}$$

Recalling the expression for τ_1 in (3.41) and applying the Cauchy-Schwarz inequality, we write

$$\begin{aligned}\tau_1 &= 2\alpha(\mathbf{1}_N \otimes y - \mathbf{1}_N \otimes \mathbf{P}_{X^*}(y))^T \big(s(\mathbf{1}_N \otimes y) - s(v)\big) \\ &\leq 2\alpha\sqrt{N}\|y - \mathbf{P}_{X^*}(y)\|\|s(\mathbf{1}_N\otimes y) - s(v)\|,\end{aligned}$$

and thus obtain that

$$\begin{aligned}\Delta V(y,z) \leq &-\alpha\phi(y) - (1-\mu)\|z\|^2 \\ &+ 2\alpha^2\|s(v)\|^2 - 2\alpha z^T(A\otimes I_n)s(v) \\ &+ 2\alpha\sqrt{N}\|y - \mathbf{P}_{X^*}(y)\|\|s(\mathbf{1}_N\otimes y) - s(v)\|.\end{aligned} \tag{3.57}$$

Applying Young's inequality to the last two terms in (3.57), we obtain

$$\begin{aligned}\Delta V(y,z) \leq &-\alpha\phi(y) - (1-\mu)\|z\|^2 + 2\alpha^2\|s(v)\|^2 \\ &\tfrac{\alpha^2}{\varepsilon_1}\|s(v)\|^2 + \varepsilon_1\|(A\otimes I_n)z\|^2 \\ &+ \tfrac{\alpha^2 N}{\varepsilon_2}\|y - \mathbf{P}_{X^*}(y)\|^2 + \varepsilon_2\|s(\mathbf{1}_N\otimes y) - s(v))\|^2 \\ \leq &-\alpha\phi(y) - (1-\mu-\varepsilon_1\mu)\|z\|^2 + \alpha^2\big(2+\tfrac{1}{\varepsilon_1}\big)\|s(v)\|^2 \\ &+ \tfrac{\alpha^2 N}{\varepsilon_2}\|y - \mathbf{P}_{X^*}(y)\|^2 + \varepsilon_2\|s(\mathbf{1}_N\otimes y) - s(v))\|^2,\end{aligned}$$

where ε_1 and ε_2 are some positive, real numbers whose value we select later.

We bound the last term above by applying assumption A3.3.4 and noting that $v - \mathbf{1}_N \otimes y = (A\otimes I_n)z$ (q.v. (3.30)) to obtain

$$\begin{aligned}\|s(\mathbf{1}_N\otimes y) - s(v))\|^2 &\leq L^2\|(A\otimes I_n)z\|^2 \\ &\leq L^2\mu\|z\|^2,\end{aligned}$$

where $L = \max_{i\in\mathscr{V}} L_i$ and μ is as in A3.3.1. Consequently,

$$\begin{aligned}\Delta V(y,z) \leq &-\alpha\phi(y) - (1-\mu-\varepsilon_1\mu-\varepsilon_2 L^2\mu)\|z\|^2 \\ &+ \alpha^2\big(2+\tfrac{1}{\varepsilon_1}\big)\|s(v)\|^2 + \tfrac{\alpha^2 N}{\varepsilon_2}\|y - \mathbf{P}_{X^*}(y)\|^2,\end{aligned}$$

Next, using the that fact that $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$ for any two real vectors a and b, we write

$$\begin{aligned}
\|s(v)\|^2 &= \|s(v) - s(\mathbf{1}_N \otimes y) + s(\mathbf{1}_N \otimes y)\|^2 \\
&\leq 2\|s(v) - s(\mathbf{1}_N \otimes y)\|^2 \\
&\quad + 2\|s(\mathbf{1}_N \otimes y) - s(\mathbf{1}_N \otimes \mathbf{P}_{X^*}(y)) + s(\mathbf{1}_N \otimes \mathbf{P}_{X^*}(y))\|^2 \\
&\leq 2L^2\mu\|z\|^2 + 4L^2\|\mathbf{1}_N \otimes y - \mathbf{1}_N \otimes \mathbf{P}_{X^*}(y)\|^2 \\
&\quad + 4\|s(\mathbf{1}_N \otimes \mathbf{P}_{X^*}(y))\|^2 \\
&\leq 2L^2\mu\|z\|^2 + 4L^2N\|y - \mathbf{P}_{X^*}(y)\|^2 + 4s^*
\end{aligned}$$

where s^* is well defined (as in (3.55)) since X^* is compact and each $s_i(\cdot)$ is continuous by virtue of A3.3.4. We therefore have that $\Delta V(y,z)$ satisfies (3.51), with

$$K_y = \frac{N}{\varepsilon_2} + \big(2 + \frac{1}{\varepsilon_1}\big)(4L^2N) \tag{3.58}$$

$$K_z = \big((1-\mu) - \varepsilon_1\mu - \varepsilon_2 L^2\mu - \alpha^2\big(2 + \frac{1}{\varepsilon_1}\big)(2L^2)\mu\big) \tag{3.59}$$

$$K = 4\big(2 + \frac{1}{\varepsilon_1}\big)s^*. \tag{3.60}$$

It can then be verified that choosing $\varepsilon_1 = \frac{1-\mu}{4\mu}$ and $\varepsilon_2 = \frac{1-\mu}{4L^2\mu}$ renders expressions (3.58), (3.59) and (3.60) equal to those in the statement of the lemma. □

Our main result concerning locally Lipschitz search directions is given in the following theorem.

Theorem 3.6.1: Suppose that A3.3.1, A3.3.2, A3.3.3 and A3.3.4 are satisfied. Then, the set $X^* \times \{\mathbf{0}\}$ is SPAS for the interconnected system (3.31)-(3.37).

Proof. We apply Theorem 2.5.1, with $\Gamma_0 = X^* \times \{\mathbf{0}\}$ and $\Xi = \hat{X} \times \mathscr{A}^\perp$. The function $V(y,z) = N\|y - \mathbf{P}_{X^*}(y)\|^2 + \|z\|^2$ is continuous, radially unbounded and positive definite with respect to the set Γ_0 on $\mathbb{R}^{(N+1)n}$.

Let ξ denote the vector $(y^T, z^T)^T \in \mathbb{R}^{(N+1)n}$. Let σ be an arbitrary, positive, real number, and take $\Omega = \bar{B}^n_\sigma(X^*)$. Then, under the conditions in Lemma 3.6.4, we have that for all $\xi \in (\bar{B}^n_\sigma(X^*) \cap \hat{X}) \times \mathscr{A}^\perp$, with $\mathscr{A}^\perp$ as in (3.40),

$$\Delta V(y,z) \leq W(y,z;\alpha), \tag{3.61}$$

where

$$W(y,z;\alpha) = -\alpha\phi(y) + \alpha^2 K_y\|y - \mathbf{P}_{X^*}(y)\|^2 - K_z\|z\|^2 + \alpha^2 K, \tag{3.62}$$

y^+ and z^+ are generated by (3.31)-(3.37), and K_y, K_z and K are as in (3.52), (3.53) and (3.54).

Since

$$\begin{aligned}
(\bar{B}^n_\sigma(X^*) \cap \hat{X}) \times \mathscr{A}^\perp &= (\bar{B}^n_\sigma(X^*) \cap \hat{X}) \times (\mathbb{R}^{Nn} \cap \mathscr{A}^\perp) \\
&= (\bar{B}^n_\sigma(X^*) \times \mathbb{R}^{Nn}) \cap (\hat{X} \times \mathscr{A}^\perp) \\
&= (\bar{B}^n_\sigma(X^*) \times \mathbb{R}^{Nn}) \cap \Xi \\
&\supset \bar{B}^{(N+1)n}_\sigma(\Gamma_0) \cap \Xi,
\end{aligned} \tag{3.63}$$

we see that (2.32) holds on the required set.

It remains to show that for the given number σ, there exists a positive, real number $\bar{\alpha}$, such that whenever $\alpha \in (0,\bar{\alpha})$, $W(\cdot,\cdot;\alpha)$ has properties P1, P2 and P3 specified in the statement of Theorem 2.5.1.

Let $\delta \in (0,\sigma)$ be arbitrary as in P3, and let $\hat{\delta} = \sqrt{\delta/2}$. Since $\phi(\cdot)$ in (3.62) is radially unbounded and positive definite with respect to X^* on $\hat{X}$, Lemma 2.5.2 implies that for any $K_\phi \in \mathbb{R}_{++}$, there exists an $\alpha_\phi \in \mathbb{R}_{++}$ such that

$$W(y,z;\alpha) \leq -\alpha^2 \big(K_\phi - K_y\big)\|y - \mathbf{P}_{X^*}(y)\|^2 - K_z\|z\|^2 + \alpha^2 K,$$

for all $\xi \in \big(\bar{B}^n_\sigma(X^*)\backslash B^n_{\hat{\delta}}(X^*)\big) \times \mathbb{R}^{Nn}$, provided $\alpha \in (0,\alpha_\phi)$. Choosing

$$K_\phi = \frac{K}{\hat{\delta}^2} + K_y, \tag{3.64}$$

implies that

$$W(y;\alpha) \leq -\alpha^2 \tfrac{K}{\hat{\delta}^2}\|y - \mathbf{P}_{X^*}(y)\|^2 - K_z\|z\|^2 + \alpha^2 K, \tag{3.65}$$

for all $\xi \in \big(\bar{B}^n_\sigma(X^*)\backslash B^n_{\hat{\delta}}(X^*)\big) \times \mathbb{R}^{Nn}$, whenever $\alpha \in (0,\alpha_\phi)$.

From (3.53), we note that $K_z > 0$ whenever $\alpha \in (0,\alpha_{K_z})$, where

$$\alpha_{K_z} = \frac{1-\mu}{2L\sqrt{2\mu(1+\mu)}}. \tag{3.66}$$

Consequently, $W(y,z;\alpha) < 0$ for all $\xi \in \big(\bar{B}^n_\sigma(X^*)\backslash \bar{B}^n_{\hat{\delta}}(X^*)\big) \times \mathbb{R}^{Nn}$, provided that $\alpha \in (0,\bar{\alpha})$, where

$$\bar{\alpha} = \min\{\alpha_\phi, \alpha_{K_z}\}. \tag{3.67}$$

On the other hand, from (3.62) we observe that $W(\cdot,\cdot;\alpha)$ can be no larger than $\alpha^2(K_y\hat{\delta}^2 + K)$ on the set $\bar{B}_{\hat{\delta}}(X^*) \times \mathbb{R}^{Nn}$. Therefore,

$$W(y;\alpha) \leq \alpha^2(\tfrac{K_y\delta}{2} + K),$$

for all $\xi \in \bar{B}^n_\sigma(X^*) \times \mathbb{R}^{Nn} \supset \bar{B}^{(N+1)n}_\sigma(\Gamma_0) \cap \Xi$, and P1 is satisfied with $b_W(\alpha) = \alpha^2(\tfrac{K_y\delta}{2} + K)$.

Recall that $Z(\alpha)$ denotes the set of all ξ in Ξ for which $W(y,z;\alpha)$ is non-negative. Since $\phi(\cdot)$ is positive definite with respect to X^*, and $\|\cdot\|$ with respect to $\{\mathbf{0}\}$, we see from (3.62) that $Z(\alpha) \supseteq \Gamma_0$, and that P2 is thereby satisfied.

Next, from the observation that $W(y,z;\alpha) < 0$ on $\xi \in \big(\bar{B}^n_\sigma(X^*)\backslash \bar{B}^n_{\hat{\delta}}(X^*)\big) \times \mathbb{R}^{Nn}$, we see that $Z(\alpha) \subseteq \bar{B}_{\hat{\delta}}(X^*) \times \mathbb{R}^{Nn}$, whenever $\alpha \in (0,\bar{\alpha})$. On the other hand, from (3.62) it is evident that

$$\begin{aligned} Z(\alpha) &= \big\{\xi \in \Xi \,:\, \alpha\phi(y) - \alpha^2 K_y\|y - \mathbf{P}_{X^*}(y)\|^2 \\ &\qquad + K_z\|z\|^2 \leq \alpha^2 K\big\} \\ &\subseteq S_y \times S_z, \end{aligned}$$

where

$$S_y = \{y \in \hat{X} \,:\, \phi(y) \leq \alpha K + \alpha K_y\|y - \mathbf{P}_{X^*}(y)\|^2\}, \tag{3.68}$$

and

$$S_z = \{z \in \mathscr{A}^\perp \,:\, \|z\|^2 \leq \alpha^2 \tfrac{K}{K_z}\}. \tag{3.69}$$

Clearly, $S_z \subseteq \bar{B}^{Nn}_{\hat{\delta}}(\{\mathbf{0}\})$, provided that $\alpha \in (0, \sqrt{\frac{\delta K_z}{2K}})$. Therefore, taking $\alpha \in (0, \alpha_Z)$, where

$$\alpha_Z = \min\{\bar{\alpha}, \sqrt{\tfrac{\delta K_z}{2K}}\}, \tag{3.70}$$

implies that $Z(\alpha)$ is a subset of both $\bar{B}_{\hat{\delta}}(X^*) \times \mathbb{R}^{Nn}$ and $S_y \times \bar{B}^{Nn}_{\hat{\delta}}(\{\mathbf{0}\})$. As such, it must also be a subset of the intersection of these two sets. Consequently,

$$\begin{aligned} Z(\alpha) &\subseteq (\bar{B}_{\hat{\delta}}(X^*) \times \mathbb{R}^{Nn}) \cap (S_y \times \bar{B}^{Nn}_{\hat{\delta}}(\{\mathbf{0}\})) \\ &= (\bar{B}_{\hat{\delta}}(X^*) \cap S_y) \times (\mathbb{R}^{Nn} \cap \bar{B}^{Nn}_{\hat{\delta}}(\{\mathbf{0}\})) \\ &\subseteq \bar{B}_{\hat{\delta}}(X^*) \times \bar{B}^{Nn}_{\hat{\delta}}(\{\mathbf{0}\}) \\ &\subset \bar{B}^{(N+1)n}_{\delta}(\Gamma_0), \end{aligned}$$

showing that P3 is satisfied with α_Z as in (3.70).

Having satisfied all of the conditions of Theorem 2.5.1, with $\bar{\alpha}$ as in (3.67), we conclude that under the conditions of Lemma 3.6.4, the set $X^* \times \{\mathbf{0}\}$ is SPAS for the feedback interconnection (3.31)-(3.37). $\square$

Remark 3.6.1: The so-called "small gain" conditions ensuring the SPAS of the feedback interconnection (3.25)–(3.29) shown in Figure 3.1 are given in (3.70) and (3.67). These conditions effectively ensure that value of the composite Lyapunov function $V(\cdot,\cdot)$ decreases at each iteration, until the mean and deviation variables come sufficiently close to the set $X^* \times \{\mathbf{0}\}$. $\diamondsuit$

We note that in §3.8, the theory of this section is applied by showing how the conditions of Theorem 3.6.1 may be verified for a general class of consensus optimization algorithms which are based on weighted gradient schemes.

In the sequel, we derive similar results for the case in which the agents' individual search directions $s_i(\cdot)$ are locally bounded, but not necessarily locally Lipschitz.

3.6.2 Locally Bounded Search Directions

It is straightforward to show that the strict pseudogradient assumption A3.3.2 implies that X^* is SPAS for the idealized mean dynamics (3.36) when $s_o(\cdot)$ is either locally bounded (but not locally Lipschitz) *or* locally Lipschitz (q.v. Lemma 3.5.1). However, in the absence of the local Lipschitz continuity of the search directions $s_i(\cdot)$, certain terms arising in the upper bound on $\Delta V(\cdot,\cdot)$ hinder the use of A3.3.2 in drawing the same conclusions about $X^* \times \{\mathbf{0}\}$ for the interconnection (3.31)-(3.37). For this reason, we use the stronger assumption A3.3.2$'$ for our analysis here.

We return to the results of Lemma 3.6.3. In analogy to Lemma 3.6.4, in the following lemma we refine the expression for $\Delta V(\cdot,\cdot)$ derived in Lemma 3.6.3 for the case in which the search directions are locally bounded and satisfy A3.3.2$'$. We note that the techniques surrounding the application of A3.3.2$'$ in the proof are similar to those used in the proof of Proposition 4 in [116].

Lemma 3.6.5: Consider the interconnected system (3.31)-(3.37) and the composite Lyapunov function candidate $V(y,z) = N\|y - \mathbf{P}_{X^*}(y)\|^2 + \|z\|^2$. Suppose that A3.3.2$'$ and A3.3.4$'$ are satisfied in addition to the conditions of Lemma 3.6.3. Then, given an arbitrarily large compact set $\Omega \subset \mathbb{R}^n$ containing X^*, it holds that for all $(y^T, z^T)^T \in$

$\Omega \cap \hat{X} \times \mathscr{A}^\perp$, with $\mathscr{A}^\perp$ as in (3.40),

$$\Delta V(y,z) \leq -\alpha\phi(y) - K_z\|z\|^2 + \alpha^2 K, \tag{3.71}$$

where y^+ and z^+ are generated by (3.31)-(3.37),

$$K_z = \tfrac{1}{2}(1-\mu), \tag{3.72}$$
$$K = 2NB^2 + \tfrac{\mu}{1-\mu}NL_\phi^2, \tag{3.73}$$

and L_ϕ denotes the largest of the Lipschitz constants associated to the functions $\phi_i(\cdot)$ over the set $\Omega \cap \hat{X}$.

Proof. We begin with expression (3.46) for the upper bound on $\Delta V(y,z)$. Recalling the expression for τ_1 in (3.41), we note that

$$\begin{aligned}
-2\alpha(y-\mathbf{P}_{X^*}(y))^T s_o(y) + \tau_1 &= -2\alpha(y-\mathbf{P}_{X^*}(y))^T(\mathbf{1}_N^T \otimes I_n)s(v) \\
&= -2\alpha\left[(\mathbf{1}_N \otimes I_n)y - v + v - (\mathbf{1}_N \otimes I_n)\mathbf{P}_{X^*}(y)\right]^T s(v) \\
&= 2\alpha z^T(A\otimes I_n)s(v) - 2\alpha\sum_{i\in\mathscr{V}}(v_i - \mathbf{P}_{X^*}(y))^T s_i(v_i),
\end{aligned}$$

where the first term is obtained from (3.30). Combining this expression with (3.46) from Lemma 3.6.3 gives

$$\begin{aligned}
\Delta V(y,z) \leq{}& -2\alpha\sum_{i\in\mathscr{V}}(v_i-\mathbf{P}_{X^*}(y))^T s_i(v_i) \\
& - (1-\mu)\|z\|^2 + 2\alpha^2\|s(v)\|^2.
\end{aligned} \tag{3.74}$$

Since $x_i \in X_i$ for each $t\in\mathbb{N}$ and v_i is a convex combination of $\{x_1,\ldots,x_N\}$, we see that $v_i \in \hat{X}$ for each $t \in \mathbb{N}$, and we may therefore apply A3.3.2$'$ to the first term in the above inequality to obtain

$$\begin{aligned}
-2\alpha\sum_{i\in\mathscr{V}}(v_i-\mathbf{P}_{X^*}(y))^T s_i(v_i) &\leq -\alpha\sum_{i\in\mathscr{V}}\phi_i(v_i) \\
&\leq -\alpha\phi(y) + \alpha\sum_{i\in\mathscr{V}}\big(\phi_i(y)-\phi_i(v_i)\big) \\
&\leq -\alpha\phi(y) + \alpha L_\phi\sum_{i\in\mathscr{V}}\|y-v_i\| \\
&\leq -\alpha\phi(y) + \alpha L_\phi\sqrt{N}\|(A\otimes I_n)z\|,
\end{aligned} \tag{3.75}$$

where the third inequality is obtained by invoking the Lipschitz continuity of the functions $\phi_i(\cdot)$, and the last inequality by means of (3.30) and the equivalence of norms. Using the above inequality, we express (3.74) as

$$\begin{aligned}
\Delta V(y,z) \leq{}& -\alpha\phi(y) - (1-\mu)\|z\|^2 + 2\alpha^2\|s(v)\|^2 \\
& + \alpha L_\phi\sqrt{N}\|(A\otimes I_n)z\|.
\end{aligned} \tag{3.76}$$

Applying Young's inequality to the last term gives

$$\alpha L_\phi\sqrt{N}\|(A\otimes I_n)z\| \leq \varepsilon_1\|(A\otimes I_n)z\|^2 + \alpha^2\tfrac{L_\phi^2 N}{2\varepsilon_1} \tag{3.77}$$

where ε_1 is some positive, real number whose value we select next. Using the above inequality and applying

A3.3.1 and A3.3.4′ to (3.76), we obtain

$$\Delta V(y,z) \leq -\alpha\phi(y) - (1-\mu-\varepsilon_1\mu)\|z\|^2 + \alpha^2\Big(2NB^2 + \tfrac{NL_\phi^2}{2\varepsilon_1}\Big), \tag{3.78}$$

from which it can be verified that selecting $\varepsilon_1 = \frac{1-\mu}{2\mu}$ yields the desired expression (3.71). □

Our main result concerning CO algorithms employing bounded search directions is given in the following theorem.

Theorem 3.6.2: Suppose that A3.3.1, A3.3.2′, A3.3.3 and A3.3.4′ are satisfied. Then, the set $X^* \times \{\mathbf{0}\}$ is SPAS for the interconnected system (3.31)-(3.37).

Proof. We apply Theorem 2.5.1, with $\Gamma_0 = X^* \times \{\mathbf{0}\}$ and $\Xi = \hat{X} \times \mathscr{A}^\perp$. The function $V(y,z) = N\|y - \mathbf{P}_{X^*}(y)\|^2 + \|z\|^2$ is continuous, radially unbounded and positive definite with respect to the set Γ_0 on $\mathbb{R}^{(N+1)n}$.

Let ξ denote the vector $(y^T, z^T)^T \in \mathbb{R}^{(N+1)n}$. Let σ be an arbitrary, positive, real number, and take $\Omega = \bar{B}^n_\sigma(X^*)$.

Then, under the conditions in Lemma 3.6.5, we have that for all $\xi \in (\bar{B}^n_\sigma(X^*) \cap \hat{X}) \times \mathscr{A}^\perp$, with $\mathscr{A}^\perp$ as in (3.40),

$$\Delta V(y,z) \leq W(y,z;\alpha), \tag{3.79}$$

where

$$W(y,z;\alpha) = -\alpha\phi(y) - K_z\|z\|^2 + \alpha^2 K, \tag{3.80}$$

y^+ and z^+ are generated by (3.31)-(3.37), and K_z and K are as in (3.72) and (3.73).

By arguments identical to those leading to expression (3.63), we see that (2.32) holds on the required set. It remains to show that for the given number σ, there exists a positive, real number $\bar{\alpha}$, such that whenever $\alpha \in (0,\bar{\alpha})$, $W(\cdot,\cdot;\alpha)$ has properties P1, P2 and P3 specified in the statement of Theorem 2.5.1.

We note that $W(\cdot,\cdot;\alpha)$ is bounded from above by $b_W(\alpha) = \alpha^2 K$, which diminishes with α, as required by P1. Since $\phi(\cdot)$ is radially unbounded and positive definite with respect to X^*, the set

$$Z(\alpha) = \{\xi \in \Xi \mid -\alpha\phi(y) - K_z\|z\|^2 + \alpha^2 K \geq 0\}$$

contains Γ_0, thereby satisfying P2. To see that P3 is satisfied, let $\delta \in (0,\sigma]$ be arbitrary, and let $\hat{\delta} = \sqrt{\delta/2}$. Note that

$$Z(\alpha) \subseteq S_y \times S_z, \tag{3.81}$$

where

$$S_y = \{y \in \mathbb{R}^n \mid \phi(y) \leq \alpha K\}, \tag{3.82}$$

and

$$S_z = \{z \in \mathbb{R}^{Nn} \mid \|z\|^2 \leq \alpha^2 \tfrac{K}{K_z}\}. \tag{3.83}$$

By Lemma 2.5.1 (with $\hat{\rho} = \hat{\delta}$), there exists a number $c \in \mathbb{R}_{++}$ such that $\Phi_c = \{y \in \mathbb{R}^n \mid \phi(y) \leq c\}$ is strictly contained inside the set $\bar{B}^n_{\hat{\delta}}(X^*)$. Therefore, taking $\alpha \in (0, \frac{c}{K})$ ensures that S_y is contained inside Φ_c, and hence inside $\bar{B}^n_{\hat{\delta}}(X^*)$. On the other hand, from (3.83) we see that $S_z \subseteq \bar{B}^{Nn}_{\hat{\delta}}(\{\mathbf{0}\})$, provided that $\alpha \in (0, \sqrt{K_z\delta/2K})$.

Consequently, whenever $\alpha \in (0, \alpha_Z)$, where

$$\alpha_Z = \min\{\tfrac{c}{K}, \sqrt{\tfrac{K_z\delta}{2K}}\}, \tag{3.84}$$

we have that

$$\begin{aligned} Z(\alpha) &\subseteq S_y \times S_z \\ &\subseteq \bar{B}^n_{\hat{\delta}}(X^*) \times \bar{B}^{Nn}_{\hat{\delta}}(\{\mathbf{0}\}) \\ &\subseteq \bar{B}^{(N+1)n}_{\delta}(\Gamma_0), \end{aligned}$$

which shows that P3 is also satisfied.

Having satisfied all of the conditions of Theorem 2.5.1, with $\bar{\alpha}$ being any positive, real number, we conclude that under the conditions of Lemma 3.6.5, the set $X^* \times \{\mathbf{0}\}$ is SPAS for the feedback interconnection (3.31)-(3.37). □

Remark 3.6.2: In the absence of the local Lipschitz continuity of the search directions $s_i(\cdot)$, it is possible to show that under the strict pseudogradient assumption A3.3.2, $\Delta V(y,z) \le W(y,z)$, where $W(\cdot,\cdot)$ is a function satisfying all properties listed in the statement of Theorem 2.5.1 except the last; namely, the set $Z(\alpha)$ cannot be made to fit inside an arbitrarily small ball centred at $\Gamma_0 = X^* \times \{\mathbf{0}\}$. Consequently, within this framework A3.3.2 allows us only to conclude the semiglobal, practical asymptotic stability of Γ_0, where "practical stability" is taken in the sense of Definition 5.14.1 in [3]. ♢

Having derived the conditions under which $X^* \times \{\mathbf{0}\}$ is SPAS for (3.25)–(3.29), in the sequel we show how the properties of the composite Lyapunov function under those conditions can be exploited to derive accuracy and convergence rate results for the general class of CO algorithms represented by (3.5).

3.7 Convergence Properties of the Agents' Estimation Errors

We return to the equations governing the evolution of the generalized consensus optimization scheme (3.5) in its original coordinates. The ultimate upper bound on agents' collective estimation errors can be derived without explicitly solving any difference equations, which is a benefit inherent in Lyapunov-based analyses. The observation leveraged in deriving these bounds is given in the statement of the following claim:

Claim 3.7.1: Let $V(y,z) = N\|y - \mathbf{P}_{X^*}(y)\|^2 + \|z\|^2$. Then,

$$\|x - \mathbf{P}_{\mathbf{X}^*}(x)\|^2 \le 6V(y,z), \tag{3.85}$$

where $x = [x_1^T, \ldots, x_N^T]^T = z + \mathbf{1}_N \otimes y$, and

$$\mathbf{X}^* = \prod_{i=1}^{N} X^*.$$

Proof. First, note that

$$\mathbf{P}_{\mathbf{X}^*}(x) = \begin{bmatrix} \mathbf{P}_{X^*}(x_1) \\ \vdots \\ \mathbf{P}_{X^*}(x_N) \end{bmatrix},$$

which can be shown using the definition of the projection operator, and the fact that optimizing over a set of variables is equivalent to optimizing first over any subset of those variables, and then optimizing the result over the remaining variables. Consequently,

$$\begin{aligned}\|x-\mathbf{P}_{\mathbf{X}^*}(x)\|^2 &= \sum_{i\in\mathscr{V}} \|x_i-\mathbf{P}_{X^*}(x_i)\|^2 \\ &= \sum_{i\in\mathscr{V}} \|x_i-y+y-\mathbf{P}_{X^*}(y)+\mathbf{P}_{X^*}(y)-\mathbf{P}_{X^*}(x_i)\|^2 \\ &\leq \sum_{i\in\mathscr{V}} \big(4\|x_i-y\|^2+4\|y-\mathbf{P}_{X^*}(y)\|^2 \\ &\qquad +2\|\mathbf{P}_{X^*}(y)-\mathbf{P}_{X^*}(x_i)\|^2\big),\end{aligned}$$

which is obtained by applying Young's inequality twice. Next, using the fact that the projection operator is non-expansive (q.v. Proposition 2.2.1, [14]), we obtain

$$\begin{aligned}\|x-\mathbf{P}_{\mathbf{X}^*}(x)\|^2 &\leq \sum_{i\in\mathscr{V}} \big(6\|x_i-y\|^2+4\|y-\mathbf{P}_{X^*}(y)\|^2\big) \\ &= 6\|z\|^2+4N\|y-\mathbf{P}_{X^*}(y)\|^2,\end{aligned}$$

from which (3.85) follows. $\diamondsuit$

Since $V(\cdot,\cdot)$ in the above claim is the Lyapunov function considered in both Theorem 3.6.2 and Theorem 3.6.1, we have that

$$\limsup_{t\to\infty} \|x(t)-\mathbf{P}_{\mathbf{X}^*}(x(t))\|^2 \leq 6 \lim_{t\to\infty} V(y(t),z(t)) \tag{3.86}$$

under the conditions of either theorem. Moreover, under either set of conditions, the function $V(\cdot,\cdot)$ is such that $V(y^+,z^+) < V(y,z)$, for any $(y^T,z^T)^T \notin Z(\alpha)$, with $Z(\alpha)$ as in (2.33). In other words, $V(\cdot,\cdot)$ decreases monotonically along any sequence generated by (3.31)-(3.37), so long as the sequence remains outside of $Z(\alpha)$. Since $V(\cdot,\cdot)$ is positive definite with respect to X^*, the sequence $(V(y(t),z(t)))_{t=0}^{\infty}$ is bounded from below by zero, and there exists a number $\bar{V} \in \mathbb{R}_+$ such that

$$\lim_{t\to\infty} V(y(t),z(t)) = \bar{V}. \tag{3.87}$$

Intuitively, $\bar{V}$ should be no larger than the maximum value attained by $V(\cdot,\cdot)$ on any compact set which is attractive for (3.31)-(3.37). Since $Z(\alpha)$ can be made to fit inside an arbitrarily small ball centered at $X^*\times\{\mathbf{0}\}$, it is possible to render $\bar{V}$ arbitrarily small, thereby rendering the ultimate upper bound on agent's collective estimates given in (3.86) also arbitrarily small.

We make these statements precise in the following lemma.

Lemma 3.7.1: Consider the consensus optimization algorithm (3.5), and suppose that the conditions of either Theorem 3.6.1 or 3.6.2 hold. Then, for every $x(0)\in\mathbb{R}^{Nn}$ and every $\varepsilon\in\mathbb{R}_{++}$, there exists a number $\alpha^*\in\mathbb{R}_{++}$ such that whenever $\alpha\in(0,\alpha^*)$,

$$\limsup_{t\to\infty} \|x(t)-\mathbf{P}_{\mathbf{X}^*}(x(t))\| \leq \varepsilon, \tag{3.88}$$

where $x(t)$ and $\mathbf{X}^*$ are as in Claim 3.7.1. Moreover, there exists a $T\in\mathbb{N}$ (depending on $x(0)$, ε and α) such that

$$\|x(t)-\mathbf{P}_{\mathbf{X}^*}(x(t))\| \leq 2\sqrt{N}\varepsilon, \tag{3.89}$$

for all $t \geq T$.

Proof. Consider the function

$$V(y,z) = N\|y - \mathbf{P}_{X^*}(y)\|^2 + \|z\|^2, \tag{3.90}$$

where y and z are as in either Theorem 3.6.1 or 3.6.2. We follow the proof of Lemma 2.5.1. Let

$$\rho = \sqrt{\tfrac{2}{3}}\varepsilon, \tag{3.91}$$

and let $\bar{\bar{l}}$ be the minimum value attained by $V(\cdot,\cdot)$ over the set $\partial\bar{B}_{\rho}^{(N+1)n}(\Gamma_0)$, where $\Gamma_0 = X^* \times \mathbf{0}$. For any $r \in \mathbb{R}_{++}$, let Γ_r denote the r-sublevel set of $V(\cdot,\cdot)$ – that is,

$$\Gamma_r = \{\xi \in \mathbb{R}^{(N+1)n} \mid V(\xi) \leq r\}.$$

From (3.90), we see that $\bar{\bar{l}} = \rho^2$, and that $\bar{\bar{l}}$ coincides with $\bar{l}$, the minimum that $V(\cdot,\cdot)$ attains on $\Gamma_{\bar{\bar{l}}} \backslash B_{\rho}(\Gamma_0)$.

According to the proof of Lemma 2.5.1, Γ_l is strictly contained inside $\bar{B}_{\rho}(\Gamma_0)$, for any $l \in (0,\bar{l})$; we choose $l = \frac{1}{2}\bar{l} = \frac{1}{2}\rho^2$, which, with (3.91) gives

$$l = \tfrac{1}{3}\varepsilon^2. \tag{3.92}$$

Let $\delta = \frac{\rho}{2\sqrt{N}}$ be the radius of the largest ball centred at Γ_0 and contained in $\Gamma_{l/2}$, so that the maximum value attained by $V(\cdot,\cdot)$ on $\bar{B}_{\delta}(\Gamma_0)$ is given by

$$V_{\delta} = \max_{\xi \in \bar{B}_{\delta}(\Gamma_0)} V(\xi) = \tfrac{l}{2} = \tfrac{1}{6}\varepsilon^2. \tag{3.93}$$

Next, let

$$\sigma = \sqrt{\|y(0) - \mathbf{P}_{X^*}(y(0))\|^2 + \|z(0)\|^2}, \tag{3.94}$$

where $y(0) = \frac{1}{N}(\mathbf{1}_N^T \otimes I_n)x(0)$ and $z(0) = (M \otimes I_n)x(0)$, with M as in (3.27).

The conditions of Theorem 2.5.1 are shown to hold under those of either Theorem 3.6.1 or 3.6.2. With δ constructed as above, it is shown in the proof of Theorem 2.5.1 that there exists a positive, real number α^* such that whenever $\alpha \in (0,\alpha^*)$, the set $\bar{B}_r(\Gamma_0)$ is attractive for the feedback-interconnected system (3.31)-(3.37) on $\bar{B}_{\sigma}(\Gamma_0)$ (q.v. proof of Theorem 2.5.1, §2.5.2), for any $r \in [\delta,\sigma)$, including $r = \rho$. Consequently, any sequence generated by (3.31)-(3.37) and initialized on $\bar{B}_{\sigma}(\Gamma_0)$ eventually becomes arbitrarily close to $\bar{B}_{\delta}(\Gamma_0)$. The continuity of $V(\cdot,\cdot)$ then implies that the limit $\bar{V}$ in (3.87) of the sequence $(V(y(t),z(t)))_{t=0}^{\infty}$ can be no larger than the largest value that $V(\cdot,\cdot)$ attains on $\bar{B}_{\delta}(\Gamma_0)$. These observations, together with Claim 3.7.1 imply that

$$\begin{aligned}\limsup_{t\to\infty} \|x(t) - \mathbf{P}_{\mathbf{X}^*}(x(t))\|^2 &\leq 6\lim_{t\to\infty} V(y(t),z(t)) \\ &= 6\bar{V} \\ &\leq 6V_{\delta}.\end{aligned} \tag{3.95}$$

Combining (3.95) and (3.93) yields the desired ultimate upper bound (3.88) on the agents' collective estimation errors.

In order to prove the second assertion of this lemma, we note that under the assumptions of either Theorem 3.6.1 or 3.6.2, it is shown that $\Delta V(y,z) \leq W(y,z;\alpha)$, where the form of $W(\cdot,\cdot;\alpha)$ depends on which set of assumptions is adopted. Specifically, under the assumptions of Theorem 3.6.1, $W(\cdot,\cdot;\alpha)$ may be taken as in (3.62), while under the assumptions of Theorem 3.6.2, $W(\cdot,\cdot;\alpha)$ may be taken as the right-hand side of (3.71). In either case, the form of $W(\cdot,\cdot;\alpha)$ indicates that for any $r \in (\delta,\rho)$, with δ constructed as above, there exists a positive, real number γ such that

$$-\gamma = \max_{(y^T,z^T)^T \in \mathbb{R}^{(N+1)n} \setminus B_r(\Gamma_0)} W(y,z;\alpha). \tag{3.96}$$

In the proof of Theorem 2.5.1 (q.v. §2.5.2) it is shown that so long as $\alpha \in (0,\alpha^*)$ (with α^* depending on σ, ρ and δ, among other quantities), the bound

$$V(y^+,z^+) \le V(y,z) - \gamma \tag{3.97}$$

holds for all $(y^T,z^T)^T \in \mathbb{R}^{(N+1)n} \setminus B_r(\Gamma_0)$. We take $r=\rho$ and solve this difference inequality to obtain

$$V(y(t),z(t)) \le V_0 - t\gamma, \tag{3.98}$$

where $V_0 = V(y(0),z(0))$ is determined by $x(0)$. The bound (3.98) is valid so long as $(y^T,z^T)^T \in \mathbb{R}^{(N+1)n} \setminus B_\rho(\Gamma_0)$, which is the case whenever $V(y,z) \ge \hat{V}_\rho$, where

$$\hat{V}_\rho = \max_{(y^T,z^T)^T \in \bar{B}_\rho(\Gamma)} V(y,z). \tag{3.99}$$

From the expression for $V(\cdot,\cdot)$ it is evident that $\hat{V}_\rho = N\rho^2$ and in terms of ε, $\hat{V}_\rho = \frac{2N\varepsilon^2}{3}$. Then, from (3.98) we observe that it takes at most

$$T = \left\lceil \frac{V_0 - \frac{2}{3}N\varepsilon^2}{\gamma} \right\rceil \tag{3.100}$$

iterations to reduce the value of $V(\cdot,\cdot)$ to $\frac{2N\varepsilon^2}{3}$, when (3.5) is initialized at $x(0)$. Therefore, by Claim 3.7.1, the collective estimation error $\|x(t) - \mathbf{P}_{\mathbf{X}^*}(x(t))\|^2$ is guaranteed to reduce to $6\hat{V}_\rho$ in at most T iterations. Recalling that $\hat{V}_\rho = \frac{2N\varepsilon^2}{3}$ yields the desired estimate (3.89). $\square$

Remark 3.7.1: In addition to the conclusions drawn in Lemma 3.7.1, the proofs of Lemmas 3.6.1, 3.6.2, 3.6.3, and Theorems 3.6.1 and 2.5.1 are replete with other insights as well. For example, one may ask the converse question to the one answered by Lemma 3.7.1: for a given (sufficiently small) α, how large will the ultimate upper bound on the agents' estimation errors be?

In analogy to Remark 3.9.2, consider the case in which $s_i(\cdot) = \nabla J_i(\cdot)$, and the collective cost $J(\cdot)$ is locally strongly convex with constant β on the set $\bar{B}^n_\sigma(X^*)$. Then, taking $\phi(\cdot) = 2(J(\cdot) - J^*)$ and considering expression (3.51), we observe that $\phi(\cdot)$ dominates the quadratic term $\alpha^2 K_y \|y - \mathbf{P}_{X^*}(y)\|^2$ for all $y \in \bar{B}^n_\sigma(X^*)$, whenever $\alpha \in (0, \frac{\beta}{K_y})$. Consequently, the size of the set $Z(\alpha)$, and therefore the size of the ultimate upper bound on the agents' estimation errors, is dictated exclusively by the size of the constant term $\alpha^2 K$ in (3.51).

From the expression for K in (3.54), it is interesting to note that K vanishes when $s^* = 0$. In that case it can be shown that whenever $\alpha \in (0, \frac{\beta}{K_y})$, $X^* \times \{\mathbf{0}\}$ is asymptotically stable for (3.31)-(3.37), meaning that agents reach a perfect consensus on a collective optimizer. Rewriting expression (3.55) as

$$s^* = \max_{x^* \in X^*} \sum_{i \in \mathscr{V}} \|\nabla J_i(x^*)\|^2, \tag{3.101}$$

we observe that $s^* = 0$ if, and only if $X^* \subseteq X_i^*$, for all $i \in \mathscr{V}$, which is equivalent to requiring the intersection $\cap_{i \in \mathscr{V}} X_i^* \supseteq X^*$ to be nonempty. We thus recover the main observation made in [140], for this case.

Going further, we observe that a non-zero s^* in (3.101) can be interpreted as a measure of discord among the agents' individual preferences; a quantity that measures the extent to which agents must compromise their individual interests in order to optimize the collective objective. An insight thus gained from these considerations is that the agents' ultimate estimation errors are proportional to the discord in their individual preferences, as one would expect. $\diamondsuit$

3.8 An Application to Weighted Gradient Methods

In what follows, we show how the results of Section 3.6.1 may be applied by verifying assumptions A3.3.3, A3.3.2 and A3.3.4 for a specific instance of (3.5).

We consider a class of centralized optimization algorithms of the form

$$y(t+1) = \mathbf{P}_X \left[y(t) - \alpha_o H(t) \nabla J(y(t)) \right], \tag{3.102}$$

where for each $t \in \mathbb{N}$, $H(t) \in \mathbb{R}^{n \times n}$ is symmetric and positive definite, and $y(t) \in \mathbb{R}^n$. There are several algorithms that fall within this class – q.v. the discussion in §1.2, [15].

According to Lemma 3.9.3 and Remark 3.9.1, there exists an $\alpha_o^* \in \mathbb{R}_{++}$ such that the centralized algorithm (3.102) solves problem (3.1) whenever $\alpha_o \in (0, \alpha_o^*)$, provided that $s_o(y) = H(t)\nabla J(y)$ is locally Lipschitz continuous and satisfies A3.3.2.

Supposing we have a network of agents whose weighted adjacency matrix satisfies a consensus assumption such as A3.3.1, algorithm (3.102) may be decentralized according to (3.5) by taking $s_i \in (\Theta \circ J_i)(\cdot)$, with

$$\Theta(J_i(\cdot)) = \{H(t)\nabla J_i(\cdot)\}. \tag{3.103}$$

Clearly, $H(t)\nabla J_i(\cdot)$ is linear, and A3.3.3 is satisfied. Moreover, if there exists a number $M > 0$ such that $\|H(t)\| \leq M$ for all t, then A3.3.4 is satisfied under the assumption that the gradient of each $J_i(\cdot)$ is Lipschitz continuous.

The following lemma makes the foregoing observations precise, providing a set of conditions under which the consensus-decentralized version of algorithm (3.102) has the convergence properties specified in Theorem 3.6.1 and Lemma 3.7.1.

Lemma 3.8.1: Consider a network of agents whose goal is to solve problem (3.1), and assume that the following hold:

- The agents communicate over a graph $\mathscr{G}_C = (\mathscr{V}, \mathscr{E}_C)$ whose weighted adjacency matrix satisfies A3.3.1.
- The collective cost $J(\cdot)$ is given by (7.8) and the collective constraint X by (3.4).
- For each $i \in \mathscr{V}$, agent i has private knowledge of X_i and J_i, and implements (3.5), with $s_i \in (\Theta \circ J_i)(\cdot)$ and $(\Theta \circ J_i)(\cdot)$ as in (3.103).

Then, the conditions of Theorem 3.6.1 and Lemma 3.7.1 are satisfied provided that the following hold:

- H1: For each $i \in \mathscr{V}$, $\nabla J_i(\cdot)$ is locally Lipschitz.
- H2: For each $t \in \mathbb{N}$, $H(t)$ is symmetric and there exist positive, real numbers m and M such that $m \leq \|H(t)\| \leq M$.

- H3: The collective cost $J(\cdot)$ is convex, and such that

$$\arg\min_{x\in X} J(x) = \arg\min_{x\in\hat{X}} J(x), \tag{3.104}$$

where $\hat{X}$ is as in (3.13).

Proof. Since A3.3.1 is assumed in the statement of the Lemma, it remains to show that A3.3.3, A3.3.2 and A3.3.4 are implied by H1, H2 and H3. From the foregoing discussion, it is clear that A3.3.3 is satisfied and that H1 and H2 imply A3.3.4. It therefore only remains to show that $s_o(\cdot) = H(t)\nabla J(\cdot)$ is strictly pseudogradient with respect to $y \mapsto \|y - \mathbf{P}_{X^*}(y)\|^2$, on the set $\hat{X}$ given in (3.13).

Since $H(t)$ is symmetric and positive definite, it can be decomposed according to $H(t) = U(t)D(t)U^T(t)$, where $U(t)$ is an orthonormal matrix and $D(t) = \mathrm{diag}(\sigma(H(t)))$. For any vectors v and w in $\mathbb{R}^n$, we let $\tilde{v} = U(t)^T v$ and $\tilde{w} = U(t)^T w$, and observe that

$$\begin{aligned}
v^T H(t) w &= v^T U(t) D(t) U^T(t) w \\
&= \sum_{i=1}^{n} \sigma_i \tilde{v}_i \tilde{w}_i \\
&\geq m \sum_{i=1}^{n} \tilde{v}_i \tilde{w}_i \qquad (3.105)\\
&= m v^T U(t) U(t)^T w \\
&= m v^T w,
\end{aligned}$$

where (3.105) is obtained by applying H2. To apply this observation, we take $v = \nabla J_i(\cdot)$ and $w = y - \mathbf{P}_{X^*}(y)$, and notice that for all $y \in \mathbb{R}^n$,

$$\begin{aligned}
s_o(y)^T (y - \mathbf{P}_{X^*}(y)) &= \nabla J(y)^T H(t)^T (y - \mathbf{P}_{X^*}(y)) \\
&\geq m \nabla J(y)^T (y - \mathbf{P}_{X^*}(y)).
\end{aligned}$$

Since $J(\cdot)$ is assumed to be convex, it holds that for all $y \in \mathbb{R}^n$, $\nabla J(y)^T (y - \mathbf{P}_{X^*}(y)) \geq J(y) - J(\mathbf{P}_{X^*}(y))$. By (3.104) and the fact that $y \in \hat{X}$, $\forall t \in \mathbb{N}$, the quantity $J(y) - J(\mathbf{P}_{X^*}(y))$ is positive definite with respect to X^* on $\hat{X}$. Therefore, in light of Remark 3.3.5, A3.3.2 is satisfied with $\phi(y) = \frac{m}{2}(J(y) - J^*)$, which, from the global convexity of $J(\cdot)$ is seen to be continuous, radially unbounded and positive definite with respect to X^* on $\hat{X}$. $\square$

Remark 3.8.1: In the literature, it is often assumed that agents' individual cost functions are convex, regardless of whether their search directions are assumed to be Lipschitz continuous or not [72], [114], [116], [163], [153], [112], [173], [53]. Though H3 in Lemma 3.8.1 requires the collective cost $J(\cdot)$ to be convex, agents' individual costs need not be convex in order for the conditions of Theorem 3.6.1 and Lemma 3.7.1 to hold. $\diamondsuit$

Remark 3.8.2: One caveat to the implementation of (3.5) with $\Theta(\cdot)$ as in (3.103), is that in addition to having common access to the gain α, each agent also requires knowledge of $H(t)$, for each t. In other words, the agents must generate $H(t)$ by means of the same algorithm, initialized at a commonly known $H(0)$. We conjecture that the interesting case in which each agent updates its own matrix $H_i(t)$, which is not necessarily identical to $H_j(t)$, $i \neq j$, can be analyzed by means of the framework proposed here, under the assumption that $(H_i(t))_{t=0}^{\infty} \to (H(t))_{t=0}^{\infty}$, where for each t, $H(t)$ is symmetric and satisfies the bounds $m \leq \|H(t)\| \leq M$, for some positive, real numbers m and M. $\diamondsuit$

3.9 Proofs of Lemmas 3.5.1 and 3.5.2, and a Remark on Separation Theory

It is interesting to observe that a set of conditions guaranteeing the convergence of the decentralized algorithm (3.5) (with $s_i \in (\Theta \circ J_i)(\cdot)$ and $\Theta(\cdot)$ as in (3.103)), may be taken as the disjoint union of those conditions guaranteeing the convergence of a consensus process on its own (i.e. A3.3.1), with those conditions guaranteeing the convergence of the centralized algorithm (3.102) (i.e. A3.3.2, and A3.3.4).

This observation suggests that Theorem 3.6.1 may be regarded as a "separation theorem", akin to the separation theorems one finds in the literature on nonlinear output feedback [148]. Specifically, in the context of CO algorithms such as (3.5), the idealized mean subsystem dynamics correspond to some centralized optimization algorithm, while the idealized deviation dynamics correspond to the deviation dynamics of some standard consensus algorithm. The feature of separation theory that is appealing in this context is the possibility of designing specific CO algorithms consisting of any combination of interchangeable centralized optimization methods and consensus algorithms. A full "separation theory" for CO methods would be able to provide performance guarantees for any particular CO algorithm from the separate stability properties of its constituent centralized optimization method and consensus algorithm.

In the sections that follow, we show how those assumptions in §3.3 pertaining to the idealized mean and deviation dynamics can be used to establish their stability properties. Lemma 3.5.1 is proved in a sequence of lemmas presented in §3.9.1, and Lemma 3.5.2 is proved in §3.9.2. The proofs of these lemmas are included only for completeness; most of the techniques being applied are showcased in the proofs of Theorems 3.6.1 and 3.6.2, which pertain to the overall interconnection in Figure 3.1.

3.9.1 Proof of Lemma 3.5.1

In the following lemmata, we show that A3.3.2 and either A3.3.4 or A3.3.4$'$ are sufficient to guarantee the SPAS of X^* for the idealized mean dynamics given by (3.36). The following lemma applies to both cases.

Lemma 3.9.1: Consider the algorithm (3.36), where $s_o(y) \in \Theta(J(y))$, with $(\Theta \circ J)(\cdot)$ satisfying A3.3.2. Then, for all $y \in \hat{X}$,

$$\Delta V_{\mathscr{O}}(y) \leq -\alpha\phi(y) + \tfrac{\alpha^2}{N}\|s_o(y)\|^2, \tag{3.106}$$

where $\Delta V_{\mathscr{O}}(y) = V_{\mathscr{O}}(y^+) - V_{\mathscr{O}}(y)$, $V_{\mathscr{O}}(y) = N\|y - \mathbf{P}_{X^*}(y)\|^2$, and $\phi(\cdot) \in C^0[\mathbb{R}^n, \mathbb{R}_+]$ is radially unbounded and positive definite with respect to X^* on $\hat{X}$.

Proof. We have

$$\Delta V_{\mathscr{O}}(y) = N\|\mathbf{P}_{\hat{X}}(y^+) - \mathbf{P}_{X^*}(y^+)\|^2 - N\|y - \mathbf{P}_{X^*}(y)\|^2.$$

Since $X^* \subseteq X \subseteq \hat{X}$, $\mathbf{P}_{X^*}(y) = \mathbf{P}_{\hat{X}}(\mathbf{P}_{X^*}(y))$. Consequently,

$$\begin{aligned}\Delta V_{\mathscr{O}}(y) &= N\|\mathbf{P}_{\hat{X}}(y^+) - \mathbf{P}_{\hat{X}}(\mathbf{P}_{X^*}(y^+))\|^2 - N\|y - \mathbf{P}_{X^*}(y)\|^2 \\ &\leq N\|y^+ - \mathbf{P}_{X^*}(y^+)\|^2 - N\|y - \mathbf{P}_{X^*}(y)\|^2,\end{aligned}$$

which is obtained from the fact that the Euclidean projection is non-expansive (q.v. Proposition 2.2.1, [14]). By the definition of the projection operation (q.v. (A.2)), $\mathbf{P}_{X^*}(y^+)$ is the point inside X^* which is closest to y^+, meaning that the distance between y^+ and any other point inside X^* is at least as large as $\|y^+ - \mathbf{P}_{X^*}(y^+)\|$. In particular, we have that

$$\begin{aligned}\Delta V_{\mathscr{O}}(y) &\leq N\|y^+ - \mathbf{P}_{X^*}(y^+)\|^2 - N\|y - \mathbf{P}_{X^*}(y)\|^2 \\ &\leq N\|y^+ - \mathbf{P}_{X^*}(y)\|^2 - N\|y - \mathbf{P}_{X^*}(y)\|^2.\end{aligned}$$

Then, expanding the first term according to (3.36) and cancelling the last term gives

$$\begin{aligned}\Delta V_{\mathscr{O}}(y) &\leq N\|y - \tfrac{\alpha}{N}s_o(y) - \mathbf{P}_{X^*}(y)\|^2 - N\|y - \mathbf{P}_{X^*}(y)\|^2 \\ &\leq \tfrac{\alpha^2}{N}\|s_o(y)\|^2 - 2\alpha(y - \mathbf{P}_{X^*}(y))^T s_o(y).\end{aligned}$$

Noting from (3.36) that $y \in \hat{X}$ at each $t \in \mathbb{N}$ allows us to apply A3.3.2 (c.f. (3.15)) to obtain the desired result. □

Remark 3.9.1: From the first part of the proof of Lemma 3.9.1, it is evident that under the same conditions, the same conclusions can be drawn for the algorithm

$$y^+ = \mathbf{P}_X\left[y - \tfrac{\alpha}{N}s_o(y)\right], \tag{3.107}$$

which may be regarded as the centralized version of (3.5). ♢

The Case with Locally Lipschitz $s_o(y)$

In the proof of the following lemma, we use the assumption that $s_o(\cdot)$ is locally Lipschitz, which is implied by A3.3.4.

Lemma 3.9.2: Consider the algorithm (3.36). Suppose that in addition to the conditions in Lemma 3.9.1, $(\Theta \circ J)(\cdot) = \{s_o(\cdot)\}$, and $s_o(\cdot)$ is locally Lipschitz. Then, given an arbitrarily large compact set $\Omega \subset \mathbb{R}^n$ containing X^*, it holds that for all $y \in \Omega \cap \hat{X}$,

$$\Delta V_{\mathscr{O}}(y) \leq -\alpha\phi(y) + 2\alpha^2\tfrac{L^2}{N}\|y - \mathbf{P}_{X^*}(y)\|^2 + 2\alpha^2\tfrac{s^{*2}}{N},$$

where

$$s^* = \max_{y \in X^*}\|s_o(y)\|, \tag{3.108}$$

and L denotes the Lipschitz constant associated to $s_o(\cdot)$, over the set $\Omega \cap \hat{X}$.

Proof. Consider the expression (3.106). We have

$$\begin{aligned}\|s_o(y)\|^2 &= \|s_o(y) - s_o(\mathbf{P}_{X^*}(y)) + s_o(\mathbf{P}_{X^*}(y))\|^2 \\ &\leq 2\|s_o(y) - s_o(\mathbf{P}_{X^*}(y))\|^2 + 2\|s_o(\mathbf{P}_{X^*}(y))\|^2 \qquad (3.109) \\ &\leq 2L^2\|y - \mathbf{P}_{X^*}(y)\|^2 + 2\|s_o(\mathbf{P}_{X^*}(y))\|^2, \qquad (3.110)\end{aligned}$$

where (3.109) is obtained by applying Young's inequality, and (3.110) by invoking the Lipschitz continuity of $s_o(\cdot)$. We obtain the desired result by noting that the maximum s^* in (3.108) exists since X^* is assumed to be compact and $s_o(y)$ is continuous. □

By means of Theorem 2.5.1, the result obtained in Lemma 3.9.2 allows us to conclude that the set X^* is SPAS for the dynamics (3.36).

Lemma 3.9.3: Suppose that all the conditions of Lemma 3.9.2 are satisfied. Then, X^* is SPAS for (3.36).

Proof. We apply Theorem 2.5.1, with $\Gamma_0 = X^*$ and $\Xi = \hat{X}$. The function $V_{\mathscr{O}}(y) = N\|y - \mathbf{P}_{X^*}(y)\|^2$ is continuous, radially unbounded and positive definite with respect to the set Γ_0 on $\mathbb{R}^n$.

Let σ be an arbitrary, positive, real number, and take $\Omega = \bar{B}_\sigma(\Gamma_0)$. Then, under the conditions in Lemma 3.9.2, we have that for all $y \in \Omega \cap \hat{X} = \bar{B}_\sigma(\Gamma_0) \cap \Xi$,

$$\Delta V_{\mathscr{O}}(y) \leq W(y;\alpha), \tag{3.111}$$

where

$$W(y;\alpha) = -\alpha\phi(y) + 2\alpha^2 \tfrac{L^2}{N}\|y - \mathbf{P}_{X^*}(y)\|^2 + \alpha^2 K, \tag{3.112}$$

$$K = \frac{2s^{*2}}{N}, \tag{3.113}$$

and $\phi(\cdot) \in C^0[\mathbb{R}^n, \mathbb{R}_+]$ is radially unbounded and positive definite with respect to X^* on $\hat{X}$.

It remains to show that for the given number σ, there exists a positive, real number $\bar{\alpha}$, such that whenever $\alpha \in (0, \bar{\alpha})$, $W(\cdot;\alpha)$ has properties P1, P2 and P3 specified in the statement of Theorem 2.5.1.

Let δ be an arbitrary, positive real number as in P3. Since $\phi(\cdot)$ is radially unbounded and positive definite with respect to X^* on $\hat{X}$, Lemma 2.5.2 implies that for any $K_\phi \in \mathbb{R}_{++}$, there exists an $\alpha_\phi \in \mathbb{R}_{++}$ such that

$$W(y;\alpha) \leq -\alpha^2\left(K_\phi - \tfrac{2L^2}{N}\right)\|y - \mathbf{P}_{X^*}(y)\|^2 + \alpha^2 K,$$

for all $y \in \bar{B}_\sigma(X^*) \backslash B_\delta(X^*)$, provided $\alpha \in (0, \alpha_\phi)$. By choosing

$$K_\phi = \frac{K}{\delta^2} + \tfrac{2L^2}{N},$$

we obtain that

$$W(y;\alpha) \leq -\alpha^2 \tfrac{K}{\delta^2}\|y - \mathbf{P}_{X^*}(y)\|^2 + \alpha^2 K, \tag{3.114}$$

for all $y \in \bar{B}_\sigma(X^*) \backslash B_\delta(X^*)$, whenever $\alpha \in (0, \alpha_\phi)$.

Clearly, $W(y;\alpha) < 0$ for all $y \in \bar{B}_\sigma(X^*) \backslash \bar{B}_\delta(X^*)$. On the other hand from (3.112), we observe that $W(\cdot;\alpha)$ can be no larger than $\alpha^2(2L^2\delta^2 + K)$ on the set $\bar{B}_\delta(X^*)$. Therefore,

$$W(y;\alpha) \leq \alpha^2(\tfrac{2L^2}{N}\delta^2 + K),$$

for all $y \in \bar{B}_\sigma(X^*)$, and P1 is satisfied with $b_W(\alpha) = \alpha^2(\frac{2L^2}{N}\delta^2 + K)$.

From (3.112), the positive definiteness of $\phi(\cdot)$ with respect to X^* implies that $Z(\alpha) \supseteq X^*$, and P2 is satisfied. From the observation that $W(y;\alpha) < 0$ on $y \in \bar{B}_\sigma(X^*) \backslash \bar{B}_\delta(X^*)$, we see that $Z(\alpha) \subseteq \bar{B}_\delta(X^*)$. Since δ is arbitrary, P3 is satisfied.

Having satisfied all of the conditions of Theorem 2.5.1 (with $\alpha_Z = \bar{\alpha} = \alpha_\phi$), we conclude that under the conditions of Lemma 3.9.2, the set X^* is SPAS for (2.31). □

Remark 3.9.2: For the unconstrained case in which $s_o(y) = \nabla J(y)$, $s^* = 0$ by the first-order necessary condition for optimality. If $J(\cdot)$ is locally strongly convex with constant β, it can be shown that

$$\Delta V_{\mathscr{O}} \leq -\tfrac{\alpha}{N}(\beta - 2\alpha L^2)V_{\mathscr{O}}.$$

Then, by means of Theorem 2.5.2, we may conclude that X^* is asymptotically stable for (3.36), whenever $\alpha \in (0, \frac{\beta}{2L^2})$. $\diamondsuit$

The Case with Locally Bounded $s_o(y)$

Lemma 3.9.4: Consider the algorithm (3.36). Suppose that in addition to the conditions in Lemma 3.9.1, $(\Theta \circ J)(\cdot)$ satisfies the local boundedness assumption A3.3.4$'$. Then, for all $y \in \Omega$,

$$\Delta V_{\mathscr{O}}(y) \leq -\alpha\phi(y) + \alpha^2 \tfrac{B^2}{N}. \tag{3.115}$$

Proof. Expression (3.115) follows directly from (3.106) and (3.18). $\square$

By means of Theorem 2.5.1, the result obtained in Lemma 3.9.4 allows us to conclude that the set X^* is SPAS for the dynamics (3.36).

Lemma 3.9.5: Suppose that all the conditions of Lemma 3.9.4 are satisfied. Then, X^* is SPAS for (3.36).

Proof. We apply Theorem 2.5.1, with $\Gamma_0 = X^*$ and $\Xi = \hat{X}$. The function $V_{\mathscr{O}}(y) = N\|y - \mathbf{P}_{X^*}(y)\|^2$ is continuous, radially unbounded and positive definite with respect to the set Γ_0 on $\mathbb{R}^n$.

Let σ be an arbitrary, positive, real number, and take $\Omega = \bar{B}_\sigma(\Gamma_0)$. Then, under the conditions in Lemma 3.9.4, we have that for all $y \in \Omega \cap \hat{X} = \bar{B}_\sigma(\Gamma_0) \cap \Xi$,

$$\Delta V_{\mathscr{O}}(y) \leq -\alpha\phi(y) + \alpha^2 \tfrac{B^2}{N}. \tag{3.116}$$

Taking

$$W(y;\alpha) = -\alpha(\phi(y) - \alpha\tfrac{B^2}{N}), \tag{3.117}$$

we see that for all $y \in \bar{B}_\sigma(\Gamma_0)$, $W(y;\alpha) \leq b_W(\alpha)$, with $b_W(\alpha) = \alpha^2 \frac{B^2}{N}$ being such that $\lim_{\alpha \downarrow 0} b_W(\alpha) = 0$. By virtue of the radial unboundedness and positive definiteness of $\phi(\cdot)$ with respect to X^*, the set

$$Z(\alpha) = \{y \in \mathbb{R}^n \mid \phi(y) \leq \alpha\tfrac{B^2}{N}\}, \tag{3.118}$$

on which $W(\cdot;\alpha)$ is non-negative, is compact and contains X^* for any $\alpha > 0$. Moreover, $Z(\alpha)$ can be made to fit inside an arbitrarily small ball centered at Γ_0 by selecting α sufficiently small. Specifically, let $\delta \in \mathbb{R}_{++}$ be arbitrary. By Lemma 2.5.1 (taking $\hat{\rho} = \delta$), there exists a number $c \in \mathbb{R}_{++}$ such that the set $\Phi_c = \{y \in \mathbb{R}^n \mid \phi(y) \leq c\}$ is strictly contained inside $\bar{B}_\delta(X^*)$. From (3.118), it is evident that whenever $\alpha \in (0, \alpha_Z)$, with

$$\alpha_Z = \frac{Nc}{B^2}, \tag{3.119}$$

$Z(\alpha)$ is strictly contained inside Φ_c, and hence inside $\bar{B}_\delta(X^*)$.

Having satisfied all of the conditions of Theorem 2.5.1, with $\bar{\alpha} = \alpha_Z$, we conclude that under the conditions of Lemma 3.9.4, the set X^* is SPAS for (2.31). $\square$

3.9.2 Proof of Lemma 3.5.2

In what follows, we prove Lemma 3.5.2, by applying Theorem 2.5.2. We begin with the following claim.

Claim 3.9.1: The subspace $\mathscr{A}^\perp$ is invariant for (3.39).

Proof. Suppose that for some $t \in \mathbb{N}$, $z \in \mathscr{A}^\perp$. Let v be any vector of the form

$$v = \sum_{i\in\mathscr{V}} c_i(\mathbf{1}_N \otimes e_i), \quad c_i \in \mathbb{R},\, i \in \mathscr{V}, \tag{3.120}$$

meaning $v \in \mathscr{A}$. Then, by the properties of the Kronecker product and the column stochasticity of A,

$$\begin{aligned} v^T z^+ &= \sum_{i\in\mathscr{V}} c_i(\mathbf{1}_N \otimes e_i)^T (A \otimes I_n) z && (3.121)\\ &= \sum_{i\in\mathscr{V}} c_i(\mathbf{1}_N^T A \otimes e_i^T I_n) z && (3.122)\\ &= v^T z && (3.123)\\ &= 0, && (3.124)\end{aligned}$$

which implies that $z^+ \in \mathscr{A}^\perp$, establishing the claim. $\diamondsuit$

The Lyapunov function candidate $V_{\mathscr{C}}(z) = \|z\|^2$ evolves along the sequences generated by (3.39) according to

$$V_{\mathscr{C}}(z^+) - V_{\mathscr{C}}(z) = z^T (A \otimes I_n)^T (A \otimes I_n) z - \|z\|^2. \tag{3.125}$$

By Claim 3.9.1, $z(0) \in \mathscr{A}^\perp$ implies that $z(t) \in \mathscr{A}^\perp$, for all $t \in \mathbb{N}$. Therefore, by A3.3.1 and its implication described in Remark 3.3.2, we obtain

$$\Delta V_{\mathscr{C}}(z) \le -(1-\mu)\|z\|^2. \tag{3.126}$$

Since $(1-\mu)\|\cdot\|^2 \in \mathscr{K}$ and (3.39) is equivalent (by Claim 3.9.1) to the dynamics

$$z^+ = \mathbf{P}_{\mathscr{A}^\perp}\big((A \otimes I_n) z\big), \quad z(0) \in \mathscr{A}^\perp, \tag{3.127}$$

we conclude from Theorem 2.5.2 that $\{\mathbf{0}\}$ is globally, asymptotically stable for (3.39) on $\mathscr{A}^\perp$.

3.10 Final Remarks

As an initial step toward addressing a class of DT-DDCCPs represented by Problem 1.2.2 we introduced a set of tools for the derivation of convergence conditions for a class of consensus-decentralized optimization (CO) algorithms. The novelty in our analytic approach to this class of algorithms is to treat the evolution of the agents' mean estimate and the vector of deviations from this mean, as the feedback interconnection of two nonlinear, dynamical systems. Then, small-gain techniques are employed to derive conditions ensuring the internal stability (i.e. stability in the sense of Lyapunov) of the interconnection (q.v. Theorems 3.6.1 and 3.6.2).

We emphasize internal stability because it is the most fundamental property of relevance in control applications, and this emphasis distinguishes our analysis from that in most existing literature on decentralized optimization. One of the analytic tools we contribute in connection to this chapter is a theorem that characterizes the semiglobal, practical, asymptotic stability of a set of fixed points associated with a nonlinear discrete-time dynamical system, in terms of certain properties of its associated Lyapunov function (q.v. Theorem 2.5.1). We refer to this theorem as the SPAS theorem, and we provide its proof in Chapter 2.

The application of this analytic framework to the static optimization setting has precipitated several observations that may be considered of independent interest to the area of decentralized optimization itself. These observations include the following.

- In the literature on consensus optimization, the effects of the dynamic coupling between the mean and deviation variables are usually suppressed by a combination of two persisting assumptions: the boundedness of agents' individual constraint sets (or the subgradients used to form their search directions), and the use of diminishing step sizes. Instead of suppressing the effects of dynamic coupling among subsystems, the literature on interconnected systems emphasizes techniques that seek to exploit this coupling in order to avoid conservatism [104]. Indeed, in our case a careful examination of the interconnection structure reveals that the destabilizing effects of the projection-related terms arising in one subsystem negate related effects arising in the other (q.v. Lemma 3.6.3). The conclusion to be drawn is that the presence of the projection operation in (3.5) need not add any conservatism to the final convergence condition.

- The individual constraint sets X_i in (1.32) need not be bounded or identical. Their intersection X also need not be bounded. This is true regardless of whether $s_i(\cdot)$ in (3.5) represents a gradient or a subgradient.

- The step-size α may remain fixed throughout the execution of algorithm (3.5), regardless of whether $s_i(\cdot)$ represents a gradient or a subgradient, even when the constraint sets X_i are not bounded.

- In the literature on decentralized optimization, it is typically assumed that agents' individual cost functions $J_i(\cdot)$ are convex, whether or not they are differentiable. For the case in which each agent's search direction $s_i(\cdot)$ is locally Lipschitz and based on a gradient of the agent's private cost function, our convergence conditions indicate that the convexity of all agents' individual costs is not necessary (q.v. Lemma 3.8.1).

- In contrast with existing Lyapunov-based analyses of related decentralized optimization schemes, the application of our SPAS theorem does not require a precise characterization of the algorithm's actual set of fixed points, and allows for the study of iterative methods directly in discrete-time.

Aside from preparing us to address the DT-DDCCP, the basic analysis philosophy presented in this chapter also turns out to be useful for the analysis of continuous-time variants of (3.5), which we study in the next chapter.

Chapter 4

Continuous-Time Consensus Optimization with Positivity Constraints

4.1 Synopsis

We adapt the analytic techniques introduced in Chapter 3 to derive convergence conditions for a class of continuous-time consensus optimization dynamics. We show that introducing a tunable *consensus gain* parameter can relax the upper bound imposed on the step-size in order to guarantee the stability of the collective optimum. The analysis is made challenging by the presence of a logical projection operation used to enforce a positivity constraint on the evolution of the dynamics. This class of systems can be employed in the design of cooperative resource allocation schemes that require no central coordination. More generally, continuous-time consensus optimization dynamics have potential applications in the development of certain behavioral models of animal groups. This chapter is mostly based on work appearing in [94].

4.2 Introduction

There are several motivations for studying the continuous-time counterparts of discrete-time consensus optimization algorithms. First, the set of analytic tools and techniques available for the study of continuous-time dynamical systems in general is far better developed than that for discrete-time systems, especially when nonlinearities are involved. This is likely the reason that many of the early contributions to the field of decentralized optimization (made primarily in the context of decomposition methods) are set in continuous time [8], [79], [48], [99]. With a wider variety of tools, comes the potential for new insights. Second, designing networked, multiagent engineered systems, for which discrete-time implementations are certainly appropriate, is not the only potential application of these schemes; in Section 4.6, we explore an application of continuous-time consensus-optimization (CO) dynamics to the development of certain behavioral models of animal groups. In the context of modelling, continuous-time dynamics are naturally more appropriate. Finally, a prevalent technique in analyzing a wide variety (discrete-time) stochastic approximation methods involves establishing the stability of their continuous-time counterparts [19], and this applies also in the decentralized setting [16], [145].

With the exception of [127], [100], [54], [163] and [139], most studies of CO methods focus on discrete-time algorithms [155], [75], [114], [116], [112], [172], [169], [44], [16], [93], [91]. Analyses of such algorithms often involve a direct examination of portions of the trajectories they generate, the use of diminishing step-sizes, and the use of time-averaged sequences (i.e. Cesàro averages). Such techniques may not be viable in studying even the simplest continuous-time variants of these algorithms; for example, finding an analytic expressions for a solution to a system of nonlinear ordinary differential equations is notoriously difficult for all but a small number of cases.

On the other hand, the stability analysis of unconstrained, continuous-time CO dynamics is rather simple when approached within the Lyapunov-based framework considered in Chapter 3. Adopting the key idea in Chapter 3, we treat the evolution of the mean and deviation of agents' estimates as two coupled dynamic subsystems, and derive conditions for the GPAS of their interconnection. As in the discrete-time case, this analytic approach does not require an examination of the trajectories themselves, provides a fixed step-size convergence condition, and yields stability properties of the collective optimum that pertain to the system trajectories themselves (i.e., not their time averages). Moreover, the approach allows us to derive explicit expressions for a lower bound on the system's convergence rate and an upper bound on the agents ultimate estimation error, in terms of relevant problem parameters.

The direct continuous-time counterpart of the discrete-time algorithm introduced in [114] has a velocity vector field which is comprised of a linear graph Laplacian term engendering the system's consensus behavior, driven by a gradient term which is weighted by a tunable step-size (i.e., optimization gain) parameter. We consider a variation of these basic CO dynamics, in which we introduce a consensus gain as a second tunable parameter. Several interesting observations can be made concerning the utility of this parameter, which can adjust the strength of the consensus term relative to that of the optimization term. We show that this tuning parameter can be used to improve the ultimate accuracy of the algorithm without affecting the convergence rate of the mean of agents' estimates. More importantly, this parameter can be used to relax the upper bound imposed on the step-size in order to guarantee the stability of the collective optimum. These observations are made possible by the interconnected-systems viewpoint, and appear to have not been made elsewhere in the literature.

The class of systems that we study here is the consensus-weighted variation of the basic CO dynamics, modified by a logical projection operation used to enforce a positivity constraint on its trajectories. Remarkably, as in the discrete-time case, a careful examination of the interconnection terms in the composite Lyapunov argument leads to an elegant result concerning the effect of the logical projection operation on the evolution of the continuous-time CO dynamics. This class of systems may be useful in the design of cooperative resource allocation schemes, in which the computation of the optimal resource "pricing parameter", which is required to remain non-negative, is distributed among the agents.

Other studies of continuous-time CO-like schemes such as [127], [100], [54], [163] and [139] also employ Lyapunov analysis techniques. In contrast to the present work, [54] and [163] solve the CO problem by means of primal-dual saddle point dynamics, and employ Lyapunov arguments that generalize those originating in [8]. The setting in [54] is more general than ours in that directed graphs are considered. In [127], a stochastic, continuous-time mirror descent method is executed by a number of agents in the presence of noise. As in [154], each agent has knowledge of the collective objective function, and the consensus dynamics are exploited primarily to mitigate the effects of this noise. The choice of the Lyapunov function in [127] relates to the geometry of the constraint set and plays an important role in the method's performance for very high-dimensional problems. The setting in [139] involves a number of agents that are assumed to have a priori knowledge of the geometric properties of the set of optimizers for their individual, privately known objective functions. In contrast with our setting, these sets of optimizers are assumed to have a non-empty intersection, and the agents employ consensus dynamics in order reach an agreement on some point within this intersection. The Lyapunov-like function employed in the analysis measures the maximum among distances between agents' estimates and this intersection. In [100], the authors propose a set of “zero gradient sum” algorithms as a simple, and novel alternative to standard CO methods in solving the CO problem. When initialized appropriately, the agents’ trajectoreis evolve (and eventually reach consensus) on an invariant manifold characterized by the sum of individual cost function gradients being equal to zero. The algorithms are specified by means of a “Lyapunov-redesign”-like technique, in which a Lyapunov function candidate is selected, and the agent dynamics are designed in order to render its time derivative negative definite.

This chapter is organized as follows. In Section 4.3 we describe our problem setting and state our assumptions. We give our analysis in Section 4.4, and provide an application example with simulations in Section 4.5.

4.3 Problem Setting

We consider a network of N agents that communicate over a graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, where $\mathscr{V} = \{1, \ldots, N\}$ is a set that indexes the agents, and $\mathscr{E} \subseteq \mathscr{V} \times \mathscr{V}$ specifies the pairs of agents that communicate. For now we assume that this communication graph remains fixed in time. We associate a matrix $A \in \mathbb{R}^{N\times N}$ to $\mathscr{G}$ in the following way:

$$[A]_{i,j} = \begin{cases} -|\{k : (k,i) \in \mathscr{E}\}|, & i = j, \\ 1, & i \neq j \text{ and } (j,i) \in \mathscr{E}, \\ 0, & \text{otherwise} \end{cases} \tag{4.1}$$

Thus, the matrix A is the negative of the graph Laplacian, and we make the following standard assumption on the communication structure from which it is induced [136]:

A4.3.1: The graph $\mathscr{G}$ is connected and undirected. $\diamondsuit$

By construction, the null space of A coincides with the so-called “agreement subspace” span$\{\mathbf{1}\}$ – that is, $A\mathbf{1} = \mathbf{0}$. Since $\mathscr{G}$ is undirected, A is symmetric and therefore we also have that $\mathbf{1}^T A = \mathbf{0}^T$, and that all of A’s eigenvalues are real. Most importantly, A4.3.1 implies that the largest eigenvalue of A is unique and equal to zero; consequently, for all $z \in \text{span}\{\mathbf{1}\}^\perp$, $z^T A z \leq -\lambda_2 \|z\|^2$, where $\lambda_2 > 0$ is the Fiedler eigenvalue of $-A$.

The agents’ collective task is to cooperatively locate a point $y^* \in \mathbb{R}_+$ that minimizes the collective cost function $J : \mathbb{R} \to \mathbb{R}$, which is comprised of N additive components – namely $J(y) = J_1(y) + \ldots + J_N(y)$, with $J_i(\cdot)$ representing the ith agent’s individual objective. We make the following assumptions on this cost structure:

A4.3.2: (a) The collective cost J is convex on $\mathbb{R}_+$, and there exists a unique $y^* \in \mathbb{R}_+$ satisfying

$$J(y^*) < J(y), \quad \forall y \in \mathbb{R}_+ \setminus \{y^*\} \tag{4.2}$$

(b) There exists a number $\bar{r} \in \mathbb{R}_+ \cup \{\infty\}$ such that for all $r \in (0, \bar{r})$, there is a real number $\kappa_J(r) > 0$, for which the following holds:

$$J(y^*) - J(y) \leq -\kappa_J(r)|y - y^*|^2, \tag{4.3}$$

for all $y \in \bar{B}_r(y^*) \cap \mathbb{R}_+$.

(c) For each $i \in \mathscr{V}$, $\nabla J_i(\cdot)$ is globally Lipschitz continuous with constant L_i. $\diamondsuit$

Remark 4.3.1: If $y^* > 0$ and $J(\cdot)$ is strongly convex with constant β, then A4.3.2 (b) is satisfied with $\bar{r} = \infty$ and $\kappa_J(r) = \frac{\beta}{2}$, independently of r[1]. On the other hand, if $J(y) = |y - y^*|$ with $y^* \geq 0$, then for any desired $r > 0$,

[1]This observation follows from the inequality $J(y) \geq J(y^*) + \nabla J(y^*)(y - y^*) + \frac{\beta}{2}|y - y^*|^2$, which holds for all $y, y^* \in \mathbb{R}$ for a function

$\kappa_J(r) = \frac{1}{r}$. Intuitively, A4.3.2 (b) requires the function $y \mapsto J(y) - J(y^*)$ to be locally bounded from below by some (arbitrarily shallow) parabola, centered at y^*. Therefore, A4.3.2 (b) requires $J(\cdot)$ to be locally strongly convex, relaxing the global strong convexity assumption made in [91]. $\diamondsuit$

Each agent is assumed to have access only to the gradient of its individual cost function and its neighbors' estimates of y^*, thereby implementing the following protocol:

$$\dot{x}_i = \Big[c[A]_i x - \alpha \nabla J_i(x_i)\Big]^+_{x_i}, \quad x_i(0) \geq 0, \ \forall i \in \mathscr{V} \tag{4.4}$$

where $x_i \in \mathbb{R}$ is the ith agent's estimate of y^*, $x = [x_1, \ldots, x_N]^T$, $c \in \mathbb{R}_{++}$ and $\alpha \in \mathbb{R}_{++}$ are tuning parameters, and the logical projection operator $[q]^+_p$ is defined by

$$[q]^+_p = \begin{cases} 0, & \text{if } p \leq 0 \text{ and } q \leq 0 \\ q, & \text{otherwise,} \end{cases} \tag{4.5}$$

for all $q, p \in \mathbb{R}$. We say that the projection is *inactive* if $[q]^+_p = q$, and that it is *active* otherwise.

Aside from evolving in continuous-time, algorithm (4.4) differs from that presented in [114] in featuring the additional tuning parameter c, which can be used to emphasize the strength of the consensus term $c[A]_i x$ in relation to that of the optimization term $-\alpha \nabla J_i(x_i)$. An interesting outcome of our forthcoming analysis is that c provides a means of adjusting the ultimate collective precision of the agents' estimates of y^*, without affecting the convergence rate of their mean estimate. This observation is further discussed in the following two sections.

In the sequel, our objective is to demonstrate that for each $i \in \mathscr{V}$, $x_i(t)$ converges exponentially to y^*, to within an error whose magnitude can be made arbitrarily small through the tuning parameters c and α.

4.4 Analysis

In Chapter 3, we introduced a Lyapunov-based method for analyzing a discrete-time consensus optimization scheme. This analysis method is novel in treating the evolution of the mean and deviation variables associated with the agents' estimates as the interconnection of two dynamic subsystems whose isolated dynamics have favourable stability properties. The stability properties of these isolated subsystems are then exploited to derive conditions for the stability of their interconnection. In this chapter we build on that framework, demonstrating that it accommodates continuous-time dynamics and logical projection operations of the sort in (4.4).

We begin by rewriting (4.4) in a more convenient form. Defining the terms

$$\phi_i(x) = \Big[c[A]_i x - \alpha \nabla J_i(x_i)\Big]^+_{x_i} - \big(c[A]_i x - \alpha \nabla J_i(x_i)\big), \tag{4.6}$$

for all $i \in \mathscr{V}$, and

$$d(x) = [\nabla J_1(x_1), \ldots, \nabla J_N(x_N)]^T \tag{4.7}$$

allows us to write (4.4) as

$$\dot{x} = cAx - \alpha d(x) + \phi(x), \tag{4.8}$$

$J : \mathbb{R} \to \mathbb{R}$ that is strongly convex with constant β.

where $\phi(x) = [\phi_1(x), \ldots, \phi_N(x)]^T$. Next, we introduce the variable

$$y = \tfrac{1}{N}\mathbf{1}^T x, \tag{4.9}$$

representing the mean of the agents' estimates, and

$$z_i = x_i - y, \quad \forall i \in \mathscr{V} \tag{4.10}$$

representing the ith agent's deviation from the mean. The deviation vector $z = [z_1, \ldots, z_N]^T$ can equivalently be written as

$$z = Mx, \quad \text{where} \quad M = I - \tfrac{1}{N}\mathbf{1}\mathbf{1}^T. \tag{4.11}$$

Our aim now is to show that under the dynamics (4.8), the mean y converges to the minimizer y^*, while the deviations z_i converge to a neighbourhood of zero. To that end, we derive the dynamic equations governing the motion y and z. Since $\mathbf{1}^T A = \mathbf{0}^T$, we have

$$\dot{y} = -\tfrac{\alpha}{N}\mathbf{1}^T d(z+\mathbf{1}y) + \tfrac{1}{N}\phi(z+\mathbf{1}y). \tag{4.12}$$

By our definition of $d(x)$ in (4.7), $\mathbf{1}^T d(\mathbf{1}y) = \nabla J(y)$ and we may write (4.12) as

$$\begin{aligned} \dot{y} &= -\tfrac{\alpha}{N}\nabla J(y) + p_1(z,y) + \tfrac{1}{N}\mathbf{1}^T\phi(z+\mathbf{1}y), & (4.13)\\ &= \left[-\tfrac{\alpha}{N}\nabla J(y) + p_1(z,y) + \tfrac{1}{N}\mathbf{1}^T\phi(z+\mathbf{1}y)\right]_y^+ & (4.14) \end{aligned}$$

where

$$p_1(z,y) = \tfrac{\alpha}{N}\mathbf{1}^T\big(d(\mathbf{1}y) - d(z+\mathbf{1}y)\big). \tag{4.15}$$

It is useful to note that (4.13) is identical to (4.14), since $y(t) = \frac{1}{N}\sum_{i=1}^N x_i(t)$ is non-negative for all time due to the positive projections in (4.4). The non-negativity of y ensures that the projection applied in (4.14) is never active.

Next, we examine the motion of z. Noting that $MA = A$ since $\mathbf{1}^T A = \mathbf{0}^T$ and that $Ax = Az$ since $z = x + \mathbf{1}y$, from (4.11) and (4.8) we obtain

$$\begin{aligned} \dot{z} &= cMAx - \alpha M d(x) + M\phi(x) \\ &= cAz - \alpha M d(z+\mathbf{1}y) + M\phi(z+\mathbf{1}y). & (4.16) \end{aligned}$$

For later convenience we express (4.16) as

$$\dot{z} = cAz - \alpha M d(\mathbf{1}y^*) + p_2(z,y) + p_3(z,y) + M\phi(z+\mathbf{1}y), \tag{4.17}$$

where

$$p_2(z,y) = -\alpha M\big(d(z+\mathbf{1}y) - d(\mathbf{1}y)\big) \tag{4.18}$$

and

$$p_3(z,y) = -\alpha M\big(d(\mathbf{1}y) - d(\mathbf{1}y^*)\big). \tag{4.19}$$

For this subsystem, it is useful to note that for all $t \in \mathbb{R}_+$, $z(t) \in \operatorname{span}\{\mathbf{1}\}^\perp$. To see this, we note that $\mathbf{1}^T z = \mathbf{1}^T Mx = \mathbf{0}^T$, based on the definition of M in (4.11).

Equations (4.14) and (4.17) describe a feedback interconnection of two dynamic subsystems, whose interconnection structure is captured by the terms $p_i(z,y)$, $i = 1,2,3$, and the terms arising from the projection operation. Our analysis is based on treating these interconnection terms as perturbations to the isolated subsystem dynamics, which are given by the equations

$$\dot{\eta} = \big[-\tfrac{\alpha}{N}\nabla J(\eta)\big]^+_\eta, \quad \eta(0) \geq 0 \tag{4.20}$$

$$\dot{\zeta} = cA\zeta - \alpha M d(\mathbf{1}y^*), \quad \zeta(0) \in \operatorname{span}\{\mathbf{1}\}^\perp, \tag{4.21}$$

by setting to zero all interconnection terms in (4.14) and (4.17). In the following two Lemmas, we study the stability properties of y^* for (4.20), and $\mathbf{0}$ for (4.21).

Lemma 4.4.1: Suppose that A4.3.2 (a) holds. Then, $V_y(\eta) = \frac{1}{2}(\eta - y^*)^2$ is a Lyapunov function for (4.20), and its derivative along any trajectory of (4.20) satisfies

$$\dot{V}_y(\eta) \leq \tfrac{\alpha}{N}\big(J(y^*) - J(\eta)\big). \tag{4.22}$$

If in addition A4.3.2 (b) holds, then

$$\dot{V}_y(\eta) \leq -2\kappa_J(r)\tfrac{\alpha}{N}V_y(\eta), \tag{4.23}$$

whenever $\eta(0) \in \bar{B}_r(y^*) \cap \mathbb{R}_+$.

Proof. First we introduce the term

$$\phi_o(\eta) = \big[-\tfrac{\alpha}{N}\nabla J(\eta)\big]^+_\eta + \tfrac{\alpha}{N}\nabla J(\eta)$$

and re-write (4.20) as

$$\dot{\eta} = -\tfrac{\alpha}{N}\nabla J(\eta) + \phi_o(\eta).$$

Then

$$\dot{V}_y(\eta) = -\tfrac{\alpha}{N}(\eta - y^*)\nabla J(\eta) + (\eta - y^*)\phi_o(\eta)$$

Next, we note that $\phi_o(\eta) \geq 0$, for all $\eta \in \mathbb{R}$. Consequently, when $\eta \leq y^*$, the term $(\eta - y^*)\phi_o(\eta) \leq 0$. On the other hand when $\eta > y^* \geq 0$, $\phi_o(\eta) = 0$, since the projection is inactive when $\eta > 0$. Thus we see that for all $\eta \in \mathbb{R}$,

$$\dot{V}_y(\eta) \leq -\tfrac{\alpha}{N}(\eta - y^*)\nabla J(\eta). \tag{4.24}$$

By A4.3.2 (a), J is convex, implying that the following inequality holds for all $a, b \in \mathbb{R}$:

$$J(a) \geq J(b) + \nabla J(b)\cdot(a - b). \tag{4.25}$$

Relation (4.22) then follows by taking $a = y^*$, $b = \eta$, and rearranging (4.25).

If assumption A4.3.2 (b) additionally holds and (4.20) is initialized inside $\bar{B}_r(y^*) \cap \mathbb{R}_+$, then (4.22) becomes

$$\begin{aligned}\dot{V}_y(\eta) &\leq -\kappa_J(r)\tfrac{\alpha}{N}|\eta - y^*|^2 \\ &\leq -2\kappa_J(r)\tfrac{\alpha}{N}V_y(\eta),\end{aligned} \tag{4.26}$$

which was to be shown. ■

Remark 4.4.1: By (4.2), relation (4.22) implies that y^* is globally asymptotically stable for (4.20) whenever J satisfies A4.3.2 (a), and exponentially stable if A4.3.2 (b) also holds [80]. If $\bar{r} = \infty$ and $\liminf_{r\to\bar{r}} \kappa_J(r) > 0$, then under A4.3.2 (a) and (b), y^* is globally exponentially stable for (4.20); such is the case if, for example, J is strongly convex (c.f. Remark 4.3.1). ◇

Lemma 4.4.2: Suppose that A4.3.1 holds. Then, $V_z(\zeta) = \frac{1}{2N}\zeta^T\zeta$ is a Lyapunov function for (4.21), and its derivative along any trajectory of (4.21) satisfies

$$\dot{V}_z(\zeta) \leq -\tfrac{3c\lambda_2}{2} V_z(\zeta) + K \tag{4.27}$$

where

$$K = \tfrac{\alpha^2}{c\lambda_2 N}\|d(\mathbf{1}y^*)\|^2, \tag{4.28}$$

and λ_2 is the Fiedler eigenvalue of $-A$.

Proof. The time-derivative of V_z is

$$\begin{aligned}\dot{V}_z(\zeta) &= \nabla V_z(\zeta)^T\dot{\zeta}\\ &= \tfrac{c}{N}\zeta^T A\zeta - \alpha\zeta^T M d(\mathbf{1}y^*).\end{aligned} \tag{4.29}$$

We would like to bound the first term in the above expression as $\zeta^T A\zeta \leq -\lambda_2\|\zeta\|^2$. However, this inequality only holds if $\mathrm{span}\{\mathbf{1}\}^\perp$ is invariant for (4.21). To see that this is indeed the case, we note that the velocity of the state ζ projected onto $\mathrm{span}\{\mathbf{1}\}$ is zero — i.e.,

$$\mathbf{1}^T\dot{\zeta} = c\mathbf{1}^T A\zeta - \alpha\mathbf{1}^T M d(\mathbf{1}y^*) = \mathbf{0}^T + \mathbf{0}^T. \tag{4.30}$$

In other words, (4.21) generates no motion along $\mathrm{span}\{\mathbf{1}\}$, and initializing these dynamics on $\mathrm{span}\{\mathbf{1}\}^\perp$ implies that $\zeta(t)$ remains there forever. Next, noting that $\|M\| \leq 1$ we obtain

$$\begin{aligned}\dot{V}_z(\zeta) &\leq -\tfrac{c\lambda_2}{N}\|\zeta\|^2 + \alpha\|\zeta\|\cdot\|d(\mathbf{1}y^*)\|\\ &= -\tfrac{3}{4}\tfrac{c\lambda_2}{N}\|\zeta\|^2 - \tfrac{c\lambda_2}{4N}\left(\|\zeta\| - \tfrac{2\alpha}{c\lambda_2}\|d(\mathbf{1}y^*)\|\right)^2 + K\\ &\leq -\tfrac{3c\lambda_2}{2}V_z(\zeta) + K,\end{aligned}$$

as desired. ■

Remark 4.4.2: From relation (4.27), we can show that there exists a number $\delta(K) > 0$, such that for any $\zeta(0) \in \mathrm{span}\{\mathbf{1}\}^\perp$, $\zeta(t)$ approaches the set $\bar{B}_{\delta(K)}(\mathbf{0})$ exponentially fast and enters it in finite time. Moreover, $\delta(K)$ decreases as K does. ◇

In the following Lemma we quantify the destabilizing effect that the interconnection terms have on the coupled system (4.13)–(4.17). Most interesting is the analysis of the positive projection terms (Lemma 4.4.3 (d)).

Lemma 4.4.3: Suppose that A4.3.2 (c) holds, and consider the functions $V_y(y) = \frac{1}{2}(y-y^*)^2$ and $V_z(z) = \frac{1}{2N}z^T z$, where $y \in \mathbb{R}$ and $z \in \mathbb{R}^N$. Then, the following relations hold:

(a) $\nabla V_y(y)\cdot p_1(z,y) \leq \frac{\alpha L}{\sqrt{N}}|y-y^*|\cdot\|z\|$

(b) $\nabla V_z(z)^T p_2(z,y) \leq \frac{\alpha L}{N} \|z\|^2$

(c) $\nabla V_z(z)^T p_3(z,y) \leq \frac{\alpha L}{\sqrt{N}} |y-y^*| \cdot \|z\|$

(d) $\nabla V_y(y) \cdot \left(\frac{1}{N} \mathbf{1}^T \phi(z+\mathbf{1}y)\right) + \nabla V_z(z)^T \left(M\phi(z+\mathbf{1}y)\right) \leq 0$

Where $L = \max_i L_i$, $p_i(z,y)$, $i = 1,2,3$ are as in (4.15), (4.18) and (4.19), while $\phi(z+\mathbf{1}y) = [\phi_1(z+\mathbf{1}y), \ldots, \phi_N(z+\mathbf{1}y)]^T$, with $\phi_i(z+\mathbf{1}y)$ as in (4.6).

Proof. (a) From (4.15) and (4.7) we have:

$$\begin{aligned}\nabla V_y(y) \cdot p_1(z,y) &= \tfrac{\alpha}{N}(y-y^*) \sum_{i=1}^{N} \left(\nabla J_i(y) - \nabla J_i(z_i+y)\right) \\ &\leq \tfrac{\alpha}{N} |y-y^*| \sum_{i=1}^{N} L_i |z_i|,\end{aligned}$$

by A4.3.2 (c). The required result follows by noting that $\sum_{i=1}^{N} |z_i| = \|z\|_1 \leq \sqrt{N}\|z\|$.

(b) From (4.18), (4.7) and the fact that $z^T M = z^T$ (since $z(t) \in \text{span}\{\mathbf{1}\}^{\perp}$, $\forall t \in \mathbb{R}_+$), we obtain

$$\begin{aligned}\nabla V_z(z)^T p_2(z,y) &= -\tfrac{\alpha}{N} \sum_{i=1}^{N} z_i \left(\nabla J_i(z_i+y) - \nabla J_i(y)\right) \\ &\leq \tfrac{\alpha L}{N} \sum_{i=1}^{N} |z_i|^2,\end{aligned}$$

which is the desired result.

(c) The proof of this statement is similar to that of parts (a) and (b).

(d) For convenience, we denote by P the left-hand side of the desired inequality. Using the fact that $z^T M = z^T$, we write:

$$P = \tfrac{1}{N}(y-y^*)\mathbf{1}^T \phi(z+\mathbf{1}y) + \tfrac{1}{N} z^T \phi(z+\mathbf{1}y). \tag{4.31}$$

Next, we let $\mathscr{A}$ denote the set of active constraints in algorithm (4.4) – that is,

$$\mathscr{A} = \{i \in \mathscr{V} : x_i = 0, \text{ and } c[A]_i x - \alpha \nabla J_i(0) \leq 0\}.$$

Clearly, $\forall i \notin \mathscr{A}$, $\phi_i(x) = 0$. Therefore, (4.31) can be written as

$$P = \tfrac{1}{N}\Big(\sum_{i \in \mathscr{A}} \left[(y-y^*)\phi_i(z+\mathbf{1}y) + z_i \phi_i(z+\mathbf{1}y)\right]\Big). \tag{4.32}$$

Recalling that $z_i = x_i - y$, and that $\forall i \in \mathscr{A}$, $x_i = 0$, we obtain

$$\begin{aligned}P &= \tfrac{1}{N}\Big(\sum_{i \in \mathscr{A}} \left[y\phi_i(z+\mathbf{1}y) - y\phi_i(z+\mathbf{1}y)\right] - y^* \sum_{i \in \mathscr{A}} \phi_i(z+\mathbf{1}y)\Big) \\ &\leq 0,\end{aligned}$$

since both y^* and $\sum_{i \in \mathscr{A}} \phi_i(z+\mathbf{1}y)$ are non-negative. ■

With the above Lemmata, we are ready to state our main result.

Theorem 4.4.1: Consider the algorithm (4.4). Suppose that A4.3.1 and A4.3.2 hold, that $\frac{1}{N}\mathbf{1}^T x(0) \in \bar{B}_r(y^*) \cap \mathbb{R}_+$, and that

$$\tfrac{\alpha}{c} \in \Big(0, \min\Big\{\tfrac{\lambda_2 \kappa_J(r)}{4L^2 N}, \tfrac{\lambda_2}{2L}\Big\}\Big). \tag{4.33}$$

Then, for all $t \geq 0$

$$\|x(t) - \mathbf{1}y^*\|^2 \leq 4Ne^{-\gamma t}\Big(V(z(0), y(0)) - \tfrac{K}{\gamma}\Big) + \tfrac{4NK}{\gamma}, \tag{4.34}$$

where

$$V(z(0), y(0)) = \tfrac{1}{N}\|z(0)\|^2 + \frac{1}{2}|y(0) - y^*|^2, \tag{4.35}$$

$$\gamma = \min\{2K_y, 2NK_z\}, \tag{4.36}$$

$$K_y = \alpha\Big(\tfrac{\kappa_J(r)}{N} - \tfrac{\alpha}{c}\tfrac{4L^2}{\lambda_2}\Big), \tag{4.37}$$

$$K_z = c\Big(\tfrac{\lambda_2}{2N} - \tfrac{\alpha}{c}\tfrac{L}{N}\Big), \tag{4.38}$$

and K is as in (4.28).

Proof. Consider the composite Lyapunov function candidate $V(z,y) = V_y(y) + V_z(z)$, where $V_y(y) = \frac{1}{2}(y - y^*)^2$ and $V_z(z) = \frac{1}{2N} z^T z$, with $y \in \mathbb{R}$ and $z \in \mathbb{R}^N$. First, we note that

$$\begin{aligned}
\|x - \mathbf{1}y^*\|^2 &= \|x - \mathbf{1}y + \mathbf{1}y - \mathbf{1}y^*\|^2 \\
&\leq 2\|z\|^2 + 2N|y - y^*|^2 \\
&\leq 4NV_z(z) + 4NV_y(y) \\
&= 4NV(z,y).
\end{aligned} \tag{4.39}$$

It now remains for us to establish the behaviour of $t \mapsto V(z(t), y(t))$.

The derivative of $V(z,y)$ along the trajectories of (4.13)-(4.17) is

$$\begin{aligned}
\dot{V}(z,y) &= \nabla V_y(y)\dot{y} + \nabla V_z(z)^T \dot{z} \\
&= \nabla V_y(y)\big(-\tfrac{\alpha}{N}\nabla J(y)\big) + \nabla V_z(z)^T\big(cAz - \alpha M d(\mathbf{1}y^*)\big) \\
&\quad + \nabla V_y(y)\cdot\big(\tfrac{1}{N}\mathbf{1}^T\phi(z + \mathbf{1}y)\big) + \nabla V_z(z)^T\big(M\phi(z + \mathbf{1}y)\big) \\
&\quad + \nabla V_y(y)\cdot p_1(z,y) + \nabla V_z(z)^T\big(p_2(z,y) + p_3(z,y)\big).
\end{aligned}$$

Combining the outcomes of Lemma 4.4.1, Lemma 4.4.2 and Lemma 4.4.3, we obtain

$$\begin{aligned}
\dot{V}(z,y) &\leq -\kappa_J(r)\tfrac{\alpha}{N}|y - y^*|^2 - \big(\tfrac{3}{4}\tfrac{c\lambda_2}{N} - \tfrac{\alpha L}{N}\big)\|z\|^2 + K \\
&\quad + \tfrac{2\alpha L}{\sqrt{N}}|y - y^*|\|z\| \\
&\leq -K_y|y - y^*|^2 - K_z\|z\|^2 + K,
\end{aligned}$$

where K_y, K_z and K are as in (4.37), (4.38) and (4.28), respectively.

From (4.37) and (4.38) we see that both $K_y > 0$ and $K_z > 0$ whenever (4.33) holds. Moreover, whenever (4.33) is true, we also have that

$$\dot{V}(z,y) \leq -\gamma V(z,y) + K, \tag{4.40}$$

with γ as in (4.36). Solving the differential inequality (4.40) yields

$$V(z(t),y(t)) \leq e^{-\gamma t}\left(V(z(0),y(0)) - \frac{K}{\gamma}\right) + \frac{K}{\gamma}. \tag{4.41}$$

Then, relation (4.39) leads to the required conclusion. ■

Remark 4.4.3: The ultimate upper bound on the agents' collective error in estimating y^* is

$$\begin{aligned}
&\lim_{t\to\infty} \|x(t) - \mathbf{1}y^*\|^2 \leq \frac{4NK}{\gamma} \\
&= \frac{\alpha^2 \|d(\mathbf{1}y^*)\|^2}{8c\lambda_2 \cdot \min\left\{\alpha\left(\frac{\kappa_J(r)}{N} - \frac{\alpha}{c}\frac{4L^2}{\lambda_2}\right), c\left(\frac{\lambda_2}{2} - \frac{\alpha}{c}L\right)\right\}} &(4.42)\\
&= \frac{(\frac{\alpha}{c})^2 \|d(\mathbf{1}y^*)\|^2}{8\lambda_2 \cdot \min\left\{\frac{\alpha}{c}\left(\frac{\kappa_J(r)}{N} - \frac{\alpha}{c}\frac{4L^2}{\lambda_2}\right), \left(\frac{\lambda_2}{2} - \frac{\alpha}{c}L\right)\right\}}. &(4.43)
\end{aligned}$$

We note that this error can be made arbitrarily small by choosing $\frac{\alpha}{c}$ to be sufficiently small. From (4.33) it is evident that the parameter c can aid in relaxing the upper bound on α imposed in our previous studies (c.f. Theorem 3.1, [91]). Moreover, here we are able to derive an exponential convergence result, whereas in previous work only asymptotic convergence is demonstrated. ♢

4.5 An Application to Cooperative Resource Allocation Problems

A potentially interesting application of algorithm (4.4) is in the design of fully decentralized resource allocation algorithms. For example, consider a set of N agents that need to "fairly", and completely allocate a limited resource amongst themselves, without necessarily having access to measurements of aggregate quantities such as the net usage of the resource. The notion of fairness is related to allocating the resource in a way that takes into account each agent's individual need for it, which, even in a cooperative setting, may not be practically broadcast to all other agents. For the sake of concreteness, suppose that the ith agent's incentive to acquire a portion of the resource is encoded in the utility function

$$U_i(x_i) = a_1 \ln(x_i + 1), \tag{4.44}$$

where $x_i \in \mathbb{R}_+$ is the quantity of the resource allocated to the ith agent, and a_i reflects its individual need for the resource. In addition to these individual objectives, the agents' collective objective is to

$$\begin{aligned}
\max_{x} \quad & \sum_{i=1}^{N} U_i(x_i) \\
\text{s.t.} \quad & x_i \geq 0, \ \forall i = 1,\ldots,N \\
& \mathbf{1}^T x \leq C,
\end{aligned} \tag{4.45}$$

where $x = [x_1,\ldots,x_N]^T$ and C is the total amount of the resource available to the agents.

A prototypical instantiation of this problem is congestion control in internet-style networks [79], where the resource vied for is the bandwidth of a bottleneck link with a capacity C. There are primal [79], dual [99] and primal-dual algorithms [48] available to solve problem (4.45). However in that setting the users are assumed to be competitive, and the capacity constraint is typically enforced by having a central coordinator – i.e. the link – set a price on the resource based on its total usage. This price is communicated to each agent, which then optimizes an augmented utility function including a pricing term.

We wish to consider an alternative setting that eliminates the need for a central coordinator. In its simplest incarnation, a fully decentralized solution to the problem (4.45) is an application of the algorithm (4.4) to solving the problem dual to (4.45).

To form the dual problem, we first consider the Lagrangian

$$\begin{aligned} L(x,\lambda) &= \sum_{i=1}^{N} U_i(x_i) - \lambda\left(\mathbf{1}^T x - C\right) \\ &= \sum_{i=1}^{N}\left(U_i(x_i) - \lambda x_i\right) + \lambda C, \end{aligned} \tag{4.46}$$

with $\lambda \in \mathbb{R}$. The associated dual function is defined as

$$D(\lambda) = \max_{x\in\mathbb{R}_+^N} L(x,\lambda). \tag{4.47}$$

For the set of utilities given in (4.44), we find

$$x_i^*(\lambda) = \arg\max_{x\in\mathbb{R}_+^N} L(x,\lambda) = \max\left\{0, \tfrac{a_i-\lambda}{\lambda}\right\}, \tag{4.48}$$

and

$$D(\lambda) = \sum_{i=1}^{N}\left(a_i \ln\left(\tfrac{a_i}{\lambda}\right) - a_i + \lambda\right) + \lambda C. \tag{4.49}$$

We separate the dual function $D(\lambda)$ by writing $D(\lambda) = \sum_{i=1}^N D_i(\lambda)$, where we let

$$D_1(\lambda) = a_1 \ln\left(\tfrac{a_1}{\lambda}\right) - a_1 + (1+C)\lambda, \tag{4.50}$$

and

$$D_i(\lambda) = a_i \ln\left(\tfrac{a_i}{\lambda}\right) - a_i + \lambda, \quad i = 2,\dots,N. \tag{4.51}$$

The dual problem then is to

$$\begin{aligned} \min_{\lambda} \quad & \sum_{i=1}^{N} D_i(\lambda) \\ \text{s.t.} \quad & \lambda \geq 0. \end{aligned} \tag{4.52}$$

This problem can be solved in a decentralized manner by having each of the N agents maintain an individual estimate of the optimal resource "pricing" parameter (i.e. Lagrange multiplier) λ^*, and exchange this estimate with neighboring agents using algorithm (4.4). In particular, each agent employs the dynamic resource usage law

$$\dot{\lambda}_i = \left[c[A]_i\bar{\lambda} - \alpha\nabla D_i(\lambda_i)\right]^+_{\lambda_i}, \quad \lambda_i(0) \in \mathbb{R}_+, \tag{4.53}$$

$$x_i = \max\left\{0, \tfrac{a_i-\lambda_i}{\lambda_i}\right\}, \tag{4.54}$$

where $\bar{\lambda} = [\lambda_1, \ldots, \lambda_N]^T$, and (4.54) maximizes the Lagrangian as in (4.48). In contrast with the well-known *distributed* dual approach presented in [99], algorithm (4.53)–(4.54) is fully *decentralized*; individual agents do not need to have access to a common resource pricing parameter λ. To implement (4.53)–(4.54), each agent needs to know the common parameters α and c, which it can obtain upon joining the network. If the structures and parameters of agents' typical utilities are known to belong to some class and there is a maximum number of agents allowed to join the network, it is possible for a system designer to stipulate the required α and c *a priori*. No agent in the network needs to be aware of how many others are joined (or when they join or leave), and only the first agent to join needs to know the total available quantity C of the resource.

For illustration, we suppose that there are $N = 3$ agents, communicating over a line graph such as the one in Figure 4.1.

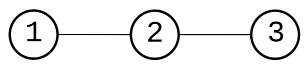

Figure 4.1: The agents' communication structure.

The corresponding A matrix to be used in (4.53) is

$$A = \begin{bmatrix} -1 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -1 \end{bmatrix}. \tag{4.55}$$

Taking the agents' utility weights to be $a_1 = 0.9$, $a_2 = 1.3$ and $a_3 = 2.1$, and the total amount of the resource to be $C = 5$ units, we obtain the optimal resource price $\lambda^* \approx 0.54$. The individual dual functions $D_i(\lambda)$ for this case are plotted in Figure 4.2, along with their sum $D(\lambda)$ and the value λ^*. We remark that from the plot of $D(\lambda)$ in Figure 4.2, it is evident that Assumption A4.3.2 (b) is satisfied for any $\bar{r} \in \mathbb{R}$, with a $\kappa_J(r)$ that decreases in $r \in (0, \bar{r})$.

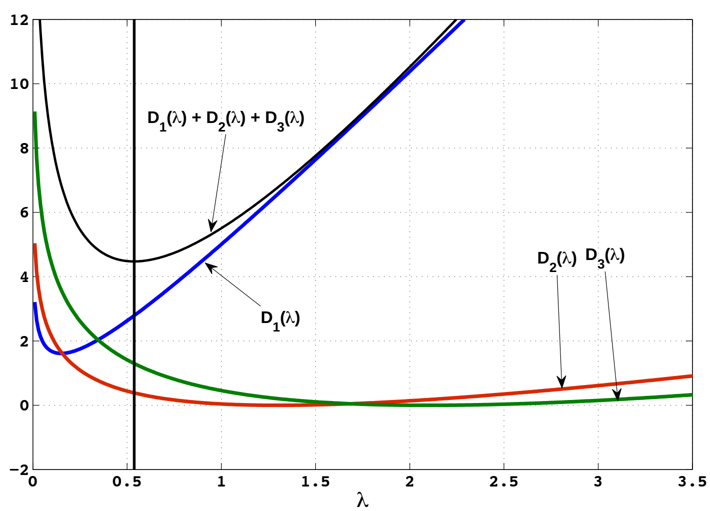

Figure 4.2: The individual dual functions and their sum, $D(\lambda)$, with a minimum at $\lambda \approx 0.54$.

In Figure 4.3, we show the outcomes of three simulations of the estimation process (4.53); in all three cases, the initial conditions are taken to be $(\lambda_1(0), \lambda_2(0), \lambda_3(0)) = (0.15, 1, 3)$, while the variations in the algorithm parameters c and α are indicated in the figure.

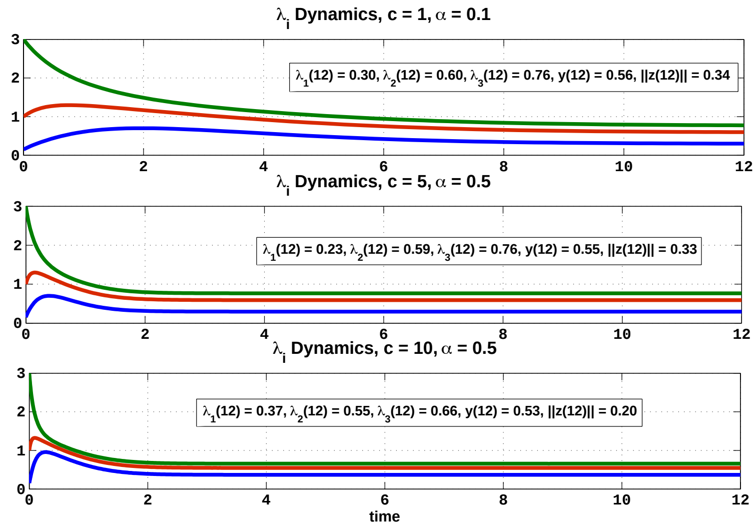

Figure 4.3: The dynamics of λ_i, $i = 1,2,3$, for three choices of algorithm parameters c and α. The final values of the λ_i, their mean y and their deviation $\|z\|$ are indicated on the graphs.

From the comparison of the first two plots in Figure 4.3 it is evident that increasing the parameter α increases the convergence rate of the mean of agents' estimates, while a comparison of the middle and bottom plots shows that increasing the value of c increases the convergence rate of the deviation error $\|z\|$ and reduces its ultimate magnitude, while having little effect on the convergence rate of the mean estimate. The effect of increasing α without increasing c is to increase the ultimate magnitude of the deviation error even though the mean of the agents' estimates converges faster.

Remark 4.5.1: We note that the dual costs $D_i(\lambda)$ in (6.1) and (4.51) do not satisfy A4.3.2 (c) at $\lambda = 0$. However, for all $i \in \mathscr{V}$ the assumption is satisfied on $[\varepsilon, \infty)$, for any arbitrarily small ε, with L_i depending on ε. All of the results derived here can be generalized to the case in which the gradients of the cost functions satisfy only a local Lipschitz condition – i.e., over some interval Ω containing y^*. The existence of an upper bound on $\frac{\alpha}{c}$ guaranteeing exponential stability would remain unchanged. However, the upper bound itself would depend on Ω and the algorithm would need to be initialized inside Ω^N. $\diamondsuit$

4.6 A Potential Application to Behavioral Modelling of Animal Groups

There are some interesting potential applications of continuous-time consensus optimization dynamics to the modelling of collective behavior of animal groups. In fact, the interest within the control community in pure consensus dynamics over general connected graphs seems to have been initiated by the work of Vicek [158], who proposed a consensus-like model for the emergence of directional coherence in the movements of randomly oriented particles in a plane, and discussed applications of the model to the study of animal group behavior.

In [120], a more elaborate model is considered in order to study the emergence of leadership in migratory animal populations. For individual agents (i.e. animals in the group), the model captures a central tradoff between the *investment cost* involved in measuring environmental signals that indicate the true desired direction of migration, and the relative ease with which this direction can be estimated by observing the motions of one's neighbors.

The model involves N agents, each of whom may take social cues from neighboring agents on a fixed, but directed interaction graph. The migration portion of the model considered is given by the Itō stochastic differential equation

$$dx_i = -k_i^2(x_i-\mu)dt - (1-k_i)^2[L]_i x dt + K dW_i, \tag{4.56}$$

where $\mu \in \mathbb{R}$ is the true desired migration direction, x_i is agent i's the direction of motion, $k_i \in [0,1]$ indicates how much agent i is invested in measuring μ, L is the graph Laplacian of the interaction graph, dW_i is the Wiener increment, and K is related to the intensity of noise that affects agent i's measurements of both μ and its estimate from social cues. (q.v. §2 in [120]). Model (4.56) indicates that when $k_i = 0$, agent i does not measure μ and relies exclusively on social cues, while $k_i = 1$ indicates that agent i is a "leader" – i.e. he invests in measuring the environmental signal μ. Clearly, migration cannot happen if all agents are followers, and the authors are interested in the effect that L and $[k_1,\ldots,k_N]^T$ have on the group's ability to migrate.

In Theorem 1 of [120], the deterministic version of (4.56) is studied in order to determine the conditions on the graph Laplacian L and the investment gains k_i under which migration is possible. In other words, the conditions guaranteeing the stability of the point $x = [x_1,\ldots,x_N]^T = [\mu,\ldots,\mu]^T =: x^*$ for the dynamics

$$\dot{x}_i = -k_i^2(x_i-\mu) - (1-k_i)^2[L]_i x \tag{4.57}$$

are being sought. The derivation of these conditions relies heavily on the linearity of the term $-k_i^2(x_i-\mu)$.

With $J_i(x) = -\frac{1}{2}k_i^2(x_i-\mu)^2$, $A = -L$ and $c_i := (1-k_i)^2$, we note that (4.57) can be written as

$$\dot{x}_i = c_i[A]_i x - \nabla J_i(x), \tag{4.58}$$

which closely resembles the form for the unconstrained version of the CO dynamics (4.4) studied in this chapter. The model (4.58) differs from (4.4) in that A pertains to a directed graph and the gains c_i weight each row of A differently; consequently, our analytical framework does not directly apply to the study of (4.58). However, the resemblance of (4.58) to (4.4) suggests that adaptations accommodating these differences are likely possible. If that is the case, then the techniques presented in this chapter may become instrumental to deriving conditions under which group behaviors more complex than migration are possible, and they may help enable the study of more general behavioral models for which $\nabla J_i(\cdot)$ are not necessarily linear.

4.7 Final Remarks

In this chapter we analyzed a continuous-time, consensus-based multi-agent optimization scheme by means of Lyapunov techniques. Our analysis accommodates collective optimization problems with positivity constraints. We derived explicit convergence rate and ultimate error bounds, and conditions guaranteeing the practical exponential stability of the collective optimum. Moreover, we showed that introducing an additional tuning parameter, which can be used to emphasize the strength of the consensus component relative to that of the optimization component, can improve the performance of the algorithm and relax the conditions guaranteeing its stability. Whereas such schemes typically find applications in sensor fusion problems, we observe that another particularly appropriate application of the scheme is in solving dual optimization problems in a fully decentralized manner.

Chapter 5

Decentralized Extremum-Seeking Control Based on Consensus Optimization

5.1 Synopsis

In this chapter, we show how the analytic framework developed in Chapter 3 enables the study of systems involving the dynamic interaction between discrete-time CO algorithms, and continuous-time dynamical systems. We address the class of discrete-time, dynamic DCCPs represented by Problem 1.2.2, and we propose that a solution may be based on a variant of a consensus optimization (CO) scheme. By means of Lyapunov and small-gain techniques we derive convergence conditions for the sampled-data feedback interconnection of a numerical CO-based decision updating process and the agent dynamics, which are assumed to evolve in continuous time. Under a minor extension, the results presented here can be viewed as a proposal for a novel multivariable extremum-seeking scheme. Alternatively, our consideration of continuous-time dynamics in the problem can be viewed as a generalization of the problem settings considered in existing literature on CO, which is typically focused on optimization of static maps.

The idea of combining consensus optimization methods with extremum seeking techniques originates in [93]. However, the problem setting therein is somewhat different from that addressed here; the results presented in this chapter constitute unpublished research.

5.2 Introduction

We consider the discrete-time, dynamic decentralized coordination control problem (DT-DDCCP) described in §1.2.3, and we propose a CO-based solution of the form

$$\mathscr{D}_i : \begin{cases} \xi_i(t_{k+1}) = \sum_{j=1}^{N} [A]_{i,j} \nu_j - \alpha \hat{s}_i(Y_i(t_{k+1}^-)), \\ \nu_i(t_k) = \xi_i(t_k) \\ u_i(t) = \sum_{j \in \mathscr{V}} [A]_{i,j} \xi_{j,i}(t_k), \quad \forall t \in [t_k, t_{k+1}), \ \forall k \in \mathbb{N} \end{cases} \tag{5.1}$$

where $Y_i(\cdot)$ is a set of measurements of the plant output y_i made within the iteration interval $[t_k, t_{k+1})$, and $\hat{s}_i(Y_i(t_{k+1}^-))$ is an approximation of the search direction $s_i(\cdot)$ in the standard CO algorithm (1.33). To the best of our knowledge, a solution of the form (5.1) to the DT-DDCCP described in Problem 1.2.2 has not been previously proposed in the literature.

We study the case in which the interval $T = t_{k+1} - t_k$ between iterations is constant and the approximate search direction is given by

$$\hat{s}_i(Y_i(t_{k+1}^-)) = y_i(t_{k+1}^-), \tag{5.2}$$

where y_i relates to an agent i's private cost function as follows:

$$\lim_{T\to\infty} y_i(t_{k+1}^-) = \nabla_u J_i(l(u)), \tag{5.3}$$

where $l(\cdot)$ is the equilibrium map associated with the collective dynamics (1.8). Studying this case is a preliminary step toward analysing the more practical situation in which only cost measurements are available – i.e., $y_i(t) = J_i(x(t))$ – and the gradient $\nabla_u J_i(l(u))$ must be estimated from a sequence of such measurements within each iteration of (5.1) (q.v. Example 1.4.3, in which a simple forward-difference approximation scheme is proposed.).

Our objective is to show how one may derive conditions on the iteration period T and the step-size α in (5.1) so as to guarantee the stable behavior of the sampled-data interconnection between $\mathscr{D}_i$ and the agent dynamics Σ_i in Problem 1.2.2. Such conditions are of interest because they reveal important relationships and trade-offs between relevant problem parameters, and performance criteria such as stability and convergence rate. For example, intuition suggests that under condition P4 in §1.2.1, increasing the iteration period T results in better approximations $\hat{s}_i(\cdot)$ to the idealized search direction $s_i(\cdot)$ in (1.33), so that each iteration of (5.1) is more effective. However, increasing T also means that the overall optimization process requires more time. For specific problems, the derived conditions may suggest how best to select the tunable parameters T and α.

The study of DT-DDCCPs with solutions such as (5.1) is motivated by several practical engineering applications. One potential application relates to the optimal coordination of channel powers in meshed optical networks. In this application, individual optical channels may be abstracted as agents that seek to maximize their individual optical signal-to-noise ratios (OSNR). An increase in one channel's power on a single link improves its individual OSNR, but tends to degrade the OSNRs of all other channels sharing the link; in other words, channels' private performance measures are coupled, and achieving a collectively optimal channel power configuration may require coordination. The continuous-time dynamics in this application arise from the physical properties of the optical links in the network [147]. These links are comprised of several spans of optical fiber and amplifiers which exhibit fast, stable dynamics. Existing coordination control schemes typically focus on point-to-point network topologies, appeal to game theoretic methods (q.v. Remarks 1.2.2 and 1.4.1) and ignore the effects of amplifier dynamics, [122].

Another potential application for our framework relates to the secondary control of islanded microgrids [101] (q.v. Remark 1.1.1). While the primary control scheme is responsible for regulating the behavior of individual loads and microgeneration units [78], the role of secondary control is to coordinate power flows within the microgrid and to regulate the frequencies and voltages at its point of common coupling [61]. The primary control of microgrids may be identified with the "inner loop" control discussed in §1.2.1; in terms of our problem setting, its goal is to stabilize the equilibrium map $l(u)$ (q.v. P4). The secondary control may then be identified with the decision updating process $\mathscr{D}_i$. Our proposed coordination method seems especially suitable for this application for two additional reasons. First, in previous research, the coordination objective has in fact been expressed as a sum of convex potential functions [101]. Second, even within the more general context of active power and frequency regulation in a large-scale power grid, the notion of *primary control* is associated with local, fast time-scale feedback control, while secondary control is associated with a slower time-scale feedback control over a wider geographical region. This time-scale separation between the (assumed) inner-loop control and the decision updating processes $\mathscr{D}_i$ is also inherent in our proposed scheme.

Though we are interested in decentralized coordination control for dynamic multiagent systems, for the case in which $y_i(t) = J_i(x(t))$, the proposed scheme may more aptly be regarded as a *decentralized, multivariable extremum-seeking* scheme. Consequently, relevant literature includes [86], [7], [171], [125], [43], [55], [56], [144], [45] and [150]. References [86], [7], [171], [125] and [150] focus on centralized extremum-seeking control (ESC) schemes, involving a single plant and a single ESC controller. References [55], [56], [45] and [144] involve multiagent extremum seeking, but they rely on game-theoretic methods and consider only restricted classes of agent dynamics. With the exception of [150] and [125], all references consider continuous-time ESC schemes which are notoriously difficult to tune.

Relative to this body of literature, our contributions in [93], [90] (see also [92]) and this chapter can therefore be regarded as a proposal for a novel decentralized, cooperative multiagent ESC scheme that operates in discrete-time and locates the socially optimal minimizer of the sum of agents' individual cost functions. On the other hand, relative to the literature on distributed optimization, which is reviewed in Chapter 3, our contribution here is to generalize the problem setting from that involving optimization of static maps, to that involving optimization of dynamic maps.

5.3 Problem Setting

As in Problem 1.2.2, we consider a set of N agents that may communicate over a connected communication graph $\mathscr{G}_C = (\mathscr{V}, \mathscr{E}_C)$, where $\mathscr{V} = \{1, \ldots, N\}$ is the set that indexes the agents, and $\mathscr{E}_C \subset \mathscr{V} \times \mathscr{V}$ indicates those agent pairs that can exchange information with one another at any time.

5.3.1 Agent Dynamics

The state $x_i \in \mathbb{R}^{n_i}$ of the i'th agent (or its proximal environment) evolves according to the dynamics

$$\Sigma_i : \begin{cases} \dot{x}_i = f_i(x, u_i), \\ y_i = h_i(x) \quad \forall i \in \mathscr{V}, \end{cases} \tag{5.4}$$

where $x = [x_1^T, \ldots, x_N^T]^T \in \mathbb{R}^n$ denotes the collective state, $f_i : \mathbb{R}^n \times \mathbb{R}^{m_i} \to \mathbb{R}^{n_i}$ and $h_i : \mathbb{R}^n \to \mathbb{R}^{r_i}$ are smooth functions, $y_i \in \mathbb{R}^{r_i}$ represents the collection of measurements that agent i can take at any time, and $u_i \in \mathbb{R}^{m_i}$ are its *decision*, or *action variables*.

As in §1.2.3, we let $u = [u_1^T, \ldots, u_N^T]^T \in \mathbb{R}^m$ denote the collective decision, and Σ the collective dynamics – i.e.,

$$\Sigma : \begin{cases} \dot{x} = f(x, u) \\ y = h(x), \end{cases} \tag{5.5}$$

where

$$f(x, u) = \begin{bmatrix} f_1(x, u_1) \\ \vdots \\ f_N(x, u_N) \end{bmatrix}, \tag{5.6}$$

and

$$h(x) = \begin{bmatrix} h_1(x) \\ \vdots \\ h_N(x) \end{bmatrix}. \tag{5.7}$$

We formalize property P4 in §1.2.3 by the following two assumptions.

A5.3.1 (Existence of a C^0 Equilibrium Map)**:** There exists an *equilibrium map* $l \in C^0[\mathbb{R}^m, \mathbb{R}^n]$ associated to Σ, so that $f(x,u) = 0$ if, and only if $x = l(u)$. $\diamondsuit$

An example of an equilibrium map for two double-integrator systems is given in Example 1.2.2.

Next, we introduce the transient error variable $z \in \mathbb{R}^n$, which we define as

$$z(t) = x(t) - l(u(t)), \quad \forall t \in \mathbb{R}_+,\ \forall u \in \mathbb{R}^m. \tag{5.8}$$

Whenever $u(t)$ is constant, the dynamics of the transient error are given by

$$\Sigma' : \begin{cases} \dot{z} = f(z + l(u), u), \\ y = h(z + l(u)) \end{cases} \tag{5.9}$$

where $f(\cdot,\cdot)$ and $h(\cdot)$ are as in (5.6) and (5.7). We make the following assumption concerning the transient error dynamics.

A5.3.2 (Asymptotic Stability of Σ')**:** For any fixed $u \in \mathbb{R}^m$, the point $z = 0$ is globally asymptotically stable for Σ'. $\diamondsuit$

Assumption A5.3.1 implies that for any fixed $u \in \mathbb{R}^m$, $\lim_{t\to\infty} x(t) = l(u)$.

Example 5.3.1 (Asymptotic Stability of Σ')**:** As in Example 1.2.2, consider a two-agent system in which the state of the ith agent evolves according to

$$\Sigma_i : \begin{cases} \dot{x}_{i,1} = x_{i,2} \\ \dot{x}_{i,2} = -x_{i,1} - k_i x_{i,2} + u_i \\ y_i = h_i(x), \end{cases} \tag{5.10}$$

where $\forall i \in \mathscr{V} = \{1,2\}$, $k_i \in \mathbb{R}_{++}$. In this case the transient error dynamics are given by

$$\Sigma' : \begin{cases} \begin{bmatrix} \dot{z}_{1,1} \\ \dot{z}_{1,2} \\ \dot{z}_{2,1} \\ \dot{z}_{2,2} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & -k_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & -k_2 \end{bmatrix} \\ y = h(z + l(u)) \end{cases} \tag{5.11}$$

for which A5.3.2 is clearly satisfied. $\diamondsuit$

5.3.2 The Coordination Objective

In accordance with P5 in §1.2.2, we consider problem scenarios in which the agents' collective task is to achieve some desirable collective state configuration $x^* \in X^* \subset \mathbb{R}^n$. We assume that this *goal set* X^* is specified as the solution to some unconstrained optimization problem over the set of Σ's equilibrium states $l(\mathbb{R}^m)$ – i.e.,

$$X^* = l(U^*), \tag{5.12}$$

where

$$U^* = \arg\min_{u \in \mathbb{R}^m} \sum_{i \in \mathscr{V}} J_i(l(u)), \tag{5.13}$$

$l(\cdot)$ is the equilibrium map associated with Σ, and $J_i : \mathbb{R}^n \to \mathbb{R}$ is a function whose values may be regarded as a measure of agent i's individual performance. We refer to

$$J(x) = \sum_{i \in \mathscr{V}} J_i(x) \tag{5.14}$$

as the *collective cost*, and we make the following assumption.

A5.3.3 (Convexity of the Collective Cost and Compactness of U^*)**:** The composition $J(l(u))$ is convex in u, and its set of minimizers U^* is nonempty and compact. $\diamondsuit$

We assume that the measured variables y_i relate to the agents' individual performance measures as follows.

A5.3.4 (Gradient-Related Measurements)**:** For all $i \in \mathscr{V}$, the composition $J_i(l(u))$ is differentiable at each $u \in \mathbb{R}^m$, and the functions $h_i(\cdot)$ in (5.4) are such that

$$h_i(l(u)) = \nabla_u J_i(l(u)). \tag{5.15}$$

$\diamondsuit$

The assumed continuity of $h_i(\cdot)$ implies that

$$\lim_{t \to \infty} h(x(t)) = h\big(\lim_{t \to \infty} x(t)\big). \tag{5.16}$$

Therefore, A5.3.1, A5.3.2 and A5.3.4 imply that for any fixed $u \in \mathbb{R}^m$,

$$\lim_{t \to \infty} y_i(t) = \lim_{t \to \infty} h_i(x(t)) = \nabla_u J_i(l(u)). \tag{5.17}$$

5.3.3 The Decentralized Coordination Control Algorithm

Each agent $i \in \mathscr{V}$ updates its decision u_i according to a CO-like rule given by

$$\mathscr{D}_i : \begin{cases} \xi_i(t_{k+1}) = \sum_{j=1}^{N} [A]_{i,j} \nu_j - \alpha \hat{s}_i(y_i(t_{k+1}^-)), \\ \nu_i(t_k) = \xi_i(t_k) \\ u_i(t) = \sum_{j \in \mathscr{V}} [A]_{i,j} \xi_{j,i}(t_k), \quad \forall t \in [t_k, t_{k+1}), \ \forall k \in \mathbb{N} \end{cases} \tag{5.18}$$

where $\forall k \in \mathbb{N}$, $t_{k+1} - t_k = T > 0$, $\xi_i = [\xi_{i,1}^T, \ldots, \xi_{i,N}^T]^T \in \mathbb{R}^m$, $\xi_{i,j} \in \mathbb{R}^{m_j}$, $A \in \mathbb{R}^{N\times N}$ is a weighted adjacency matrix associated to $\mathscr{G}_C$, and $\hat{s}_i(\cdot)$ denotes a single measurement of y_i taken just prior to the beginning of each iteration – i.e.,

$$\hat{s}_i(y_i(t_{k+1}^-)) = y_i(t_{k+1}^-). \tag{5.19}$$

From (5.17) and the fact that $u_i(t)$ is held constant within every iteration interval $[t_k, t_{k+1})$, we observe that A5.3.1, A5.3.2 and A5.3.4 imply that

$$\lim_{T\to\infty} \hat{s}_i(y_i(t_{k+1}^-)) = \nabla_u J_i(l(u))|_{u=u(t_k)}. \tag{5.20}$$

We define $s_i : \mathbb{R}^m \to \mathbb{R}^m$ as the steady-state value of $\hat{s}_i(\cdot)$, for any fixed $u \in \mathbb{R}^m$. Under A5.3.1, A5.3.2 and A5.3.4, we have that

$$\begin{aligned} s_i(u) &= h_i(l(u)) \\ &= \nabla_u J_i(l(u)). \end{aligned} \tag{5.21}$$

We assume that the matrix A in (5.18) satisfies condition A3.3.1, and we denote the second smallest eigenvalue of its square as

$$\kappa = \lambda_2(A^2). \tag{5.22}$$

Written compactly, (5.18) reads as

$$\mathscr{D} : \begin{cases} \xi(t_{k+1}) = (A \otimes I_m)\xi(t_k) - \alpha\hat{s}(h(x(t_{k+1}^-))) \\ u(t) = E(A\otimes I_m)\xi(t_k), \quad \forall t \in [t_k, t_{k+1}),\ \forall k \in \mathbb{N}, \end{cases} \tag{5.23}$$

where $\xi = [\xi_1^T, \ldots, \xi_N^T]^T \in \mathbb{R}^{Nm}$, $\hat{s}(h(x(t_{k+1}^-))) = [\hat{s}_1(y_1(t_{k+1}^-))^T, \ldots, \hat{s}_N(y_N(t_{k+1}^-))^T]^T$ and

$$E = \left[\, E_1 \,\vdots\, \cdots \,\vdots\, E_N \,\right] \in \mathbb{R}^{m\times Nm}, \tag{5.24}$$

with $E_i \in \mathbb{R}^{m\times m}$ being given by

$$E_i = \mathrm{blkdiag}(\mathbf{0}_{m_1}, \cdots, \mathbf{0}_{m_{i-1}}, I_{m_i}, \mathbf{0}_{m_{i+1}}, \cdots, \mathbf{0}_{m_N}), \quad \forall i \in \mathscr{V}. \tag{5.25}$$

Remark 5.3.1 (Contrasting Standard CO to $\mathscr{D}$)**:** In terms of the function $s_i(\cdot)$ and the composed (steady-state) costs $J_i(l(u))$, the standard (unconstrained) CO algorithm (1.33) is implemented as

$$\xi_i^+ = [A]_{i,j}\xi_j - \alpha s_i(\xi_i), \tag{5.26}$$

where $s_i(\cdot)$ is as in (5.21). This update rule is shown in [114] to solve the static problem

$$\min_{u\in\mathbb{R}^m} \sum_{i\in\mathscr{V}} J_i(l(u)) \tag{5.27}$$

whenever $s_i(\xi_i) \in \partial J_i(l(\xi_i))$. Alternatively, (1.33) may also be implemented as

$$\xi_i^+ = [A]_{i,j}\xi_j - \alpha s_i([A]_{i,j}\xi_j), \tag{5.28}$$

where $s_i([A]_{i,j}\xi_j) \in \partial J_i(l([A]_{i,j}\xi_j))$. The latter implementation appears in [116] and in Chapter 3 of this thesis, where we focus on static optimization problems.

In contrast to problem settings that standard CO is intended to address, in the decentralized coordination control context, agents may not have the freedom to evaluate the gradient of their private cost functions anywhere within the search space. Specifically, the quantities $s_i([A]_{i,j}\xi_j) \in \partial J_i(l([A]_{i,j}\xi_j))$ or $s_i(\xi_i) \in \partial J_i(l(\xi_i))$ may not be accessible to agent i.

Reinforcing our observations in Example 1.4.2, we note that $\mathscr{D}_i$ differs from the standard CO algorithm in two ways. First, the implemented search direction $\hat{s}_i(y_i(t_{k+1}^-))$ is an approximation of $s_i(u)$ in (5.21), and the quality of this approximation is degraded by the presence of the transient error $z(t)$. Namely, from (5.19) and (5.8),

$$\hat{s}_i(y_i(t_{k+1}^-)) = h_i(l(u)+z).$$

Second, even in the absence of the transient error dynamics (i.e. the case in which $\hat{s}_i(y(t_{k+1}^-)) = s_i(u_k)$), agent i may not have the freedom to evaluate the function $s_i(\cdot)$ at any arbitrary point within the search space because within the DCCP context (as in the game-theoretic context – q.v. Remarks 1.2.2 and 1.4.1), it is assumed that an agent cannot directly manipulate other agents' decision variables. In particular, agent i may not be able to evaluate $s_i(\cdot)$ at ξ_i (as in (5.26)) or at $[A]_{i,j}\xi_j$ (as in (5.28)). $\diamondsuit$

Since $h_i(\cdot)$ is presumed to be smooth and $l(\cdot)$ continuous, the composition $h_i(l(u))$ is also continuous, and therefore $s_i(\cdot)$ is locally Lipschitz continuous and satisfies A3.3.4. In particular, for every compact $\hat{\Omega}_\mu \subset \mathbb{R}^m$, and for every $i \in \mathscr{V}$, there exists a number $L_i \in \mathbb{R}_+$ such that for all $\mu, \nu \in \hat{\Omega}_\mu$,

$$\|s_i(\mu) - s_i(\nu)\| \leq L_i \|\mu - \nu\|. \tag{5.29}$$

Let

$$s_o(u) = \sum_{i \in \mathscr{V}} s_i(u). \tag{5.30}$$

By the linearity of the gradient operator, $s_o(u) = \nabla_u J(l(u))$. Consequently, the convexity of $u \mapsto J(l(u))$ (q.v. A5.3.3) and Remark 3.3.6 imply that $s_o(\cdot)$ satisfies the strict pseudogradient assumption A3.3.2, with $\phi(\mu) = 2(J(\cdot; d(t)) - J^*)$, for example. As in Remark 3.3.5, we note that A3.3.2 is equivalent to having the inequality

$$2 s_o(\mu)^T (\mu - \mathbf{P}_{U^*}(\mu)) \geq \phi(\mu) \tag{5.31}$$

hold for all μ in some given Ω_μ.

In the next section we establish that under assumptions A5.3.1 to A5.3.4 and A3.3.1, the decision updating processes $\mathscr{D}_i$ in (5.18) indeed solve Problem 1.2.2.

5.4 Analysis

Let

$$\Xi^* = \prod_{i \in \mathscr{V}} U^*,$$

where U^* is given in (5.13). Our objective in this section is to derive conditions under which the set $\{\mathbf{0}\} \times \Xi^* \subset \mathbb{R}^n \times \mathbb{R}^{Nm}$ is SPAS for the sampled-data feedback interconnection of the transient error dynamics Σ' in (5.9), and the collective decision updating process $\mathscr{D}$ in (5.23). Since the change of coordinates in (5.8) is smooth, showing that $\{\mathbf{0}\} \times \Xi^*$ is SPAS for $\Sigma' - \mathscr{D}$ is equivalent to showing that $l(U^*) \times \Xi^* = X^* \times \Xi^*$ is SPAS for $\Sigma - \mathscr{D}$. Therefore, to demonstrate that $\{\mathbf{0}\} \times \Xi^*$ is SPAS for $\Sigma' - \mathscr{D}$, is to establish that $\mathscr{D}_i$ in (5.18) solves Problem 1.2.2.

The key tool in establishing the SPAS of $\{\mathbf{0}\} \times \Xi^*$ for $\Sigma' - \mathscr{D}$ is Theorem 2.5.1, and we apply it as follows. We derive a set of small-gain conditions in Theorem 5.5.1, and we show in Theorem 5.5.3 that these conditions suffice to guarantee that the conditions of the SPAS Theorem 2.5.1 are satisfied.

We arrive at the derivation of the small-gain conditions in Theorem 5.5.1 in several stages by means of Lemmas 5.4.1 to 5.4.5. Lemma 5.4.1 quantifies the effect that the dynamic coupling between Σ' and $\mathscr{D}$ has on the evolution of Σ', the transient error dynamics. Next, as in Chapter 3, we break down the analysis of the decision updating processes $\mathscr{D}_i$ in (5.18) into three stages. Lemma 5.4.2 examines the evolution of the agents' mean estimate, Lemma 5.4.3 examines the evolution of the vector of deviations from the mean, and Lemma 5.4.4 combines the results of these two lemmas. In analogy to Lemma 3.6.3, in Lemma 5.4.4 it is observed that certain interconnection terms arising as perturbations to the "idealized" mean subsystem dynamics actually help negate the destabilizing effects of certain interconnection terms arising as perturbations to the "idealized" deviation subsystem dynamics. Finally, Lemma 5.4.5 combines the analyses of Lemmas 5.4.1 to 5.4.4, in preparation for the proof of Theorem 5.5.1.

Remark 5.4.1: Throughout this chapter we construct several sets to which we repeatedly make reference. For convenience, we collect their definitions here.

- Let $\Omega_\xi \subset \mathbb{R}^{Nm}$ be an arbitrarily large compact set containing $\prod_{i\in\mathscr{V}} U^*$.
- Define $\Omega_u = \{u \in \mathbb{R}^m \mid u = E(A\otimes I_m)\xi,\ \xi \in \Omega_\xi\}$, with E as in (5.24).
- Let $\Omega_x \subset \mathbb{R}^n$ be an arbitrarily large compact set containing $l(\Omega_u)$, with $l(\cdot)$ as in A5.3.1.
- Define $\hat{\Omega}_x = \{x \in \mathbb{R}^n \mid x = \Phi(t,x_o),\ x_o \in \Omega_x,\ t\in[0,T]\}$, where $\Phi \in C^1[\mathbb{R}_+ \times \mathbb{R}^n, \mathbb{R}^n]$ is the state transition function associated to Σ, and $T = t_{k+1} - t_k$.
- Define $\hat{\Omega}_u = \Omega_u \cup \{u^+ \in \mathbb{R}^m \mid u^+ = E(A^2\otimes I_m)\xi - \alpha E(A\otimes I_m)h(x),\ \xi\in\Omega_\xi,\ x\in\hat{\Omega}_x\}$, where $h(x) = [h_1(x)^T,\ldots,h_N(x)^T]^T$.
- Define $\Omega_\mu = \{\mu \in \mathbb{R}^m \mid \mu = \frac{1}{N}(\mathbf{1}_N^T\otimes I_m)\xi,\ \xi\in\Omega_\xi\}$.
- Define $\hat{\Omega}_\mu = \hat{\Omega}_u \cup \{\mu \in \mathbb{R}^m \mid \mu = \frac{1}{N}(\mathbf{1}_N^T\otimes I_m)\xi,\ \xi\in\Omega_\xi\}$
- Define $\Omega_\zeta = \{\zeta \in \mathbb{R}^{Nm} \mid \zeta = M\xi,\ \xi\in\Omega_\xi\}$, where $M = I_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T$.
- Define $\Omega_z = \{z\in\mathbb{R}^n \mid z = x - l(u),\ x\in\hat{\Omega}_x,\ u\in\Omega_u\}$.

$\diamondsuit$

5.4.1 The Closed-Loop Evolution of the Transient Error

We begin by considering how the evolution of the transient variable z, defined in (5.8), is affected by the presence of the $\mathscr{D}_i$. Within each iteration, $\mathscr{D}_i$ produces a constant control input $u(t_k)$. Consequently, the dynamics of z are given by (5.9), over every interval $[t_k, t_{k+1})$, $k\in\mathbb{N}$. We are interested in the magnitude of $z(t_{k+1}^-)$, and the evolution of the sequence $(z(t_{k+1}^-))_{k=1}^\infty$, since this quantity degrades the approximation $\hat{s}_i(y_i(t_{k+1}^-))$ of the search direction $s_i(u(t_k))$ at each iteration of $\mathscr{D}_i$ (q.v. Remark 5.3.1)

We adopt the following notational convention.

$$z := z(t_{k+1}^-)$$
$$z^+ := z(t_{k+2}^-) \tag{5.32}$$

and

$$q := q(t_k)$$
$$q^+ := q(t_{k+1}), \tag{5.33}$$

where q stands for any other quantity, including ξ_i, u, and $\hat{s}_i(\cdot)$.

Lemma 5.4.1: Consider the system $\Sigma - \mathscr{D}$ and suppose that conditions A5.3.1 and A5.3.2 hold. Pick any sets Ω_ξ and Ω_x as in Remark 5.4.1, and suppose that for some iteration $k \in \mathbb{N}$, $u(t_k) \in \Omega_u$ and $x(t_k) \in \Omega_x$. Then, there exists a function $\gamma(\cdot) \in \mathscr{K}_\infty$, a function $\sigma(\cdot) \in \mathscr{L}$ and a number $L_\gamma \in \mathbb{R}_+$ such that

$$\|z^+\|^2 \leq \sigma(T)\left[2L_\gamma\|z\|^2 + \gamma(2\|l(u) - l(u^+)\|^2)\right]. \tag{5.34}$$

Proof. A well known fact that follows from converse Lyapunov theory (q.v. Theorem 4.17 in [80], for example), the characterization of definite functions in terms of comparison functions (q.v. Lemma 4.3 in [80], or the discussion in §24.B in [62] for example) and the properties of comparison functions (q.v. Lemma 4.2 in [80] or §24.A in [62]), is that A5.3.2 implies the existence of a function $\beta(\cdot,\cdot) \in \mathscr{KL}$, such that for a fixed $u \in \mathbb{R}^m$ and any $t > t_0$, the trajectories of Σ' can be bounded as

$$\|z(t)\| \leq \beta(\|z(t_0)\|, t - t_0). \tag{5.35}$$

Let $g_1 : s \mapsto s^{1/2}$ and $g_2 : s \mapsto s^2$. Clearly, both g_1 and g_2 are $\mathscr{K}_\infty$-class functions, and therefore $s \mapsto g_2(\beta(g_1(s),t)) \in \mathscr{K}$ (q.v. Lemma 4.2, [80]). Moreover, since the composition of an increasing and a decreasing function is a decreasing function, the composition $t \mapsto g_2(\beta(g_1(s),t))$ is decreasing. Consequently, the function $\tilde{\beta} : (s,t) \mapsto g_2(\beta(g_1(s),t))$ is a $\mathscr{KL}$-class function. Observing that $\tilde{\beta}(s^2,t) \equiv \beta^2(s,t)$, we write

$$\|z(t)\|^2 \leq \tilde{\beta}(\|z(t_0)\|^2, t - t_0).$$

Lemma 8 in [143] states that for any $\mathscr{KL}$-class function $\tilde{\beta}(\cdot,\cdot)$, there exist $\gamma \in \mathscr{K}_\infty$ and $\tilde{\sigma} \in \mathscr{K}_\infty$ such that for all $[s,t]^T \in \mathbb{R}_+^2$,

$$\tilde{\beta}(s,t) \leq \gamma(s)\tilde{\sigma}(e^{-t}).$$

We define $\sigma : t \mapsto \tilde{\sigma}(e^{-t})$. It is easy to see that since $t \mapsto e^{-t} \in \mathscr{L}$ and $\tilde{\sigma} \in \mathscr{K}_\infty$, the composition σ is an $\mathscr{L}$-class function (q.v. property (a) in §24.A in [62]). We therefore have that

$$\|z(t)\|^2 \leq \sigma(t - t_0)\gamma(\|z(t_0)\|^2). \tag{5.36}$$

Let us now consider the kth iteration of the decision updating processes $\mathscr{D}$, and suppose that $u(t_k) \in \Omega_u$ and

$x(t_k) \in \Omega_x$. From (5.36), we have that

$$
\begin{aligned}
\|z(t_{k+2}^-)\|^2 &\leq \sigma(t_{k+2}^- - t_{k+1})\gamma(\|z(t_{k+1})\|^2) \\
&= \sigma(T)\gamma(\|x(t_{k+1}) - l(u(t_{k+1}))\|^2) && (5.37) \\
&= \sigma(T)\gamma(\|x(t_{k+1}^-) - l(u(t_{k+1}^-)) + l(u(t_{k+1}^-)) - l(u(t_{k+1}))\|^2) && (5.38) \\
&\leq \sigma(T)\gamma(2\|z(t_{k+1}^-)\|^2 + 2\|l(u(t_k)) - l(u(t_{k+1}))\|^2), && (5.39)
\end{aligned}
$$

where (5.37) follows from the definition of the transient error in (5.8), (5.38) follows from the fact that all trajectories of a system such as Σ are continuous, and (5.39) follows from Young's inequality and the monotonicity of $\gamma(\cdot)$.

From the construction of $\hat{\Omega}_x$ in Remark 5.4.1, $x(t_k) \in \Omega_x \implies x(t_{k+1}^-) \in \hat{\Omega}_x$, while $u(t_{k+1}^-) = u(t_k) \in \Omega_u$. Therefore in (5.39),

$$\|z(t_{k+1}^-)\| \leq r_z = \max_{x \in \hat{\Omega}_x,\, u \in \Omega_u} \|x - l(u)\|.$$

Let

$$r_l = \max_{u \in \hat{\Omega}_u} \|l(u)\|.$$

The numbers r_z and r_l are well defined because Ω_u, $\hat{\Omega}_u$ and $\hat{\Omega}_x$ are compact and $l(\cdot)$ is continuous[1].

Since γ is continuous, it is Lipschitz continuous on $[0, 2r_z^2 + 4r_l^2]$ with some constant L_γ. Using the notation defined in (5.32) and (5.33), we thus obtain

$$
\begin{aligned}
\|z^+\|^2 &\leq \sigma(T)\gamma(2\|z\|^2 + 2\|l(u) - l(u^+)\|^2) \\
&= \sigma(T)\gamma(2\|z\|^2 + 2\|l(u) - l(u^+)\|^2) - \sigma(T)\gamma(2\|l(u) - l(u^+)\|^2) \\
&\qquad + \sigma(T)\gamma(2\|l(u) - l(u^+)\|^2) \\
&\leq \sigma(T)2L_\gamma\|z\|^2 + \sigma(T)\gamma(2\|l(u) - l(u^+)\|^2),
\end{aligned}
$$

which was to be shown. □

5.4.2 The Closed-Loop Evolution of $\mathscr{D}$

Lemma 5.4.1 shows that when $u = u^+$ for all $k \in \mathbb{N}$, one may select T such that the sequence $(\|z(t_{k+1}^-)\|)_{k=1}^\infty$ converges monotonically to $\|z\| = 0$. As expected, the presence of the decision updating processes $\mathscr{D}_i$ perturbs the evolution of the dynamics Σ'. Analogously, we now quantify the effect that the dynamics Σ' have on the otherwise unperturbed evolution of consensus optimization schemes of the form (5.18).

Using our notational convention in (5.32) and (5.33), we express $\hat{s}_i(y_i(t_{k+1}^-))$ in terms of $s_i(u)$ in (5.21) as

[1]The compactness of $\hat{\Omega}_u$ follows from the compactness of Ω_x, Ω_u, and the continuity of $h(\cdot)$. The compactness of $\hat{\Omega}_x$ follows from the asymptotic stability of $l(u)$ (for any u) for Σ and arguments analogous to those used in the proof of Claim 2.5.2 (q.v. the proof of Theorem 2.5.1). Namely, it can be shown that when a point $l(u)$ is globally asymptotically stable for Σ, the positive semiorbits of points inside any compact set form another compact set containing $l(u)$.

follows.

$$
\begin{aligned}
\hat{s}_i(y_i(t_{k+1}^-)) &= y_i(t_{k+1}^-)\\
&= h_i(x(t_{k+1}^-))\\
&= h_i(z(t_{k+1}^-) + l(u(t_{k+1}^-)))\\
&= h_i(z + l(u))\\
&= h_i(l(u)) - p_{hi}\\
&= s_i(u) - p_{hi},
\end{aligned}
$$

where

$$
p_{hi} = h_i(l(u)) - h_i(z + l(u)), \tag{5.40}
$$

and $s_i(\cdot)$ is as in (5.21). Then, using (5.21), equation (5.23) can be written as

$$
\mathscr{D} : \begin{cases} \xi^+ = (A \otimes I_m)\xi - \alpha s(\mathbf{1}_N \otimes u) + \alpha p_h \\ u = E(A \otimes I_m)\xi, \quad \forall t \in [t_k, t_{k+1}), \ \forall k \in \mathbb{N}, \end{cases} \tag{5.41}
$$

where

$$
s(\mathbf{1}_N \otimes u) = \begin{bmatrix} s_1(u) \\ \vdots \\ s_N(u) \end{bmatrix}, \tag{5.42}
$$

$$
p_h = [p_{h1}^T, \ldots, p_{hN}^T]^T, \tag{5.43}
$$

and E is as in (5.24).

The Mean and Deviation Dynamics

We denote the agents' mean estimate of some $u^* \in U^*$ by μ, and we define it as

$$
\mu \triangleq \tfrac{1}{N}(\mathbf{1}_N^T \otimes I_m)\xi. \tag{5.44}
$$

Following the arguments in §3.4.1, we observe that based on (5.41), this mean estimate evolves according to

$$
\begin{aligned}
\mu^+ &= \tfrac{1}{N}(\mathbf{1}_N^T \otimes I_m)(A \otimes I_m)\xi - \tfrac{\alpha}{N}(\mathbf{1}_N^T \otimes I_m)s(\mathbf{1}_N \otimes u) + \tfrac{\alpha}{N}(\mathbf{1}_N^T \otimes I_m)p_h\\
&= \mu - \tfrac{\alpha}{N}(\mathbf{1}_N^T \otimes I_m)s(\mathbf{1}_N \otimes u) + \tfrac{\alpha}{N}(\mathbf{1}_N^T \otimes I_m)p_h\\
&= \mu - \tfrac{\alpha}{N}(\mathbf{1}_N^T \otimes I_m)s(\mathbf{1}_N \otimes \mu) + \tfrac{\alpha}{N}(\mathbf{1}_N^T \otimes I_m)p_s + \tfrac{\alpha}{N}(\mathbf{1}_N^T \otimes I_m)p_h\\
&= \mu - \tfrac{\alpha}{N}s_o(\mu) + \tfrac{\alpha}{N}(\mathbf{1}_N^T \otimes I_m)p_s + \tfrac{\alpha}{N}(\mathbf{1}_N^T \otimes I_m)p_h
\end{aligned} \tag{5.45}
$$

where $s_o(\cdot)$ is as in (5.30), and

$$
p_s = s(\mathbf{1}_N \otimes \mu) - s(\mathbf{1}_N \otimes u) = \begin{bmatrix} s_1(\mu) - s_1(u) \\ \vdots \\ s_N(\mu) - s_N(u) \end{bmatrix}. \tag{5.46}
$$

We denote the vector of deviations from this mean by $\zeta = [\zeta_1^T, \ldots, \zeta_N^T]^T \in \mathbb{R}^{Nm}$, and we define it as

$$\zeta = (M \otimes I_m)\xi, \tag{5.47}$$

where M is as in (3.27). Equivalently, for each $i \in \mathscr{V}$, $\zeta_i = \xi_i - \mu$. Based on (5.41), the deviation variable ζ evolves according to

$$\begin{aligned}\zeta^+ &= (M \otimes I_m)(A \otimes I_m)\xi - \alpha(M \otimes I_m)s(\mathbf{1}_N \otimes u) + \alpha(M \otimes I_m)p_h \\ &= (A \otimes I_m)\zeta - \alpha(M \otimes I_m)s(\mathbf{1}_N \otimes u) + \alpha(M \otimes I_m)p_h.\end{aligned} \tag{5.48}$$

Following the developments of Lemmas 3.6.1 and 3.6.2, in the next two lemmas we examine the evolution of the function $V_{\mathscr{O}} = N\|\mu - \mathbf{P}_{U^*}(\mu)\|^2$ along the sequences of (5.45), and the evolution of the function $V_{\mathscr{C}} = \|\zeta\|^2$ along the sequences of (5.48).

Lemma 5.4.2 (Preliminary Mean Subsystem Analysis)**:** Consider the function $V_{\mathscr{O}}(\mu) = N\|\mu - \mathbf{P}_{U^*}(\mu)\|^2$ and the system (5.45), and suppose that A5.3.3 holds. Then, for all μ generated by (5.45),

$$\Delta V_{\mathscr{O}}(\mu) \leq -\alpha\phi(\mu) + \tau_1 + \tau_2,$$

where

$$\tau_1 = 2\alpha\sqrt{N}\|\mu - \mathbf{P}_{U^*}(\mu)\| \cdot \|h(l(\mu)) - h(z + l(u))\| \tag{5.49}$$

$$\tau_2 = \tfrac{\alpha^2}{N}\|(\mathbf{1}_N^T \otimes I_m)h(z + l(u))\|^2, \tag{5.50}$$

Proof. Following the proof of Lemma 3.6.1, we see that

$$\begin{aligned}\Delta V_{\mathscr{O}}(\mu) &\leq N\|\mu - \tfrac{\alpha}{N}s_o(\mu) + \tfrac{\alpha}{N}(\mathbf{1}_N^T \otimes I_m)p_s + \tfrac{\alpha}{N}(\mathbf{1}_N^T \otimes I_m)p_h - \mathbf{P}_{U^*}(\mu)\|^2 \\ &\qquad - N\|\mu - \mathbf{P}_{U^*}(\mu)\|^2, \\ &\leq N\|\tfrac{\alpha}{N}(\mathbf{1}_N^T \otimes I_m)(p_s + p_h) - \tfrac{\alpha}{N}s_o(\mu)\|^2 \\ &\qquad + 2\alpha(\mu - \mathbf{P}_{U^*}(\mu))^T\Big((\mathbf{1}_N^T \otimes I_m)(p_s + p_h) - s_o(\mu)\Big) \\ &= -2\alpha(\mu - \mathbf{P}_{U^*}(\mu))^T s_o(\mu) + 2\alpha(\mu - \mathbf{P}_{U^*}(\mu))^T(\mathbf{1}_N^T \otimes I_m)(p_s + p_h) \\ &\qquad + \tfrac{\alpha^2}{N}\|(\mathbf{1}_N^T \otimes I_m)(p_s + p_h - s(\mathbf{1}_N \otimes \mu))\|^2.\end{aligned} \tag{5.51}$$

We may apply (5.31) to the first term in (5.51) to obtain

$$-2\alpha(\mu - \mathbf{P}_{U^*}(\mu))^T s_o(\mu) \leq -\alpha\phi(\mu), \tag{5.52}$$

where $\phi(\cdot)$ is a function having the properties specified in Definition 3.3.1. Using (5.52) and bounding the second term, we write

$$\begin{aligned}\Delta V_{\mathscr{O}}(\mu) &\leq -\alpha\phi(\mu) + 2\alpha\sqrt{N}\|\mu - \mathbf{P}_{U^*}(\mu)\| \cdot \|p_s + p_h\| \\ &\qquad + \tfrac{\alpha^2}{N}\|(\mathbf{1}_N^T \otimes I_m)(p_s + p_h - s(\mathbf{1}_N \otimes \mu))\|^2.\end{aligned} \tag{5.53}$$

We note that

$$
\begin{aligned}
p_s + p_h &= s(\mathbf{1}_N \otimes \mu) - s(\mathbf{1}_N \otimes u) + s(\mathbf{1}_N \otimes u) - h(z + l(u)) \\
&= s(\mathbf{1}_N \otimes \mu) - h(z + l(u)) \qquad (5.54)\\
&= h(l(\mu)) - h(z + l(u)). \qquad (5.55)
\end{aligned}
$$

We use (5.54) and (5.55) to write (5.53) as

$$
\begin{aligned}
\Delta V_{\mathscr{O}}(\mu) &\leq -\alpha\phi(\mu) + 2\alpha\sqrt{N}\|\mu - \mathbf{P}_{U^*}(\mu)\| \cdot \|h(l(\mu)) - h(z + l(u))\| \\
&\quad + \tfrac{\alpha^2}{N}\|(\mathbf{1}_N^T \otimes I_m)h(z + l(u))\|^2, \\
&\leq -\alpha\phi(\mu) + \tau_1 + \tau_2, \qquad (5.56)
\end{aligned}
$$

which was to be shown. □

Lemma 5.4.3 (Preliminary Deviation Subsystem Analysis)**:** Consider the function $V_{\mathscr{C}}(\zeta) = \|\zeta\|^2$ and the system (5.48), where A satisfies A3.3.1. Then, for all ζ generated by (5.48),

$$
\Delta V_{\mathscr{C}}(\zeta) \leq -(1-\kappa)\|\zeta\|^2 + \tau_3 + \tau_4
$$

where

$$
\begin{aligned}
\tau_3 &= \alpha^2\|(M \otimes I_m)h(z + l(u))\|^2 \qquad (5.57)\\
\tau_4 &= -2\alpha h(z + l(u))^T(A \otimes I_m)\zeta \qquad (5.58)
\end{aligned}
$$

Proof. Following the proof of Lemma 3.6.2, we see that

$$
\begin{aligned}
\Delta V_{\mathscr{C}}(\zeta) &= \|(A \otimes I_m)\xi - \alpha s(\mathbf{1}_N \otimes u) + \alpha p_h\|^2 - \|\zeta\|^2 \\
&= \|(A \otimes I_m)\xi - \alpha h(z + l(u))\|^2 - \|\zeta\|^2 \\
&\leq -(1-\kappa)\|\zeta\|^2 + \alpha^2\|(M \otimes I_m)h(z + l(u))\|^2 - 2\alpha h(z + l(u))^T(A \otimes I_m)\zeta \\
&\leq -(1-\kappa)\|\zeta\|^2 + \tau_3 + \tau_4 \qquad (5.59)
\end{aligned}
$$

where we have applied A3.3.1 to obtain the first term in (5.59). □

5.4.3 Composite Lyapunov Analysis

In the following two lemmas, we propose a composite Lyapunov function candidate $V(\mu, \zeta, z)$ and derive an expression for an upper bound for $\Delta V(\mu, \zeta, z)$. In the next Lemma, which is analogous to Lemma 3.6.3, we simplify the expression that provides an upper bound on $\Delta V_{\mathscr{O}} + \Delta V_{\mathscr{C}}$.

Lemma 5.4.4 (Preliminary Composite Analysis for $\mathscr{D}$)**:** Consider the system (5.45)-(5.48), and suppose that all of the conditions of Lemmas 5.4.2 and 5.4.3 are satisfied. Then, the function $V_{\mathscr{D}}(\mu, \zeta) = V_{\mathscr{O}}(\mu) + V_{\mathscr{C}}(\zeta) = N\|\mu - \mathbf{P}_{U^*}(\mu)\|^2 + \|\zeta\|^2$ is such that for all μ^+ and ζ^+ generated by (5.45)-(5.48),

$$
\begin{aligned}
\Delta V_{\mathscr{D}}(\mu, \zeta) \leq -\alpha\phi(\mu) - (1-\kappa)\|\zeta\|^2 + \tau_1 + \tau_4 \\
+ \alpha^2\|h(z + l(u))\|^2,
\end{aligned}
$$

where τ_1 and τ_4 are as in Lemma 5.4.2 and Lemma 5.4.3, respectively.

Proof.

Claim 5.4.1: For any quantity $q \in \mathbb{R}^{Nm}$, it holds that

$$\tfrac{1}{N}\|(\mathbf{1}_N^T \otimes I_m)q\|^2 + \|(M \otimes I_m)q\|^2 = \|q\|^2.$$

Proof.

$$\begin{aligned}
\tfrac{1}{N}\|(\mathbf{1}_N^T \otimes I_m)q\|^2 + \|(M \otimes I_n)q\|^2 &= \tfrac{1}{N}q^T(\mathbf{1}_N^T \otimes I_m)^T(\mathbf{1}_N^T \otimes I_m)q + q^T(M \otimes I_m)q \\
&= q^T\Big(\tfrac{1}{N}\mathbf{1}_N\mathbf{1}_N^T \otimes I_m + (M \otimes I_m)\Big)q \\
&= q^T\Big((\tfrac{1}{N}\mathbf{1}_N\mathbf{1}_N^T + M) \otimes I_m\Big)q \\
&= q^T\Big(I_N \otimes I_m\Big)q \\
&= \|q\|^2
\end{aligned}$$

$\diamondsuit$

Applying this claim to the sum $\tau_2 + \tau_3$ yields the desired result. □

In the next lemma we refine the conclusions of Lemmas 5.4.4 and 5.4.1.

Lemma 5.4.5: Consider the system $\Sigma' - \mathscr{D}$ and suppose that the conditions of Lemmas 5.4.4 and 5.4.1 hold. Let Ω_ξ and Ω_x be arbitrarily chosen sets as described in Remark 5.4.1, and let the sets $\hat{\Omega}_x$, $\hat{\Omega}_u$ and $\hat{\Omega}_\mu$ be defined from Ω_ξ and Ω_x, as described in Remark 5.4.1. Then, whenever $u(t_k) \in \Omega_u$ and $x(t_k) \in \Omega_x$ for some iteration $k \in \mathbb{N}$, it holds for the composite Lyapunov function candidate

$$V(\mu,\zeta,z) = N\|\mu - \mathbf{P}_{U^*}(\mu)\|^2 + \|\zeta\|^2 + \|z\|^2, \tag{5.60}$$

that

$$\Delta V(\mu,\zeta,z) \le -\alpha\phi(\mu) + \alpha^2 K_\mu\|\mu - \mathbf{P}_{U^*}(\mu)\|^2 - K_\zeta\|\zeta\|^2 - K_z\|z\|^2 + K, \tag{5.61}$$

where

$$K_\mu = \tfrac{N}{\sigma(T)} + 8L_hL_l^2(1 + 4\sigma(T)L_\gamma L_l^2 + \tfrac{1}{\sigma(T)}) \tag{5.62}$$

$$K_\zeta = 1 - \kappa - \sigma(T)\kappa\Big(1 + 2L_h^2 + 16L_\gamma L_l^2(1 + \alpha L_hL_l^2)\Big) - 4\alpha^2\kappa(\tfrac{1}{\sigma(T)} + 1)L_h^2L_l^2 \tag{5.63}$$

$$K_z = 1 - 2\sigma(T)\Big(L_\gamma + L_h^2 + 4\alpha^2L_\gamma L_l^2L_h^2\Big) - 2\alpha^2(\tfrac{1}{\sigma(T)} + 1)L_h^2 \tag{5.64}$$

$$K = 8\alpha^2(\tfrac{1}{\sigma(T)} + 1)s^* + \sigma(T)\gamma(32\alpha^2L_l^2s^*), \tag{5.65}$$

$\sigma(\cdot)$ and L_γ are as in Lemma 5.4.1,

$$s^* = \max_{\mu \in U^*}\|h(l(\mu))\|^2, \tag{5.66}$$

L_h is the Lipschitz constant associated to $h(\cdot)$ on $\hat{\Omega}_x$, L_l is the Lipschitz constant associated to $l(\cdot)$ on $\hat{\Omega}_u$, and κ is as in (5.22).

Proof. Combining the conclusions of Lemmas 5.4.4 and 5.4.1, we see that

$$\begin{aligned}\Delta V(\mu,\zeta,z) \le & -\alpha\phi(\mu) - (1-\kappa)\|\zeta\|^2 - (1-2L_\gamma\sigma(T))\|z\|^2 \\ & + \tau_1 + \tau_4 + \alpha^2\|h(z+l(u))\|^2 \\ & + \sigma(T)\gamma(2\|l(u)-l(u^+)\|^2). \end{aligned} \tag{5.67}$$

The following three claims are useful in bounding the terms in (5.67).

Claim 5.4.2: For all $u \in \mathbb{R}^m$,

$$\|u^+ - u\|^2 \le 8\kappa\|\zeta\|^2 + 2\alpha^2\|h(z+l(u))\|^2.$$

Proof. Using (5.41), we write

$$\begin{aligned}\|u^+ - u\|^2 &= \|E(A\otimes I_m)\xi^+ - E(A\otimes I_m)\xi\|^2 \\ &= \|E(A\otimes I_m)(\xi^+ - \xi)\|^2 \\ &\le \|(A\otimes I_m)(\xi^+ - \xi)\|^2 \\ &\le \kappa\|\xi^+ - \xi\|^2 \\ &= \kappa\|(A\otimes I_m)\xi - \alpha h(z+l(u)) - \xi\|^2 \\ &= \kappa\|((A-I_N)\otimes I_m)\xi - \alpha h(z+l(u))\|^2 \\ &= \kappa\|((A-I_N)\otimes I_m)\zeta - \alpha h(z+l(u))\|^2,\end{aligned}$$

where the last equality is obtained by noting that $\xi = \zeta + \mathbf{1}_N \otimes \mu$, and that $((A-I_N)\otimes I_m)(\mathbf{1}_N\otimes\mu) = \mathbf{0}$. Letting $Q = (A-I_N)^2$ and applying Young's inequality to the above expression, we obtain

$$\|u^+ - u\|^2 \le \kappa 2\zeta^T(Q^2\otimes I_m)\zeta + 2\alpha^2\|h(z+l(u))\|^2.$$

The spectral mapping theorem implies that for any $\lambda \in \sigma(A)$, $\lambda^2 - 2\lambda + 1 \in \sigma(Q^2)$. By A3.3.1, $\lambda \in (-1,1]$, for any $\lambda \in \sigma(A)$. Therefore, $\lambda_o \in [0,4]$, for any $\lambda_o \in \sigma(Q^2)$. With this observation we may bound the above expression as

$$\begin{aligned}\|u^+ - u\|^2 &\le \kappa 8\lambda_1(A)\|\zeta\|^2 + 2\alpha^2\|h(z+l(u))\|^2 \\ &= 8\kappa\|\zeta\|^2 + 2\alpha^2\|h(z+l(u))\|^2,\end{aligned}$$

and the claim is proved. $\diamondsuit$

Claim 5.4.3: For all u and μ in $\mathbb{R}^m$,

$$\|u-\mu\|^2 \le \kappa\|\zeta\|^2$$

Proof. We recall from (5.23) that $u = E(A\otimes I_m)\xi$, where E is as in (5.24). From the definition of the deviation variable in (5.47), we see that $\xi = \zeta + \mathbf{1}_N \otimes \mu$, and therefore

$$\begin{aligned} u &= E(A \otimes I_m)\xi \\ &= E(A \otimes I_m)\zeta + E(A \otimes I_m)(\mathbf{1}_N \otimes \mu) \\ &= E(A \otimes I_m)\zeta + \mu, \end{aligned}$$

owing to the properties of the Kronecker product, the row-stochasticity of A, and the fact that $E(\mathbf{1}_N \otimes \mu) = \mu$. Therefore,

$$\begin{aligned} \|u - \mu\|^2 &\leq \|E(A \otimes I_m)\zeta\|^2 \\ &\leq \|(A \otimes I_m)\zeta\|^2 \\ &\leq \kappa \|\zeta\|^2, \end{aligned}$$

and the claim is proved. $\diamondsuit$

Claim 5.4.4: For all $u \in \Omega_u$ and $z \in \Omega_z$

$$\|h(z + l(u))\|^2 \leq 2L_h^2 \|z\|^2 + 4L_h^2 L_l^2 \kappa \|\zeta\|^2 + 8L_h^2 L_l^2 \|\mu - \mathbf{P}_{U^*}(\mu)\|^2 + 8s^*.$$

Proof. The continuity of $l(\cdot)$ implies its local Lipschitz continuity, and we let L_l denote its Lipschitz constant on the set $\hat{\Omega}_\mu$, defined in Remark 5.4.1. Moreover, the continuity of $h(\cdot)$ implies its Lipschitz continuity, and we let L_h denote its Lipschitz constant on the set $\hat{\Omega}_x$, defined in Remark 5.4.1. We may then write

$$\begin{aligned} \|h(z + l(u))\|^2 &= \|h(z + l(u)) - h(l(u)) + h(l(u))\|^2 \\ &\leq 2\|h(z + l(u)) - h(l(u))\|^2 + 2\|h(l(u)) - h(l(\mu)) + h(l(\mu))\|^2 \\ &\leq 2L_h^2 \|z\|^2 + 4\|h(l(u)) - h(l(\mu))\|^2 + 4\|h(l(\mu)) - h(l(\mathbf{P}_{U^*}(\mu))) + h(l(\mathbf{P}_{U^*}(\mu)))\|^2 \\ &\leq 2L_h^2 \|z\|^2 + 4L_h^2 L_l^2 \|u - \mu\|^2 + 8\|h(l(\mu)) - h(l(\mathbf{P}_{U^*}(\mu)))\|^2 + 8\|h(l(\mathbf{P}_{U^*}(\mu)))\|^2 \\ &\leq 2L_h^2 \|z\|^2 + 4L_h^2 L_l^2 \|u - \mu\|^2 + 8L_h^2 L_l^2 \|\mu - \mathbf{P}_{U^*}(\mu)\|^2 + 8\|h(l(\mathbf{P}_{U^*}(\mu)))\|^2. \end{aligned}$$

Then, by Claim 5.4.3 and the definition of s^* in (5.66), we obtain

$$\|h(z + l(u))\|^2 \leq 2L_h^2 \|z\|^2 + 4L_h^2 L_l^2 \kappa \|\zeta\|^2 + 8L_h^2 L_l^2 \|\mu - \mathbf{P}_{U^*}(\mu)\|^2 + 8s^*,$$

which was to be shown. $\diamondsuit$

Claim 5.4.5: For all $\mu \in \Omega_\mu$ and $z \in \Omega_z$,

$$\|h(l(\mu)) - h(z + l(u))\|^2 \leq 2L_h^2 \kappa \|\zeta\|^2 + 2L_h^2 \|z\|^2.$$

Proof.

$$\begin{aligned} \|h(l(\mu)) - h(z + l(u))\|^2 &\leq L_h^2 \|l(\mu) - l(u) - z\|^2 \\ &\leq 2L_h^2 \|l(\mu) - l(u)\|^2 + 2L_h^2 \|z\|^2. \end{aligned}$$

By Claim 5.4.3, we obtain

$$\|h(l(\mu)) - h(z+l(u))\|^2 \leq 2L_h^2 \kappa \|\zeta\|^2 + 2L_h^2 \|z\|^2,$$

which was to be shown. $\diamondsuit$

We now return to expression (5.67). Recalling the expressions for τ_1 and τ_4 from Lemmas 5.4.2 and 5.4.3, we write

$$\begin{aligned}\Delta V(\mu,\zeta,z) \leq & -\alpha\phi(\mu) - (1-\kappa)\|\zeta\|^2 - (1-2L_\gamma \sigma(T))\|z\|^2 \\ & + 2\alpha\sqrt{N}\|\mu - \mathbf{P}_{U^*}(\mu)\| \cdot \|h(l(\mu)) - h(z+l(u))\| \\ & - 2\alpha h(z+l(u))^T (A\otimes I_m)\zeta + \alpha^2 \|h(z+l(u))\|^2 \\ & + \sigma(T)\gamma(2\|l(u) - l(u^+)\|^2). \end{aligned} \tag{5.68}$$

We apply Young's inequality to the fourth and fifth terms above to obtain

$$\begin{aligned}\Delta V(\mu,\zeta,z) \leq & -\alpha\phi(\mu) - (1-\kappa)\|\zeta\|^2 - (1-2L_\gamma \sigma(T))\|z\|^2 \\ & + \alpha^2 \tfrac{N}{\varepsilon_1}\|\mu - \mathbf{P}_{U^*}(\mu)\|^2 + \varepsilon_1 \|h(l(\mu)) - h(z+l(u))\|^2 \\ & \alpha^2 \tfrac{1}{\varepsilon_2}\|h(z+l(u))\|^2 + \varepsilon_2 \kappa \|\zeta\|^2 + \alpha^2 \|h(z+l(u))\|^2 \\ & + \sigma(T)\gamma(2\|l(u) - l(u^+)\|^2). \end{aligned} \tag{5.69}$$

which holds for any positive, real ε_1 and ε_2. Choosing $\varepsilon_1 = \varepsilon_2 = \sigma(T)$ and combining like terms, we obtain

$$\begin{aligned}\Delta V(\mu,\zeta,z) \leq & -\alpha\phi(\mu) - (1-\kappa)\|\zeta\|^2 - (1-2L_\gamma \sigma(T))\|z\|^2 \\ & + \alpha^2 \tfrac{N}{\sigma(T)}\|\mu - \mathbf{P}_{U^*}(\mu)\|^2 + \sigma(T) \|h(l(\mu)) - h(z+l(u))\|^2 \\ & \alpha^2 (\tfrac{1}{\sigma(T)} + 1)\|h(z+l(u))\|^2 + \sigma(T) \kappa \|\zeta\|^2 \\ & + \sigma(T)\gamma(2\|l(u) - l(u^+)\|^2). \end{aligned} \tag{5.70}$$

Next we examine the last term in the above expression. From the definition of $\hat{\Omega}_u$, it follows that both u and u^+ belong to $\hat{\Omega}_u$ whenever $u \in \Omega_u$. Therefore we may write

$$\begin{aligned}\gamma(2\|l(u) - l(u^+)\|^2) & \leq \gamma(2L_l^2 \|u - u^+\|^2) \\ & \leq \gamma(16L_l^2 \kappa \|\zeta\|^2 + 4L_l^2 \alpha^2 \|h(z+l(u))\|^2), \end{aligned} \tag{5.71}$$

where the second inequality is obtained by applying Claim 5.4.2. Adding and subtracting $\gamma(4L_l^2\alpha^2\|h(z+l(u))\|^2)$ to the right-hand side, we obtain

$$\begin{aligned}\gamma(2\|l(u) - l(u^+)\|^2) & \leq \gamma(16L_l^2 \kappa \|\zeta\|^2 + 4L_l^2 \alpha^2 \|h(z+l(u))\|^2) \pm \gamma(4L_l^2 \alpha^2 \|h(z+l(u))\|^2) \\ & \leq 16L_\gamma L_l^2 \kappa \|\zeta\|^2 + \gamma(4L_l^2 \alpha^2 \|h(z+l(u))\|^2), \end{aligned} \tag{5.72}$$

where L_γ is as in the proof of Lemma 5.4.1. Next, recalling Claim 5.4.4, we write

$$\begin{aligned}\gamma(2\|l(u)-l(u^+)\|^2) \leq 16L_\gamma L_l^2\kappa\|\zeta\|^2 & \\ &+\gamma(c_1\|z\|^2+c_2\|\zeta\|^2+c_3\|\mu-\mathbf{P}_{U^*}(\mu)\|^2+32L_l^2\alpha^2 s^*) \\ &\pm\gamma(c_2\|\zeta\|^2+c_3\|\mu-\mathbf{P}_{U^*}(\mu)\|^2+32L_l^2\alpha^2 s^*) \\ &\pm\gamma(c_3\|\mu-\mathbf{P}_{U^*}(\mu)\|^2+32L_l^2\alpha^2 s^*) \\ &\pm\gamma(32L_l^2\alpha^2 s^*), \end{aligned} \tag{5.73}$$

where

$$\begin{aligned} c_1 &= 8L_l^2\alpha^2L_h^2 \\ c_2 &= 16\alpha^2L_h^2L_l^4\kappa \\ c_3 &= 32\alpha^2L_h^2L_l^4. \end{aligned} \tag{5.74}$$

Expression (5.73) becomes

$$\begin{aligned}\gamma(2\|l(u)-l(u^+)\|^2) &\leq 16L_\gamma L_l^2\kappa\|\zeta\|^2+L_\gamma c_1\|z\|^2+L_\gamma c_2\|\zeta\|^2+L_\gamma c_3\|\mu-\mathbf{P}_{U^*}(\mu)\|^2+\gamma(32L_l^2\alpha^2s^*) \\ &= 32\alpha^2L_\gamma L_h^2L_l^4\|\mu-\mathbf{P}_{U^*}(\mu)\|^2+16L_\gamma L_l^2\kappa(1+\alpha^2L_h^2L_l^2)\|\zeta\|^2 \\ &\quad +8\alpha^2L_\gamma L_l^2L_h^2\|z\|^2+\gamma(32L_l^2\alpha^2s^*) \end{aligned} \tag{5.75}$$

Finally, combining expressions (5.75) and (5.70), applying Claims 5.4.5 and 5.4.4 to the relevant terms in (5.70) and then combining like terms, we obtain the sought expression (5.61). □

5.5 Semiglobal, Practical, Asymptotic Stability for $\Sigma-\mathscr{D}$

Using the conclusions of Lemma 5.4.5, we derive small-gain conditions on the tunable parameters α and T appearing in the $\Sigma-\mathscr{D}$ interconnection, ensuring that $V(\mu,\zeta,z)$ is a valid Lyapunov function for the system $\Sigma-\mathscr{D}$. We then show how the conditions of Theorem 2.5.1 can be verified for $V(\mu,\zeta,z)$.

5.5.1 Small Gain Conditions

Theorem 5.5.1: Consider the function $\hat{W}:(\mu,\zeta,z)\mapsto -\Delta V(\mu,\zeta,z)+K$, given by

$$\hat{W}(\mu,\zeta,z;\alpha,T)=\alpha\phi(\mu)-\alpha^2K_\mu\|\mu-\mathbf{P}_{U^*}(\mu)\|^2+K_\zeta\|\zeta\|^2+K_z\|z\|^2,$$

where K_μ, K_ζ, K_z and K are as in (5.62), (5.63), (5.64) and (5.65) respectively. For any $\hat{r}\in\mathbb{R}_{++}$ and $\hat{\delta}\in\mathbb{R}_{++}$, the function $\hat{W}(\cdot,\cdot,\cdot;\alpha,T)$ is positive definite on $\bar{B}_{\hat{r}}(U^*)\times\mathbb{R}^{Nm}\times\mathbb{R}^n$, with respect to $\bar{B}_{\hat{\delta}}(U^*)\times\{\mathbf{0}\}\times\{\mathbf{0}\}$, provided that all of the following conditions hold:

$$\sigma(T)\Big(1+2L_h^2+16L_\gamma L_l^2(1+\alpha L_hL_l^2)\Big)<\frac{1-\kappa}{\kappa} \tag{5.76}$$

$$\sigma(T)L_\gamma\Big(1+L_\gamma+4\alpha^2L_h^2L_l^2\Big)+\alpha^2\big(\tfrac{1}{\sigma(T)}+1\big)L_h^2<\frac{1}{2} \tag{5.77}$$

and

$$\alpha\Big(\tfrac{N}{\sigma(T)}+8L_hL_l^2(1+4\sigma(T)L_\gamma L_l^2+\tfrac{1}{\sigma(T)})\Big)\|\mu-\mathbf{P}_{U^*}(\mu)\|^2<\phi(\mu),\quad \forall\mu\in\bar{B}_{\hat{r}}(U^*)\backslash B_{\hat{\delta}}(U^*). \tag{5.78}$$

Moreover, for any given set of problem parameters $\{N,\kappa,L_\gamma,L_l,L_h\}$, there exist positive, real numbers $\bar{\alpha}$ and $\bar{T}$ such that (5.76), (5.77) and (5.78), are satisfied whenever $\alpha\in(0,\bar{\alpha})$ and $T\in(\bar{T},\infty)$.

Proof. Conditions (5.76), (5.77) and (5.78) follow directly by imposing the positive definiteness of the terms $K_\zeta\|\zeta\|^2$, $K_z\|z\|^2$ and $\alpha\phi(\mu)-\alpha^2K_\mu\|\mu-\mathbf{P}_{U^*}(\mu)\|^2$ in $\hat{W}(\cdot,\cdot,\cdot;\alpha,T)$, respectively.

The following is *one* way to construct $\bar{\alpha}$ and $\bar{T}$. From the properties of $\mathscr{L}$-class functions such as $\sigma(\cdot)$, it is evident that for any $c\in\mathbb{R}_{++}$, there exists a T_c such that $\sigma(T_c)<c$.

Let $\alpha_\zeta=1$, and take T_ζ to be such that

$$\sigma(T_\zeta)\Big(1+2L_h^2+16L_\gamma L_l^2(1+\alpha_\zeta L_hL_l^2)\Big)<\frac{1-\kappa}{\kappa}.$$

Then, condition (5.76) is satisfied whenever $\alpha\in(0,\alpha_\zeta)$ and $T\in(T_\zeta,\infty)$.

Next, take T_z to be such that

$$\sigma(T_z)L_\gamma\Big(1+L_\gamma+4\alpha_\zeta^2L_h^2L_l^2\Big)<\frac{1}{4}$$

where $\alpha_\zeta=1$, as above. Then take α_z to be such that

$$\alpha_z^2\Big(\max\{\tfrac{1}{\sigma(T_\zeta)},\tfrac{1}{\sigma(T_z)}\}+1\Big)L_h^2<\frac{1}{4}$$

Then, conditions (5.76) and (5.77) are satisfied whenever $\alpha\in(0,\min\{\alpha_\zeta,\alpha_z\})$ and $T\in(\max\{T_\zeta,T_z\},\infty)$.

To satisfy condition (5.78), we apply Lemma 2.5.2, which states that for any positive, real numbers K_ϕ, $\hat{r}$ and $\hat{\delta}$, there exists a number $\alpha_\phi\in\mathbb{R}_{++}$ such that

$$\alpha K_\phi\|\mu-\mathbf{P}_{U^*}(\mu)\|^2\leq\phi(\mu)\quad\forall\mu\in\bar{B}_{\hat{r}}(U^*)\backslash B_{\hat{\delta}}(U^*), \tag{5.79}$$

whenever $\alpha\in(0,\alpha_\phi)$. The proof of Lemma 2.5.2 indicates that one such α_ϕ can be taken as

$$\alpha_\phi=\frac{c}{K_\phi\hat{r}^2}, \tag{5.80}$$

where $c\in\mathbb{R}_{++}$ is such that the c-sublevel set of $\phi(\cdot)$ is the largest sublevel set of $\phi(\cdot)$ to be strictly contained inside $\bar{B}_{\hat{\delta}}(U^*)$ (by Lemma 2.5.1, such a c always exists). Let

$$\bar{T}=\max\{T_\zeta,T_z\}. \tag{5.81}$$

Then, taking any

$$K_\phi>\tfrac{N}{\sigma(\bar{T})}+8L_hL_l^2(1+4\sigma(\bar{T})L_\gamma L_l^2+\tfrac{1}{\sigma(\bar{T})}) \tag{5.82}$$

means that condition 5.78 is satisfied whenever $\alpha\in(0,\alpha_\phi)$.

In conclusion, the small-gain conditions (5.76), (5.77) and (5.78) are all satisfied whenever $\alpha\in(0,\bar{\alpha})$, where

$$\bar{\alpha}=\min\{\alpha_\zeta,\alpha_z,\alpha_\phi\}, \tag{5.83}$$

and $T \in (\bar{T}, \infty)$, where $\bar{T}$ is as in (5.81). □

The following is a restatement of Theorem 2.5.1 using notation that is more suitable to the contents of this chapter. The associated definitions and the proof of the theorem are straightforwardly adapted.

Theorem 5.5.2: Consider the system

$$q^+ = f(q; \alpha, T), \qquad q \in \mathbb{R}^\nu, \tag{5.84}$$

where $f : \mathbb{R}^\nu \to \mathbb{R}^\nu$ is parametrized by $\alpha \in \mathbb{R}_{++}$ and $T \in \mathbb{R}_{++}$. Suppose there exists a compact set $\Gamma_0 \subset \mathbb{R}^\nu$ and a function $V \in C^0[\mathbb{R}^\nu, \mathbb{R}_+]$ which is radially unbounded and positive definite with respect to Γ_0 on $\mathbb{R}^\nu$. Suppose further that for every $r \in \mathbb{R}_{++}$, there exists a number $\bar{\alpha} \in \mathbb{R}_{++}$ and a number $\bar{T} \in \mathbb{R}_{++}$ such that whenever $\alpha \in (0, \bar{\alpha})$ and $T \in (\bar{T}, \infty)$,

$$V(q^+) - V(q) \leq W(q; \alpha, T), \quad \forall q \in \bar{B}_r(\Gamma_0), \tag{5.85}$$

where the function $W \in C^0[\mathbb{R}^\nu, \mathbb{R}]$, parameterized by α and T, has the following properties:

- P1: There exists $b_W(\alpha, T) \in \mathbb{R}_{++}$ such that

$$W(q; \alpha, T) \leq b_W(\alpha, T), \quad \forall q \in \bar{B}_r(\Gamma_0),$$

  and $\lim_{\alpha \downarrow 0} b_W(\alpha, T) = 0$.

- P2: The set

$$Z(\alpha, T) = \{q \in \mathbb{R}^\nu \mid W(q; \alpha, T) \geq 0\} \tag{5.86}$$

  contains Γ_0.

- P3: For every $\delta \in (0, r)$, there exists an $\alpha_Z \in \mathbb{R}_{++}$ and a $T_Z \in \mathbb{R}_{++}$ such that whenever $\alpha \in (0, \min\{\alpha_Z, \bar{\alpha}\})$ and $T \in (\max\{\bar{T}, T_Z\}, \infty)$, $Z(\alpha, T) \subseteq \bar{B}_\delta(\Gamma_0)$.

Then, Γ_0 is semiglobally practically asymptotically stable for (5.84). ♢

In the following theorem, we use the results in Theorem 5.5.1 to verify the conditions of Theorem 5.5.2, thereby showing that the set $\Gamma_0 = U^* \times \{\mathbf{0}\} \times \{\mathbf{0}\}$ is SPAS for the system (5.45)-(5.48)-(5.34).

Theorem 5.5.3: Suppose that conditions A5.3.1, A5.3.2, A5.3.3, A5.3.4 and A3.3.1 are satisfied. Then, the set $\Gamma_0 = U^* \times \{\mathbf{0}\} \times \{\mathbf{0}\} \subset \mathbb{R}^m \times \mathbb{R}^{Nm} \times \mathbb{R}^n$ is SPAS for the system (5.45)-(5.48)-(5.34).

Proof. We apply Theorem 5.5.2, with $q = (\mu^T, \zeta^T, z^T)^T$. The function $V(q)$ in (5.60) is continuous, radially unbounded and positive definite with respect to the set $\Gamma_0 = U^* \times \{\mathbf{0}\} \times \{\mathbf{0}\}$ on $\mathbb{R}^m \times \mathbb{R}^{Nm} \times \mathbb{R}^n$.

Let $r \in \mathbb{R}_{++}$ be an arbitrary positive number, and select the sets Ω_ξ and Ω_x in Remark 5.4.1 such that $\Omega_\mu \supset \bar{B}_r^m(U^*)$, $\Omega_\zeta \supset \bar{B}_r^{Nm}(\{\mathbf{0}\})$ and $\Omega_z \supset \bar{B}_r^n(\{\mathbf{0}\})$. Then, it holds that $\Omega_\mu \times \Omega_\zeta \times \Omega_z \supset \bar{B}_r(\Gamma_0)$, and by Lemma 5.4.5, the inequality (5.85) holds on $\bar{B}_r(\Gamma_0)$ with

$$W(\mu, \zeta, z; \alpha, T) = -\alpha\phi(\mu) + \alpha^2 K_\mu \|\mu - \mathbf{P}_{U^*}(\mu)\|^2 - K_\zeta \|\zeta\|^2 - K_z \|z\|^2 + K, \tag{5.87}$$

where K_μ, K_ζ, K_z and K in (5.62), (5.63), (5.64) and (5.65) correspond to this choice of Ω_ξ and Ω_x.

Let $\delta \in (0,r)$ be arbitrary as in P3. In the proof of Theorem 5.5.1, take $\hat{\delta} = \frac{\delta}{\sqrt{3}}$ and $\hat{r} = r$. To satisfy condition (5.78), take $T = \bar{T}$, where $\bar{T}$ is as in (5.81), and select K_ϕ in (3.64) according to

$$K_\phi = \frac{K}{\hat{\delta}^2} + K_\mu,$$

where the coefficients K and K_μ given in (5.65) and(5.62) are evaluated at $\alpha = \min\{\alpha_\zeta, \alpha_z\}$ and $T = \bar{T}$. Let $\bar{\alpha}$ correspond to this choice, according to (5.80) and (5.83).

Then, according to Theorem 5.5.1

$$\begin{aligned} W(q;\alpha,T) &\leq -\alpha(K_\phi - K_\mu)\|\mu - \mathbf{P}_{U^*}(\mu)\|^2 - K_\zeta\|\zeta\|^2 - K_z\|z\|^2 + K \\ &\leq -\alpha^2 \tfrac{K}{\hat{\delta}^2}\|\mu - \mathbf{P}_{U^*}(\mu)\|^2 - K_\zeta\|\zeta\|^2 - K_z\|z\|^2 + K, \end{aligned} \tag{5.88}$$

for all $q \in \left(\bar{B}_r^m(U^*)\backslash B_{\hat{\delta}}^m(U^*)\right) \times \mathbb{R}^{Nm} \times \mathbb{R}^n$, with $K_\zeta > 0$ and $K_z > 0$, provided $\alpha \in (0,\bar{\alpha})$ and $T \in (\bar{T},\infty)$. In other words, $W(q;\alpha,T) - K < 0$ on $\left(\bar{B}_r^m(U^*)\backslash B_{\hat{\delta}}^m(U^*)\right) \times \mathbb{R}^{Nm} \times \mathbb{R}^n$.

From (5.88) and (5.87) together, we observe that on the set $\bar{B}_r^m(U^*)) \times \mathbb{R}^{Nm} \times \mathbb{R}^n$, which contains the set $\bar{B}_r(\Gamma_0)$, $W(q;\alpha,T)$ must be bounded by

$$W(q;\alpha,T) \leq b_W(\alpha,T),$$

where

$$b_W(\alpha,T) = \alpha^2 K_\mu \hat{\delta}^2 + K.$$

Recalling the expression for K given in (5.65) (and the properties of the $\mathscr{K}$-class function $\gamma(\cdot)$), we see that $\lim_{\alpha\downarrow 0} b_W(\alpha,T) = 0$, and that P1 is therefore satisfied.

Next, recall that $Z(\alpha,T)$ denotes the set of all q in $\mathbb{R}^{m+Nm+n}$ for which $W(q;\alpha,T)$ is non-negative. In particular,

$$Z(\alpha,T) = \{q \in \mathbb{R}^{m+Nm+n} \mid \alpha\phi(\mu) - \alpha^2 K_\mu\|\mu - \mathbf{P}_{U^*}(\mu)\|^2 + K_\zeta\|\zeta\|^2 + K_z\|z\|^2 \leq K\},$$

from which it is clear that any point in $\Gamma_0 = U^* \times \{\mathbf{0}\} \times \{\mathbf{0}\}$ also belongs to $Z(\alpha,T)$. Property P2 is thereby satisfied.

Note that since $\hat{\delta} = \frac{\delta}{\sqrt{3}}$, $\bar{B}_\delta(\Gamma_0) \supset \bar{B}_{\hat{\delta}}^m(U^*) \times \bar{B}_{\hat{\delta}}^{Nm}(\{\mathbf{0}\}) \times \bar{B}_{\hat{\delta}}^n(\{\mathbf{0}\})$. To show that P3 is satisfied, we aim to show that for some $\alpha_Z \in \mathbb{R}_{++}$ and $T_Z \in \mathbb{R}_{++}$,

$$\begin{aligned} \bar{B}_\delta(\Gamma_0) &\supset \bar{B}_{\hat{\delta}}^m(U^*) \times \bar{B}_{\hat{\delta}}^{Nm}(\{\mathbf{0}\}) \times \bar{B}_{\hat{\delta}}^n(\{\mathbf{0}\}) \\ &\supset S_\mu \times S_\zeta \times S_z \\ &\supset Z(\alpha,T), \end{aligned}$$

for some sets S_μ, S_ζ and S_z, whenever $\alpha \in (0,\min\{\bar{\alpha},\alpha_Z\})$ and $T \in (\max\{\bar{T},T_Z\},\infty)$.

Consider the sets

$$S_\mu = \{\mu \in \mathbb{R}^m \mid \alpha\phi(\mu) - \alpha^2 K_\mu\|\mu - \mathbf{P}_{U^*}(\mu)\|^2 \leq K\} \tag{5.89}$$

$$S_\zeta = \{\zeta \in \mathbb{R}^{Nm} \mid K_\zeta\|\zeta\|^2 \leq K\} \tag{5.90}$$

$$S_z = \{z \in \mathbb{R}^n \mid K_z\|z\|^2 \leq K\}. \tag{5.91}$$

Clearly, for all $q = [\mu^T, \zeta^T, z^T]^T \in Z(\alpha, T)$, q also belongs to $S_\mu \times S_\zeta \times S_z$, and therefore $S_\mu \times S_\zeta \times S_z \supset Z(\alpha, T)$. Meanwhile, for

$$\bar{B}^m_{\hat{\delta}}(U^*) \times \bar{B}^{Nm}_{\hat{\delta}}(\{\mathbf{0}\}) \times \bar{B}^n_{\hat{\delta}}(\{\mathbf{0}\}) \supset S_\mu \times S_\zeta \times S_z$$

to hold, it suffices that

$$\bar{B}^m_{\hat{\delta}}(U^*) \supset S_\mu, \tag{5.92}$$

$$\bar{B}^{Nm}_{\hat{\delta}}(\{\mathbf{0}\}) \supset S_\zeta, \quad \text{and} \tag{5.93}$$

$$\bar{B}^n_{\hat{\delta}}(\{\mathbf{0}\}) \supset S_z. \tag{5.94}$$

From the definition of $\bar{B}^{Nm}_{\hat{\delta}}(\{\mathbf{0}\})$, S_ζ, K and K_ζ, it follows that for $\alpha \in (0, \bar{\alpha})$ and $T \in (\bar{T}, \infty)$, (5.93) holds whenever

$$8\alpha^2 s^* + \sigma(T)\gamma(16\alpha^2 L_l s^*) \le \tfrac{\delta^2}{3}\Big[(1-\kappa) - 2L^2 N\kappa\big(\sigma(\bar{T}) + 2\bar{\alpha}^2(1+2L_\gamma L_l)\big)\Big]. \tag{5.95}$$

Let $\tilde{\alpha}_\zeta$ and $\tilde{T}_\zeta$ be any positive real numbers such that (5.95) holds whenever $\alpha \in (0, \tilde{\alpha}_\zeta)$ and $T \in (\tilde{T}_\zeta, \infty)$.

Similarly, from the definition of $\bar{B}^n_{\hat{\delta}}(\{\mathbf{0}\})$, S_z, K and K_z, it follows that for $\alpha \in (0, \bar{\alpha})$ and $T \in (\bar{T}, \infty)$, (5.94) holds whenever

$$8\alpha^2 s^* + \sigma(T)\gamma(16\alpha^2 L_l s^*) \le \tfrac{\delta^2}{3}\Big[(1 - 2L_\gamma\sigma(\bar{T})) - 2NL_h^2\big(\sigma(\bar{T}) + \bar{\alpha}^2 + 2\bar{\alpha}^2\sigma(\bar{T})L_\gamma L_l\big)\Big]. \tag{5.96}$$

Let $\tilde{\alpha}_z$ and $\tilde{T}_z$ be any positive real numbers such that (5.96) holds whenever $\alpha \in (0, \tilde{\alpha}_z)$ and $T \in (\tilde{T}_z, \infty)$.

To examine (5.92), recall that $\bar{\alpha}$ and $\bar{T}$ were selected such that for all $q \in \big(\bar{B}^m_r(U^*)\backslash B^m_{\hat{\delta}}(U^*)\big) \times \mathbb{R}^{Nm} \times \mathbb{R}^n$,

$$\alpha\phi(\mu) \ge \alpha^2\Big(\frac{K}{\hat{\delta}} + K_\mu\Big)\|\mu - \mathbf{P}_{U^*}(\mu)\|^2 \tag{5.97}$$

whenever $\alpha (\in (0, \bar{\alpha})$ and $T \in (\bar{T}, \infty)$. For each point $\mu_o \in S_\mu$, μ_o is either inside $\bar{B}^m_r(U^*)\backslash B^m_{\hat{\delta}}(U^*)$, or μ_o is inside $\bar{B}^m_{\hat{\delta}}(U^*)$. If μ_o is inside $\bar{B}^m_{\hat{\delta}}(U^*)$, then it does not contradict the desired relationship (5.92). On the other hand, if μ_o is inside $\bar{B}^m_r(U^*)\backslash B^m_{\hat{\delta}}(U^*)$, then from its membership in S_μ (q.v. (5.89)) and the fact that it satisfies (5.97) implies that

$$\alpha^2\Big(\frac{K}{\hat{\delta}} + K_\mu\Big)\|\mu_o - \mathbf{P}_{U^*}(\mu_o)\|^2 - \alpha^2 K_\mu\|\mu_o - \mathbf{P}_{U^*}(\mu_o)\|^2 \le K, \tag{5.98}$$

which means that μ_o satisfies

$$\alpha^2\|\mu_o - \mathbf{P}_{U^*}(\mu_o)\|^2 \le \hat{\delta}^2.$$

Clearly then, μ_o belongs to $\bar{B}^m_{\hat{\delta}}(U^*)$ whenever $\alpha \in (0, \min\{\bar{\alpha}, \tilde{\alpha}_\mu\})$, where $\tilde{\alpha}_\mu = 1$.

In summary, the containment relationships (5.92), (5.93) and (5.94) hold whenever $\alpha \in (0, \min\{\bar{\alpha}, \alpha_Z\})$ and $T \in (\max\{\bar{T}, T_Z\}, \infty)$, where

$$\alpha_Z = \min\{\tilde{\alpha}_\mu, \tilde{\alpha}_\zeta, \tilde{\alpha}_z\}, \quad \text{and} \tag{5.99}$$

$$T_Z = \max\{\tilde{T}_\zeta, T_z\}, \tag{5.100}$$

thus demonstrating that P3 in Theorem 5.5.3 is also satisfied. □

5.6 Final Remarks

We refined the ideas introduced in [93], and proposed an application of numerical consensus optimization algorithms to the synthesis problem of decentralized coordination control for dynamic multiagent systems. Using the analytic framework developed in Chapter 3, we derived a set of conditions, involving relevant problem parameters, under which coordination control schemes involving the dynamic interaction of discrete-time consensus optimization algorithms and continuous-time dynamical systems are guaranteed to render stable behavior. To extend this work, we will consider problem settings in which the output $y_i = h(x) = J(z + l(u))$, and devise a strategy to estimate $s_i(u)$ from several measurements of y_i within a single iteration of $\mathscr{D}$. The first candidate for such a strategy is the finite difference method, as explored in [90].

Chapter 6

Reduced Consensus Optimization

6.1 Synopsis

In this brief chapter, we develop the method of *reduced consensus optimization* (RCO), which generalizes the discrete-time consensus optimization algorithm (1.33) by allowing agents to update only a subset of the optimization variables at each iteration. Based on this method, we propose a decentralized coordination control strategy for general networked multiagent systems involving static agents. The agents' individual goals are assumed to be given in terms of privately known objective functions, and the optimal network configuration is encoded as the optimizer of the sum of these functions. In implementing a coordination control strategy based on reduced consensus optimization, agents need not be aware of the overall network size or topology. Moreover, depending on the *interference structure*, the number of real-valued variables updated and exchanged among agents at each iteration may be substantially reduced relative to that of coordination control strategies based on consensus optimization.

This chapter is based primarily on [88]. The idea for the RCO algorithm arose from the author's work on the adaptive content-caching problem [88] and from her encounter with the notion of "partial overlaps" found in [154]. It appears that a related idea may have also been described in [163], albeit in the context of decomposition-based methods.

6.2 Introduction

Consider static DCCP given by Problem 1.2.3, and the CO-based solution (1.40) proposed in Example 1.4.1. Some favourable features of this solution include the fact that in order to implement (1.40), an agent does not require knowledge of other agents' actions, except those in its graphical neighborhood, and the fact that agent i need not be aware of other agents' private objectives or the collective objective. However, each agent needs to know how many other agents are on the network, since it must maintain an estimate of each component of the collective minimizer x^*. This is a cumbersome requirement; each time the network undergoes a structural modification such as the addition of a node, each preexisting node must modify its update rule to include additional real variables representing an estimate of the added node's optimal action. Moreover, in NMAS applications involving large networks in which the actions of graphically distant agents have a negligible influence on an agent's individual performance, having each agent maintain an estimate of the entire optimal network configuration x^* seems unnecessary, and excessive. To make this point clear, we provide the following simple example.

Example 6.2.1: Consider a DT-SDCCP involving three agents whose costs are of the form indicated in Figure 6.1. Suppose that agent i's decision variable is given by $x_i \in \mathbb{R}$, and that as in Example 1.4.1, agent i has access to the analytic structure of the gradient of his private cost, so that at any time he may evaluate this gradient anywhere within $\mathbb{R}^3$, regardless of the valuation of the other agents' current actions. If the agents are to coordinate their

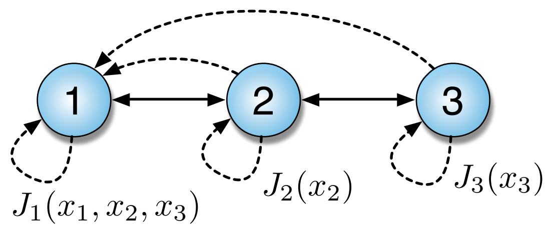

Figure 6.1: An example of a simple network of agents and its associated *interference digraph*. A dashed edge starting from node i and terminating at node j indicates that the actions of agent i interfere with the cost of agent j. The solid edges correspond to the communication graph $\mathscr{G}_C$.

actions in order to achieve the collective objective (q.v. P5 in §1.2.2) according to the CO-based decision updating process proposed in Example 1.4.1, they each update and exchange amongst themselves three variables at each iteration. For this example, one possible implementation of (1.40) is given by

$$\mathscr{D}_1 : \begin{cases} y_1 = \nabla_x J_1(x)|_{x=\xi_1} \\ \xi_{1,1}^+ = \frac{2}{3}\xi_{1,1} + \frac{1}{3}\xi_{2,1} - \alpha y_{1,1} \\ \xi_{1,2}^+ = \frac{2}{3}\xi_{1,2} + \frac{1}{3}\xi_{2,2} - \alpha y_{1,2} \\ \xi_{1,3}^+ = \frac{2}{3}\xi_{1,3} + \frac{1}{3}\xi_{2,3} - \alpha y_{1,3} \\ v_1 = \xi_1 = [\xi_{1,1}, \xi_{1,2}, \xi_{1,3}]^T \\ x_1 = \xi_{1,1} \end{cases} \tag{6.1}$$

$$\mathscr{D}_2 : \begin{cases} y_2 = \nabla_x J_2(x)|_{x=\xi_2} \\ \xi_{2,1}^+ = \frac{1}{3}\xi_{1,1} + \frac{1}{3}\xi_{2,1} + \frac{1}{3}\xi_{3,1} - \alpha y_{2,1} \\ \xi_{2,2}^+ = \frac{1}{3}\xi_{1,2} + \frac{1}{3}\xi_{2,2} + \frac{1}{3}\xi_{3,2} - \alpha y_{2,2} \\ \xi_{2,3}^+ = \frac{1}{3}\xi_{1,3} + \frac{1}{3}\xi_{2,3} + \frac{1}{3}\xi_{3,3} - \alpha y_{2,3} \\ v_2 = \xi_2 = [\xi_{2,1}, \xi_{2,2}, \xi_{2,3}]^T \\ x_2 = \xi_{2,2} \end{cases} \tag{6.2}$$

$$\mathscr{D}_3 : \begin{cases} y_3 = \nabla_x J_1(x)|_{x=\xi_3} \\ \xi_{3,1}^+ = \frac{1}{3}\xi_{2,1} + \frac{2}{3}\xi_{3,1} - \alpha y_{3,1} \\ \xi_{3,2}^+ = \frac{1}{3}\xi_{2,2} + \frac{2}{3}\xi_{3,2} - \alpha y_{3,2} \\ \xi_{3,3}^+ = \frac{1}{3}\xi_{2,3} + \frac{2}{3}\xi_{3,3} - \alpha y_{3,3} \\ v_3 = \xi_3 = [\xi_{3,1}, \xi_{3,2}, \xi_{3,3}]^T \\ x_3 = \xi_{3,3}. \end{cases} \tag{6.3}$$

Figure 6.1 indicates that the actions of agent 1 have no effect on the cost of either agent 2 or agent 3, and yet both agents 2 and 3 update variables that may be interpreted as their "suggestions" to agent 1 on how to behave. Based on the cost structure indicated in Figure 6.1, we notice that $y_{2,1} = y_{3,1} = 0$.

As an alternative solution to this DT-SDCCP, consider the following set of decision update rules.

$$\mathscr{D}_1 : \begin{cases} y_1 = \nabla_x J_1(x)|_{x=\xi_1} \\ \xi_{1,1}^+ = \xi_{1,1} - \alpha y_{1,1} \\ \xi_{1,2}^+ = \frac{1}{2}\xi_{1,2} + \frac{1}{2}\xi_{2,2} - \alpha y_{1,2} \\ \xi_{1,3}^+ = \frac{2}{3}\xi_{1,3} + \frac{1}{3}\xi_{2,3} - \alpha y_{1,3} \\ v_1 = [\xi_{1,2}, \xi_{1,3}]^T \\ x_1 = \xi_{1,1} \end{cases} \tag{6.4}$$

$$\mathscr{D}_2 : \begin{cases} y_2 = \nabla_x J_2(x)|_{x=\xi_2} \\ \xi_{2,2}^+ = \frac{1}{2}\xi_{1,2} + \frac{1}{2}\xi_{2,2} - \alpha y_{2,2} \\ \xi_{2,3}^+ = \frac{1}{3}\xi_{1,3} + \frac{1}{3}\xi_{2,3} + \frac{1}{3}\xi_{3,3} \\ v_2 = [\xi_{2,2}, \xi_{2,3}]^T \\ x_2 = \xi_{2,2} \end{cases} \tag{6.5}$$

$$\mathscr{D}_3 : \begin{cases} y_3 = \nabla_x J_1(x)|_{x=\xi_3} \\ \xi_{3,3}^+ = \frac{1}{3}\xi_{2,3} + \frac{2}{3}\xi_{3,3} - \alpha y_{3,3} \\ v_3 = \xi_{3,3} \\ x_3 = \xi_{3,3}. \end{cases} \tag{6.6}$$

The collective decision updating process in this case involves 6 real-valued variable updates, and a total of 6 real-valued variable exchanges among the agents. By comparison, the collective decision updating process in (6.1) to (6.3) involves 9 real-valued variable updates, and a total of 12 real-valued variable exchanges among the agents.

Figure 6.2 shows the outcome of forty iterations of the decision update rule (6.4)–(6.6), when the cost functions are given by

$$\begin{aligned} J_1(x) &= 6(x_1 + x_3 - 15)^2 + 4(x_2 - 4)^2 + (x_3 - 10)^2 \\ J_2(x) &= 5(x_2 + 2)^2 \\ J_3(x) &= (x_3 + 8)^2, \end{aligned}$$

the step size $\alpha = 0.01$, and the initial conditions are taken as $\xi_1(0) = (2, -3, 17)$, $(\xi_{2,2}(0), \xi_{2,3}(0)) = (-12, 15)$ and $\xi_{3,3}(0) = 0$. For these costs, the optimal network configuration is $x^* = [6, 0.67, 9]^T$. $\diamondsuit$

Motivated by these considerations, we now develop the *reduced consensus optimization* (RCO) algorithm, in which agent i does not necessarily need to maintain an estimate of x_j^*, if agent j's actions have no effect on $J_i(\cdot)$.

6.3 Problem Setting

In this chapter we deal exclusively with static problems. In accordance with Remark 1.2.7, we therefore omit reference to either Σ or u, and instead denote agent i's decision variable by $x_i \in \mathbb{R}^{n_i}$. The collective decision is denoted by $x \in \mathbb{R}^n$.

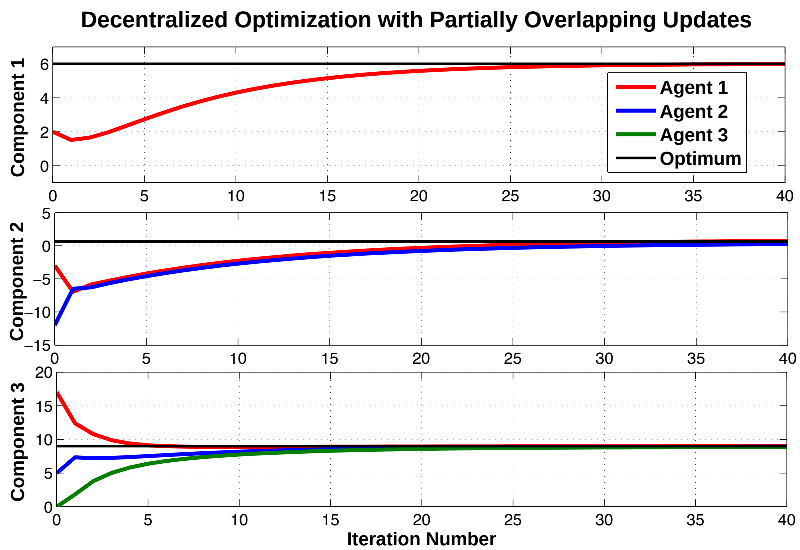

Figure 6.2: The reduced consensus optimization algorithm (6.4)–(6.6) converges to the collective optimum in forty iterations.

We consider DT-SDCCPs involving a set of N static agents, whose task is to cooperatively solve the problem

$$\min_{x\in\mathbb{R}^n} J_1(x)+\cdots+J_N(x), \tag{6.7}$$

when agent i has knowledge only of the function $J_i(\cdot)$. The agents may exchange information over the communication graph $\mathscr{G}_C=(\mathscr{E}_C,\mathscr{V})$, where $\mathscr{V}$ is the set that indexes the agents, and $\mathscr{E}_C\subset\mathscr{V}\times\mathscr{V}$ specifies those agent pairs that can exchange information.

For notational simplicity, we assume that $\mathscr{G}_C$ is connected and undirected, and that the collective cost

$$J(x)=\sum_{i\in\mathscr{V}} J_i(x) \tag{6.8}$$

is strictly convex and bounded from below, ensuring the existence of a unique, finite minimizer x^*.

6.4 Reduced Consensus Optimization Algorithm

We first describe the algorithm, and then discuss the intuition behind it. Let $\mathscr{G}_I=(\mathscr{V},\mathscr{E}_I)$ be the network's *interference digraph*, whose edge set is defined as

$$\mathscr{E}_I=\{(j,i)\in\mathscr{V}\times\mathscr{V}\mid\nabla_{x_j}J_i(x)\not\equiv 0\}. \tag{6.9}$$

In other words, (j,i) is an edge in the interference digraph if the jth component of the optimization variable x appears in the ith objective. If x_j represents the action taken by agent j, then (j,i) is an edge in the interference digraph if the actions of agent j influence the cost experienced by agent i. It should be noted that we do not assume any particular relationship between the communication structure $\mathscr{G}_C$, and the interference structure $\mathscr{G}_I$.

Next, for each $j\in\mathscr{V}$, we define $I(j)$ as the set of all agents $i\in\mathscr{V}$ whose costs are affected by the actions of agent j – i.e.,

$$I(j) = \{i \in \mathscr{V} \mid (j,i) \in \mathscr{E}_I\} \cup \{j\}. \tag{6.10}$$

For each $j \in \mathscr{V}$, we then form the graph $\mathscr{G}_j = (\mathscr{V}_j, \mathscr{E}_j)$, which we define to be any smallest connected subgraph of $\mathscr{G}_C$ containing all the nodes in $I(j)$. Specifically, suppose that G denotes the set of all connected subgraphs (q.v. §2.1) of $\mathscr{G}$ containing the vertices in $I(j)$. Then $\mathscr{G}_j$ is any element in G with the property that for any other element $\mathscr{G}_j' = (\mathscr{V}_j', \mathscr{E}_j')$, we have that $|\mathscr{V}_j| \leq |\mathscr{V}_j'|$. Constructing $\mathscr{G}_j$ is always possible since $\mathscr{G}_C$ is assumed to be connected. However, the choice of $\mathscr{G}_j$ may not be unique, since the shortest path connecting any $i,k \in I(j)$ may not be unique. We note that in general, $\mathscr{E}_j \subseteq \mathscr{E}_C$ and that $I(j) \subseteq \mathscr{V}_j \subseteq \mathscr{V}$ (i.e., $\mathscr{V}_j$ does not necessarily coincide with $I(j)$, as shown in Example 6.4.1). We let

$$\mathscr{N}_j(i) = \{k \in \mathscr{V}_j \backslash \{i\} \mid (k,i) \in \mathscr{E}_j\}$$

denote the set of agent i's neighbors on $\mathscr{G}_j$. Finally with the set

$$S(i) = \{j \in \mathscr{V} \mid i \in \mathscr{V}_j\}, \tag{6.11}$$

we identify those subgraphs $\mathscr{G}_j$ to which agent i belongs.

Then, the (unconstrained) RCO algorithm is implemented by each agent $i \in \mathscr{V}$ as follows:

$$\begin{aligned} \xi_{i,j}(t+1) &= \begin{cases} \sum_{k=1}^N a_{j,k}^{(i)} \xi_{k,j}(t) - \alpha \nabla_{x_j} J_i(\xi_i(t)), & j \in S(i) \\ 0 & j \in \mathscr{V} \backslash S(i) \end{cases} \\ x_i(t) &= \xi_i(t), \end{aligned} \tag{6.12}$$

where ξ_i represents agent i's estimate of the optimal collective decision x^*, $\xi_{i,j}$ his "opinion" of agent j's optimal action,

$$a_{j,k}^{(i)} = \begin{cases} \min\left\{\frac{1}{|\mathscr{N}_j(i)|+1}, \frac{1}{|\mathscr{N}_j(k)|+1}\right\}, & \text{if } k \in \mathscr{N}_j(i) \\ 1 - \sum_{m=1}^N a_{j,m}^{(i)}, & \text{if } k = i \text{ and } i \in \mathscr{V}_j, \\ 0 & \text{otherwise,} \end{cases} \tag{6.13}$$

and α is a sufficiently small, fixed step-size.

The main difference between (6.12) and the standard (unconstrained) CO algorithm (1.33) in the DCCP context, is that agent i need not maintain estimates of *all* other agents' optimal actions. Indeed, when the interference structure is sparse, the set $S(i)$ in (6.12) could have a much smaller cardinality than $\mathscr{V}$. In RCO, agent i updates an estimate of precisely those components of x^* associated to all agents $j \in S(i)$. Conversely, the set of agents that estimate j's optimal action x_j^* is given by $\mathscr{V}_j$.

Remark 6.4.1: In order to implement algorithm 6.12, agent i generally need not be aware of the size N of the network, its overall topology $\mathscr{G}_C$, or its overall interference digraph $\mathscr{G}_I$. However, he must have knowledge of the set $S(i)$. For the case in which $\mathscr{V}_j = I(j)$, $\forall j \in \mathscr{V}$, this is tantamount to knowing which other agents in the network are affecting his private cost. In cooperative DCCP scenarios such as those that we consider in this thesis, it may be possible to design some form of "discovery" protocol that the agents execute initially in order to discern the sets $S(i)$. Moreover, in applications in which the costs themselves are subject to design, agents may simply be equipped a priori with this requisite knowledge. In Chapter 7, we explore one such application. $\diamondsuit$

Remark 6.4.2: Though we focus on DT-SDCCPs in this chapter, the formulation of the RCO algorithm is equally applicable in the context of dynamic DCCPs. $\diamondsuit$

In the following section, we work through an example in order to show how the various sets introduced in this section can be identified.

6.4.1 Some Intuition

The intuition behind RCO is that each subgraph $\mathscr{G}_j$ has a separate consensus matrix $A_j \in \mathbb{R}^{|\mathscr{V}_j|\times|\mathscr{V}_j|}$ associated to it, and there are N concurrent, dynamically coupled consensus optimization processes taking place. Subgraph $\mathscr{G}_j$ connects all those agents whose individual costs are affected by the actions of agent j. The weights $a_{j,k}^{(i)}$ pertaining to the jth subgraph are assigned according to a well-known algorithm (q.v. [167], §III-B in [114], or Algorithm 2.3.1 in §3.3), which ensures that A_j satisfies certain technical conditions that suffice for asymptotic consensus in (6.12). These technical conditions are discussed in greater detail in §3.3.

The following example demonstrates how the various sets and subgraphs defined in the previous section can be identified, and how the link weights $a_{j,k}^{(i)}$ may be assigned.

Example 6.4.1: Consider a network of four nodes shown in Figure 6.3. Each node corresponds to an agent whose private cost is indicated next to the node. Figure 6.3 (b) indicates the interference digraph induced by these costs.

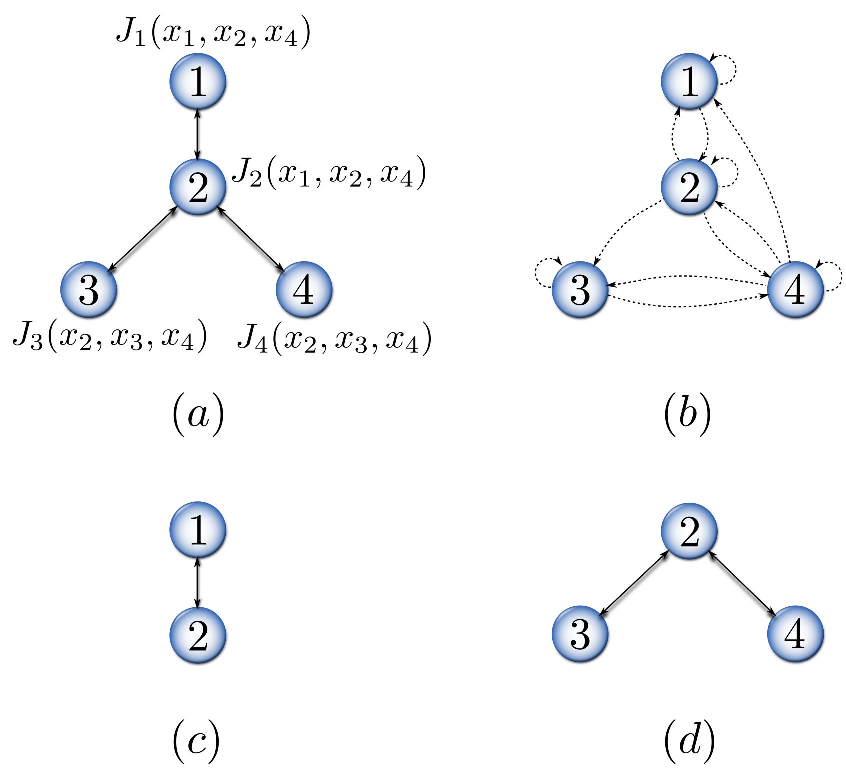

Figure 6.3: A network with $\mathscr{G}_C$ and agents' individual costs shown in (a), and the associated interference structure shown in (b). The subgraphs $\mathscr{G}_2 = \mathscr{G}_4 = \mathscr{G}_C$, whereas $\mathscr{G}_1$ and $\mathscr{G}_3$ are shown in (c) and (d), respectively. Each subgraph $\mathscr{G}_j$ is constructed by including all the nodes and edges along a shortest path from node j to each node i, whose cost is affected by the actions of node j. In this case $S(1) = S(2) = \{1,2,4\}$ and $S(3) = \{2,3,4\}$, as expected, since nodes 1, 2 and 4 interfere with the costs of nodes 1 and 2 and nodes 2, 3 and 4 affect J_3. However, $S(2) = \{1,2,3,4\}$ even though $\nabla_{x_3} J_2(x) \equiv 0$; the inclusion of node 3 in $S(2)$ is necessary since node 2 must indirectly act as a "conduit" for information passing between nodes 3 and 4, who do interfere with one another.

The interference digraph shown in Figure 6.3 (b) corresponds to the following interference sets defined in

(6.10):

$$I(1) = \{1,2\}, \qquad I(2) = \{1,2,3,4\}$$
$$I(3) = \{3,4\}, \qquad I(4) = \{1,2,3,4\}.$$

The smallest subgraphs $\mathscr{G}_1$ and $\mathscr{G}_3$ containing the sets $I(1)$ and $I(3)$ respectively, are shown in Figure 6.3 (c) and (d), while $\mathscr{G}_2 = \mathscr{G}_4 = \mathscr{G}_C$. Recalling that the set $S(i)$ defined in (6.11) identifies those subgraphs to which node i belongs, we observe that

$$S(1) = \{1,2,4\}, \qquad S(2) = \{1,2,3,4\}$$
$$S(3) = \{2,3,4\}, \qquad S(4) = \{2,3,4\}.$$

In this example, $\mathscr{V}_j = I(j)$ for all j except $j = 2$. Since $S(i)$ identifies those components of x^* that agent i estimates through the RCO update rule (6.12), we note that node 2 must maintain an estimate of x_3^* – not because node 3 affects its cost, but because node 2 must act as an information conduit between nodes 3 and 4, who interfere with one another.

The values of the weights $a_{j,k}^{(i)}$ for the example shown in Figure 6.3 (a) may be chosen as shown in Figure 6.4. These weights are obtained using expression (6.13), or equivalently, by applying Algorithm 2.3.1 from Chapter 3

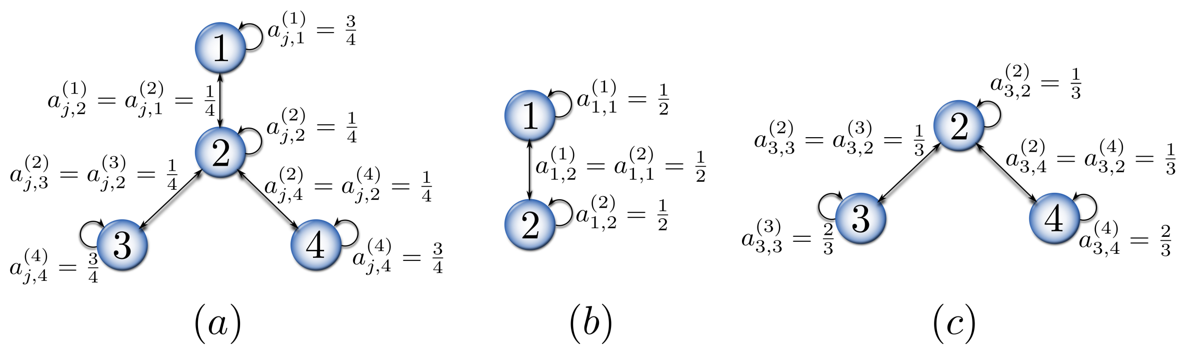

Figure 6.4: Assigning the weights $a_{j,k}^{(i)}$ to each subgraph $\mathscr{G}_j$, according to (6.13). (a) Edge (and self-loop) weights for the subgraphs $\mathscr{G}_j$, $j \in \{2,4\}$. (b) Subgraph $\mathscr{G}_1$. (c) Subgraph $\mathscr{G}_3$.

to each subgraph $\mathscr{G}_j$. $\diamondsuit$

6.5 Convergence and Stability

In Chapter 3 we provide a convergence and stability analysis of consensus-optimization methods in more general settings involving constrained convex (i.e. not strictly convex) problems and generalized search directions. The analytic techniques presented there can be adapted to the RCO case, and we omit such an analysis here. The outcome of such an analysis would be the derivation of an upper bound on the step-size α in terms of relevant problem parameters, and the conclusion that whenever α satisfies this bound, the point

$$\hat{x}^* = \begin{bmatrix} \mathbf{1}_{|\mathscr{V}_1|} \otimes x_1^* \\ \vdots \\ \mathbf{1}_{|\mathscr{V}_N|} \otimes x_N^* \end{bmatrix} \tag{6.14}$$

is *semiglobally, practically, asymptotically stable* for (6.12) (q.v. Definition 2.5.3), where $x^* = [{x_1^*}^T, \ldots, {x_N^*}^T]^T$ solves problem (6.7).

6.6 Final Remarks

In the design of decentralized coordination strategies for general NMAS in which the interference digraph $\mathscr{G}_I$ is not complete, the RCO algorithm (6.12) allows us to realize two important benefits when compared to consensus optimization (1.33). First, agent i need not be aware of any agent $j \in \mathscr{V} \backslash S(i)$. In other words, whenever $S(i) \neq \mathscr{V}$, agent i is *nescient* with respect to the number of agents on the network N, and the network's overall interconnection structure $\mathscr{G}_C$. As a consequence, agent i's update rule does not change when nodes are added or taken out of the network, so long as the set $S(i)$ is unaffected by the change. Second, for NMAS with sparse interference structures (for example those with $|S(i)| \ll N$, for most $i \in \mathscr{V}$), RCO requires drastically fewer real-number updates and exchanges among the agents at each iteration. RCO thereby reduces the communication and processing overhead associated with coordinating the agents. In the following chapter (q.v. §7.5.2), we quantify this reduction for an RCO-based content-caching strategy designed for the four-node example network shown in Figure 6.3.

Chapter 7

Decentralized Content Caching Strategies for Content-Centric Networks

7.1 Synopsis

In this chapter we consider the problem of energy-efficient Internet content delivery over networks in which individual nodes are equipped with content caching capabilities. We cast this problem as an instance of the class of DT-SDCCPs represented by Problem 1.2.3. We present a flexible, yet systematic methodology for the design of cooperative, decentralized caching strategies that can adapt to real-time changes in regional content popularity. The methodology is based on the reduced consensus optimization method described in Chapter 6. The outcome of the design is a set of dynamic update rules that stipulate how much and which portions of each content piece an individual network node ought to cache. In implementing these update rules, the nodes achieve a collectively optimal caching configuration through nearest-neighbor interactions and measurements of local content request rates only. The desired caching behavior is encoded in the design of individual nodes' costs, and can incorporate a variety of network performance criteria. We focus on the goal of minimizing the energy consumption of the network as a whole, designing a network cost whose minimum achieves a trade off between transport and caching energy costs in response to changes in content demand. We assess the performance of the proposed scheme along several metrics, against the prevalently deployed "least frequently used" cache eviction policy.

7.2 Introduction

A significant portion of today's internet traffic involves the delivery of content such as video to a multitude of geographically distributed users who are typically indifferent to where that content is stored or accessed from. According to CISCO's 2012 VNI report [68], video streaming and downloads presently account for over 86% of all internet traffic, and services such as IPTV and Video-on-Demand (VoD) constitute the fastest growing internet service class. Current trends toward ubiquitous mobile computing suggest that Internet traffic will continue to be dominated by the distributed on-demand consumption of video content.

The problem of content dissemination to a distributed set of users is ideally addressed by some form of multicasting, whereby each piece of data is delivered through a single transmission from the source, and copies are created only at branch points in the distribution tree. Multicast technologies such as IP and application layer multicast are well-suited for the delivery of real-time multimedia services such as IPTV and live video streaming, in which requests for the same data are served simultaneously. However, the efficiency of multicasting cannot be directly exploited for the delivery of on-demand services such as VoD and time-shifted TV, in which content requests arrive asynchronously. When the same piece of data is accessed at different times from multiple locations, caching the data temporarily at intermediate nodes can enable multicast delivery, significantly reducing load at origin severs, access latency and network congestion [105]. It is therefore widely accepted that caching is essential to enabling on-demand content access. [105], [23], [135], [83], [70], [20].

Several caching-based content distribution technologies are already extensively deployed, including privately owned content delivery networks (CDNs) such as Akamai and Limelight, Peer-to-Peer (P2P) systems such as BitTorrent, and web caching solutions such as Squid and NetCache. In all cases, the central idea is that replicating content and caching it throughout the network facilitates the realization of important performance benefits such as reduced network congestion and server loading, reduced access latencies, and improved tolerance to transport disruptions. However, these technologies are typically incompatible with one another, and they are overlay solutions implemented atop a host-centric Internet which was never intended for the mass distribution of content. As such, they often result in the suboptimal utilization of network resources [160], [38].

The Internet's fundamental incompatibility with today's content consumption trends has therefore prompted research into a new networking paradigm known as *information-centric networking* (ICN) [4], [70], in which content is directly accessed by name, rather than the address of its host. In an information-centric network, content can be delivered from any network location that caches a valid copy of that content, data packets can be transparently cached as they travel toward their consumers, and content can be assembled at its destination from data packets that may arrive from multiple locations. Compared to existing overlay solutions, ICN architectures are therefore more naturally poised to leverage in-network caching [70], [126].

The use of in-network caching has also been investigated as a possible means of reducing a network's energy consumption [95], [60], [33], [98]. Although information and communication technologies (ICTs) are currently estimated to account for only 2% of the world's total carbon footprint, this proportion is expected to grow as ICT energy efficiency improvements plateau due to fundamental theoretical and physical limitations [81]. The aggressive growth in the number of users and the variety of demanded services motivate research into new ways of reducing the energy consumption of ICTs.

Regardless of the adopted technology (i.e., P2P, CDNs, or ICN), the extent to which the benefits of in-network caching can be realized depends crucially on the efficacy of the implemented *content caching strategy* (CCS). A CCS is a set of policies or algorithms that prescribe how much of what content each participating network node ought to cache. These may include various file placement and eviction policies, as well as protocols that ensure the consistency of content replicas [123].

An "effective" CCS is one that is *decentralized*, *adaptive* to real-time changes in regional content demand patterns, and allows individual nodes to be *nescient* with regard to the operation of the collective. In a decentralized CCS, network nodes collectively achieve a desirable caching configuration by individually making independent caching decisions based on local network measurements and interactions with their nearest neighbors. The difficulties associated with implementing centralized coordination schemes in large networks obviate the need for decentralization; the acquisition of network measurements by a central node and the subsequent dissemination of coordination signals to each node consume transport resources, while communication delays accumulated in transmitting these signals over several hops can adversely affect the stability and robustness of any coordination scheme. Nodes are nescient if they require no a priori knowledge in order to execute the CCS. The extent to which this property applies determines how flexible and modular the network design can be. Decentralized content caching strategies with nescient nodes allow the network to undergo structural changes such as node additions and deletions, without requiring all nodes to be reprogrammed every time such changes occur.

7.2.1 Contributions and Related Literature

Although there is an appreciable effort within the research community to develop various networking architectures and implementation-level mechanisms that enable in-network caching, there has been little attention devoted to the development of content caching strategies with the network-level view of optimizing the use of network resources on the whole.

Existing work addressing the design of CCSs includes [23], [74], [28], [83], [73], [20], [42], [32], [126], [98] and [96], among others.

Many papers restrict their attention to the performance of individual caches [23], [74], or network substructures such as trees or paths from sources to consumers [28], [126], [83], [73], [20]. Although the focus on tree substructures is appropriate for en-route web-caching schemes developed for IP networking, it may lead to CCS designs that fail to leverage all of the operational features offered by ICN architectures such as content-centric networking (CCN) [70]. For example, the packetization of content in CCN allows for content to be partially requested and stored; as mentioned in [70], a destination node in a meshed content-centric network need not be restricted to assembling requested content from data packets cached by nodes that lie solely along a path toward the origin server. With appropriate interest broadcasting techniques in place, transport inefficiencies such as those described in [42] can thus be avoided.

Most approaches to the design of CCSs are based on variations of file placement policies (such as fixed probability caching), in combination with standard file eviction policies (such as least recently used or least frequently used), which are concepts borrowed from the literature on web caching [123]. Although simple to implement, such designs are heavily based on heuristics and the optimality of the collective behavior of caches in a general topology network is difficult to guarantee [96], [32], [126]. Sometimes these policies are tuned to optimize performance based on network models or simulations that make use of simplified traffic predictions, which may not reflect actual traffic patterns once the CCS is deployed [96], [20].

Some CCS design approaches guarantee a quantifiably suboptimal performance only in the case that certain symmetry assumptions are satisfied (such as all caches having the same size, and all content demand rates being equal at each node) [20]. Others require nodes to know or estimate the operational parameters (such as cache size) of other nodes in the network, and the efficacy with which the caching resources are utilized is known to be affected by the accuracy of such estimates [126].

In this chapter we propose a broadly applicable methodology for the design of decentralized, adaptive CCSs that systematically improve the efficiency with which a network's caching and transport resources are utilized. Although the CCSs developed here can be adapted to content delivery technologies such as CDNs and web caching, they are designed to leverage the operational features of networks with content-aware forwarding capabilities, such as those found in CCN.

Elaborating on our work in [89], we base our CCS designs on a provably convergent, decentralized optimization algorithm called *reduced consensus optimization* (RCO), which was initially proposed in [88]. An exposition of RCO can also be found in Chapter 6 of this thesis. In RCO, a number of nodes on a connected graph cooperate in minimizing the overall *network cost*, which encodes a set of network-wide performance objectives. This network cost is comprised as the sum of nodes' individual, privately known cost functions. There is no special structure imposed on the network topology, and the nodes are nescient with respect to the structure and size of the overall network. In particular, while most existing CCS designs consider special network structures such as trees, CCSs based on RCO allow sources, destinations and intermediary routers to be interconnected in general mesh topologies (q.v. Figure 7.1).

We focus on the problem of reducing the energy consumed by the network's transport and storage resources. Adopting the so-called "energy-proportional computing" model [10], we characterize the optimal network caching configuration as a minimizer of the network cost function. Each node's individual cost function depends on local content demand rates (measured in real time), and embodies the basic tradeoff induced by the energy-proportional computing model: caching more content locally may reduce long-distance data transport costs at the expense of increased caching costs, and vice versa. In this way, the optimal network caching configuration depends on the intensity and regional distribution of demand for content at any given time. In implementing the proposed CCS, individual nodes dynamically adjust how much, and which portions of each (popular) content piece each stores, in response to their individual costs and the caching decisions made by neighboring nodes. Through nearest-neighbor interactions, the nodes collectively balance the goal of minimizing excessive copying of content throughout the network, with the goal of minimizing redundant transport of data over long distances. Although we focus on energy efficiency, the proposed design methods can explicitly incorporate other network performance objectives as well.

We assume that individual nodes are heterogeneous with respect to their caching capacities and energy efficiency parameters; to execute the CCS, a node does not need to know or estimate these characteristics as they pertain to other nodes in the network.

This chapter is organized as follows. After specifying the adaptive content caching problem in §7.3, we consider the goal of minimizing the energy consumed by a network's transport and caching resources in §7.4. We propose a systematic method for the design of nodes' individual objectives based on the notions of *segmentation* and *clustering*, which we introduce in §7.4.1. A detailed design example demonstrates the application of the proposed design methodology in §7.5. The design method is also applied to the eleven-node European optical backbone network, COST239, and the performance of the resulting content-caching strategy is compared along several metrics against that of the well-known "least frequently used" (LFU) cache eviction policy in §7.6.

7.3 A Description of the Adaptive Content Caching Problem

We consider an abstract network of N nodes that may exchange information over a given connected, undirected graph $\mathscr{G}_C = (\mathscr{V}, \mathscr{E}_C)$, where $\mathscr{V} = \{1, \ldots, N\}$ is the set indexing the nodes and $\mathscr{E}_C \subset \mathscr{V} \times \mathscr{V}$ is the set of links between them.

We assume that there is a set of content repository nodes (or sources) that generate and store permanent copies of a large number of files that may need to be accessed throughout the network, as shown in Figure 7.1. This set of source nodes may include some of the nodes within $\mathscr{G}_C$, or it may be entirely external to $\mathscr{G}_C$. However, all nodes within $\mathscr{G}_C$ are able to access any file stored at any source node.

We let $\mathscr{F} = \{1, \ldots, F\}$ denote the set of F most popular content pieces that are to be cached throughout the network. Each content piece may represent a single file, or an aggregate of related files with similar popularities. At each instant, the ith node in the network experiences a demand for content piece $k \in \mathscr{F}$ that can be calculated as a windowed average request rate – i.e., supposing that $(t_{n_k^i})_{n_k^i=1}^{\infty} \subset \mathbb{R}_+$ denotes the sequence of time instants at which requests for content piece k arrive at node i, the demand for k at node i can be calculated as

$$d_{i,k}(t) = \frac{1}{W} \int_{t-W}^{t} \sum_{n_k^i=1}^{\infty} \delta(t - t_{n_k^i}) dt, \tag{7.1}$$

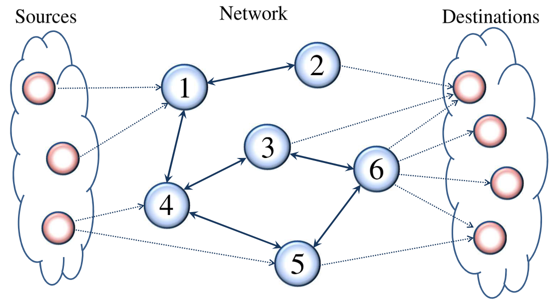

Figure 7.1: Each node in the network can access files from the source repositories. In general, the source nodes, the destination nodes and the intermediary routers can be interconnected over a general mesh topology. The ith node measures a demand $d_{i,k}$ for content piece k.

where $\delta(\cdot)$ is the Dirac delta function, and W is the window size in units of time. Intuitively, for a given traffic pattern observed at node i, selecting larger values for W yields a set of signals $d_{i,k}(t)$ exhibiting less volatility.

The signal $d_{i,k}$, which can be measured by node i, represents an aggregated request rate that originates either from a pool of users directly connected to node i, or from other nodes within the network (q.v. Figure 7.1).

We assume that each content piece is divided into P_k packets of q data units in size and that each node $i \in \mathscr{V}$ is equipped with a cache of size $B_i q$ units of data. Moreover, these packets can be individually requested from the source repository or other nodes. This mechanism allows each node to store selected portions of various content pieces, if desired.

Given a communication graph $\mathscr{G}_C = (\mathscr{V}, \mathscr{E}_C)$, the content catalog $\mathscr{F}$, content sizes P_k, $\forall k \in \mathscr{F}$, cache sizes B_i, $\forall i \in \mathscr{V}$, and the set of demand rates $d_{i,k}(t)$, the adaptive content-caching problem is to decide which portions of which content pieces each node ought to cache at time t. One straightforward way to encode such a decision is to assume that nodes store only contiguous blocks of a content piece. In that case, the decision can be characterized in terms of only two numbers: $\sigma_{i,k} \in \{\frac{1}{P_k}, \ldots, \frac{P_k-1}{P_k}, 1\}$, indicating the location at which node i *starts* to store its contiguous block of $k \in \mathscr{F}$, and $\phi_{i,k} \in \{\frac{1}{P_k}, \ldots, \frac{P_k-1}{P_k}, 1\}$, the *fraction* of the whole content piece that this block represents. Since $\forall k \in \mathscr{F}$, P_k is likely a large integer, we may approximate $\sigma_{i,k}$ and $\phi_{i,k}$ by allowing them to take values in the real unit interval, as shown in Figure 7.2.

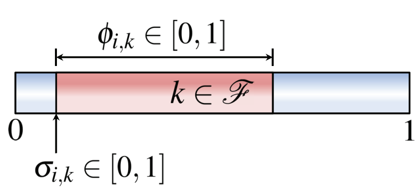

Figure 7.2: Router i stores a contiguous block of content piece k, starting at $\sigma_{i,k}$, and having a size of $\phi_{i,k} P_k q$ units of data.

Collecting these variables, we let

$$x_i = [\sigma_{i,1}, \phi_{i,1}, \ldots, \sigma_{i,F}, \phi_{i,F}]^T \in \mathbb{R}^{2F}, \quad \forall i \in \mathscr{V} \tag{7.2}$$

denote the vector of node i's caching decisions, and we refer to $x = [x_1^T, \ldots, x_N^T]^T \in \mathbb{R}^{2NF}$ as the *network caching configuration*. In relation to our formulation of the DT-SDCCP in Chapter 1, the variable x_i corresponds to agent i's decision variable, while x corresponds to the collective decision.

With this framework, the adaptive content caching problem can be rephrased as follows: what constitutes the best network caching configuration? What is deemed "best" depends on the set of network performance objectives, which may include factors such as minimization of network congestion and improvement of load balancing, reduction of access latency, enforcement of various QoS measures for different classes of content, robustness to transport disruptions and node failures, elimination of redundant traffic flows, minimization of energy costs, and others. Such performance objectives can often be expressed within the formalism of convex optimization. We therefore assume that there exists a network caching configuration x^* that best meets a given set of network performance criteria, and that x^* can be expressed as a solution to an optimization problem of the form

$$
\begin{aligned}
x^* = \arg\min_{x\in\mathbb{R}^{2NF}} \quad & J(x) && && (7.3)\\
\text{s.t.} \quad & \sigma_{i,k} \geq 0, && \forall i \in \mathscr{V},\ \forall k \in \mathscr{F} && (7.4)\\
& \phi_{i,k} \geq 0, && \forall i \in \mathscr{V},\ \forall k \in \mathscr{F} && (7.5)\\
& \sigma_{i,k} + \phi_{i,k} \leq 1, && \forall i \in \mathscr{V},\ \forall k \in \mathscr{F} && (7.6)\\
& \sum_{k=1}^{F} \phi_{i,k} P_k \leq B_i, && \forall i \in \mathscr{V}, && (7.7)
\end{aligned}
$$

where (7.7) represents node i's cache capacity constraint, and inequalities (7.4)-(7.6) are required by the manner in which the decision variables are defined (q.v. Figure 7.2). Further, we assume that the function $J : \mathbb{R}^{2NF} \to \mathbb{R}$ can be written as

$$J(x) = J_1(x) + \cdots + J_N(x), \tag{7.8}$$

with $J_i(x)$ representing node i's individual performance loss. Then, letting

$$X_0 := \left\{ x_i \in \mathbb{R}^{2F} \;\middle|\; \sigma_{i,k} \geq 0,\ \phi_{i,k} \geq 0,\ \sigma_{i,k} + \phi_{i,k} \leq 1,\ \ \forall k \in \mathscr{F} \right\}, \tag{7.9}$$

and

$$X_i := X_0 \cap \left\{ x_i \in \mathbb{R}^{2F} \;\middle|\; \sum_{k\in\mathscr{F}} \phi_{i,k} P_k \leq B_i \right\}, \tag{7.10}$$

we observe that equations (7.3) to (7.7) have the form (1.31), and that we may therefore employ the CO algorithm (1.33) in order to locate x^* in a decentralized manner. We also note that this formulation of the decentralized content caching problem belongs to the class of DT-SDCCPs represented by Problem 1.2.3, with $u \equiv x$ and the exogenous environmental conditions $d(t)$ (described in P5 of §1.2.2) corresponding to the demand rates $d_i(t)$ measured locally by each node in the network.

In this chapter, we choose to focus on a set of network performance objectives that are described in the sequel.

7.4 Energy-Efficient Content Delivery

Building on the work in [95], [60], [33] and [98], in the sequel we consider the minimization of the network's energy consumption as the primary performance objective. As in [95], [33], [60] and [98], we adopt the *energy-proportional computing model* [10], in which it is assumed that the energy consumption of network transport and routing equipment is proportional to its utilization. In [95], [33], [60] and [98], the transport resources considered include transmission, routing and switching equipment.

In adopting this model, we approximate the amount of energy consumed by a node as being proportional to the amount of data it caches, in addition to the amount of data it transports from other nodes toward the consumer. Depending on whether a node is a core, edge or access router, its energy efficiency profile is expected to be different; we account for this heterogeneity by allowing the costs to depend on $E_{ca,i}$, the amount of power (in Watts) that node i consumes by caching q bits of data, and $E_{tr,i}$, the amount of energy (in Joules) that node i consumes by transporting q bits of data.

Though the energy-proportional computing model is not without criticism (since devices tend to consume some amount of energy even when idling), studies such as those in [159] and [33] suggest that its adoption suffices for our purposes.

In the context of content distribution, the energy-proportional model induces a basic performance trade-off: while transport energy is reduced by caching as much content as possible close to consumer demand, caching more content consumes more caching energy. To capture this trade-off, we propose that individual agents' caching behavior should be governed by the following set of operational principles:

Operational Principles

1. The fraction $\phi_{i,k}(t)$ of content piece k that node i caches should be positively correlated with $d_{i,k}(t)$ – i.e., node i should cache more of the content that is in high demand, and less of the content that is not in demand.

2. To avoid excessive copying of content within the network, node i should avoid caching the same portions of the same content that nearby nodes cache.

3. When $d_{i,k}(t)$ is high, some nodes in i's graphical vicinity should coordinate among themselves to maximize their collective caching coverage of content piece k, so as to make those portions not cached by i available for short-distance transport to i.

These operational principles are to be encoded into the update rules governing individual agents' caching decisions by means of appropriately designed individual cost functions $J_i(\cdot)$. There are many degrees of freedom associated with designing a cost and interference structure in order to enforce these principles. We introduce the notions of *segmentation* and *clustering* and propose their use in guiding the design process.

7.4.1 Segmentation and Clustering

Let each content piece be divided into M *segments*, where $M \leq N$ is a design parameter. Then, to each node $i \in \mathscr{V}$ assign a number $s_i \in \{1,\ldots,M\}$, meaning that node i caches a portion of each content piece corresponding to the s_ith segment. Next, to each node i, assign a *cluster* of $M-1$ nodes $C_i \subset \mathscr{V}\backslash\{i\}$ such that for all distinct k and j in $C_i \cup \{i\}$, $s_k \neq s_j$. In other words, each node in each cluster is assigned to a different segment of content.

Let $H_{i,j}$ be the number of hops separating nodes i and j along a shortest path between them over $\mathscr{G}_C$, and let $H_i = \max_{j\in C_i} H_{i,j}$. We identify the number of segments M and the assignment of clusters C_i, $\forall i \in \mathscr{V}$ as important degrees of freedom in the design process. Although in this chapter we do not address the question of how to optimally assign segments and clusters to each node, we suggest that for a given network topology $\mathscr{G}_C$, these should be chosen so as to minimize H_i, for each i. However, we note that for a fixed M, the problem of minimizing H_i via segmentation and cluster assignments for a given network topology can be framed as a graph coloring problem.

7.4.2 Designing the Costs $J_i(\cdot)$

With M and C_i specified, we consider the following general structure for the cost functions:

$$\begin{aligned} J_i(x) = \sum_{k\in\mathscr{F}} \Big[& E_{ca,i}\phi_{i,k}P_k + E_{tr,i}d_{i,k}(t)(1-\phi_{i,k})P_k \\ & + d_{i,k}(t)c_i\big((\sum_{j=1}^{M}\phi_{\pi_i(j),k})-1\big)^2 + b_{i,1}\sigma^2_{\pi_i(1),k} \\ & + \sum_{j=1}^{M-1} b_{i,j+1}(\sigma_{\pi_i(j),k}+\phi_{\pi_i(j),k}-\sigma_{\pi_i(j+1),k})^2\Big], \end{aligned} \tag{7.11}$$

where $\pi_i : \{1,\dots,M\} \to C_i \cup \{i\}$ is the permutation defined as $\pi_i : s_k \mapsto k \in \mathscr{V}$, and $b_{i,j},\ j \in \{1,\dots,M\}$ and c_i are positive real tuning parameters. Node i is to minimize its individual cost $J_i(\cdot)$ by attempting to influence the behavior of all other nodes that affect it. The rationale for the proposed cost design can be explained as follows. The first term in (7.11) penalizes node i for expending caching energy in proportion to the amount of data cached, while the second term is a penalty for having to transport that content which is not being cached. The second term encourages node i to cache larger portions of that content which is in high demand, in accordance with the first operational principle. On the other hand, when $d_{i,k} = 0$, the only term pertaining to content k in $J_i(x)$ is the caching penalty term, prompting node i to cache less of the content that is not in demand. The first two terms thereby capture the basic performance trade-off induced by the energy-proportional computing model. The remaining terms promote cooperation among nodes, as per the second and third operational principles.

The fourth and fifth terms in (7.11) are intended to encourage nodes to maintain caching boundaries between their respective content segments. We therefore refer to these as the *segmentation boundary* terms. Ideally, the contiguous block of content piece k that node $\pi_i(j+1) \in C_i \cup \{i\}$ caches should begin precisely where the block cached by node $\pi_i(j)$ ends – for all segments $j \in \{1,\dots,M-1\}$. When this is the case, the fifth term in (7.11) is identically zero. The fourth term encourages those nodes to whom the first segments are assigned to cache all content pieces starting with the first packet. Together, these terms have two functions. First, they penalize excessive copying of content within a cluster. Second, they penalize caching "gaps", thereby promoting the caching coverage of content within a cluster in a systematic way.

The maximization of content coverage is promoted further by the third term in (7.11), to which we refer as the *coverage* term. This term allows node i to encourage other nodes within the cluster C_i to maximize the collective caching coverage of each content piece k, in proportion to its demand. The fact that this term is proportional to $d_{i,k}(t)$ reflects the importance that node i attributes to this task; when content k is in high demand, it is very important to node i that nearby nodes cooperate in caching k in its entirety, if possible. This term is identically zero when each content piece is cached in entirety within the cluster C_i, thereby enforcing the third operational principle.

We note that this cost structure does not preclude either the extreme possibility that a given content piece k is not cached at all within the network, or that each node in the network caches a complete copy of content k (provided that $P_k \le B_i$, $\forall i \in \mathscr{V}$). If demands $d_{i,k}(t)$ fall to zero for all i, then we expect that eventually $(\sigma_{i,k}, \phi_{i,k}) \to (0,0)$ as well, due to the first and fifth terms in (7.11). On the other hand, when demand for content k is sufficiently high for all nodes in the graphical vicinity of node i, we expect that node i would cache some portions of the same content cached by his neighbors. Such would be the case if the $E_{tr,i}$ is so large that the second term in (7.11) dominates the effect of the others within the constraint sets X_i. Another type of possible qualitative behavior is that in a given cluster, the majority of a content piece is cached by a single node, regardless of its segment assignment. The qualitative characteristics of collective network behavior resulting from the proposed costs is explored further in Section 7.5.2.

To close this section, we wish to illustrate the concepts introduced so far by means of an example. Consider the network shown in Figure 7.3. Supposing that each content piece is segmented into $M = 4$ parts, one way to assign the segments for the given graph is as indicated next to the nodes. This segmentation allows us to define the cluster sets $C_1, \ldots, C_6$ in such a way that $H_i \leq 2$, for all i. This cluster assignment is not unique; if we specify $M = 4$ and $H_i \leq 2\ \forall i \in \mathscr{V}$ for this graph, then an alternative cluster assignment to the one shown involves $C_1 = \{2,5,6\}$ instead of $\{2,4,5\}$.

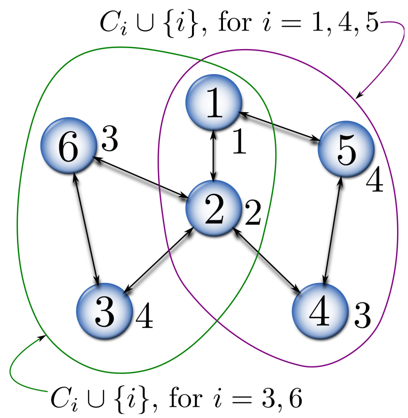

Figure 7.3: Segmentation and clustering example for a six-node network. As shown, $C_1 = \{2,4,5\}$ and therefore $\nabla_{x_j} J_1(x) \equiv 0$ for all $j \notin C_1 \cup \{1\}$, for $J_1(x)$ defined as in (7.11).

The best cluster assignment for node 2 is $C_2 = \{1,3,4\}$, with which $H_2 = 1$. The meaning of this maximal hop distance is that when $d_{2,k}(t)$ is high, under ideal circumstances node 2 should be able to assemble most of content piece k from within a 1-hop radius. In this case, the ideal circumstance for node 2 is characterized by a low value for its individual cost. Intuitively, this situation can also be interpreted as meaning that node 2 is not required to compromise with its neighbors at the corresponding network configuration and for the current set of network demands.

We remark that node i need not assemble any content piece $k \in \mathscr{F}$ exclusively from the nodes in the cluster C_i. For example, if node 3 happens to not be caching the portions of content piece k required by node 2, then 2 may receive these portions from node 5, or ultimately from a source node storing the original copy of this content piece (q.v. Figure 7.1). This model is consistent with the forwarding engine describing the operation of CCN [70], in which nodes acquire individual packets comprising the content from the nearest nodes that happen to cache these packets.

The significance of the set C_i (and hence the choice of cost functions (7.11)) is that it identifies a subset of nodes on $\mathscr{G}_C$ whose behavior node i attempts to influence in order to minimize its own cost. To relate the cluster sets to previously defined sets, we note that $C_i \cup \{i\}$ contains those nodes whose caching decisions affect node i's cost (c.f. the definition of the set $I(j)$ in (6.10)), and that $j \in C_i$ iff $i \in I(j)$. Moreover, $C_i \subseteq S(i)$, with $S(i)$ defined as in (6.11).

7.5 A Design Example

We intend to develop decentralized content caching strategies on the basis of the RCO algorithm developed in Chapter 6. Continuing with Example 6.4.1, we now show in detail how this may be done. For the sake of concreteness, let us consider the example network shown in Figure 6.3 (a) of Chapter 6, and a content catalogue $\mathscr{F} = \{1,2\}$ containing only two content pieces. Node i's caching decision at time t is then given by

$$x_i(t) = [\sigma_{i,1}(t), \phi_{i,1}(t), \sigma_{i,2}(t), \phi_{i,2}(t)]^T, \quad i \in \{1,2,3,4\}. \tag{7.12}$$

We are given the cache sizes B_i, transport efficiencies $E_{tr,i}$ and caching efficiencies $E_{ca,i}$ for each node $i \in \{1,2,3,4\}$, as well as the sizes P_1 and P_2 of the content pieces in $\mathscr{F}$. With nodes' individual costs assigned according to (7.11), the goal is to develop an update rule for each node, depending on locally available information only, such that the nodes' collective caching decisions eventually converge to the optimal network caching configuration x^* given by (7.3)–(7.7).

Recalling the method of RCO described in Chapter 6, we begin by specifying the sets $S(i)$ in (6.11) and the weights $a_{j,k}^{(i)}$ (6.13) that are needed for implementing (6.12). We choose $M = 3$ and assign the segments and clusters as indicated in Figure 7.4. With these segment and cluster assignments, the costs (7.11) ascribed to each

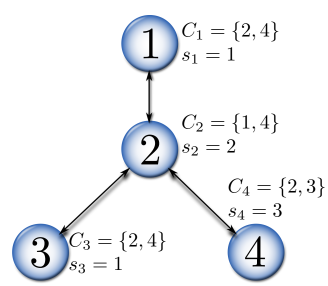

Figure 7.4: Segmentation assignment for the graph $\mathscr{G}_C$ in the design example.

node in $\mathscr{G}_C$ induce the interference structure shown in Figure 6.3 (b) – namely,

$$I(1) = \{1,2\}, \qquad I(2) = \{1,2,3,4\}$$
$$I(3) = \{3,4\}, \qquad I(4) = \{1,2,3,4\}.$$

The smallest subgraphs $\mathscr{G}_1$ and $\mathscr{G}_3$ containing the sets $I(1)$ and $I(3)$ respectively, are shown in Figure 6.3 (c) and (d), while $\mathscr{G}_2 = \mathscr{G}_4 = \mathscr{G}_C$. Recalling that the set $S(i)$ identifies those subgraphs to which node i belongs, we observe that

$$S(1) = \{1,2,4\}, \qquad S(2) = \{1,2,3,4\}$$
$$S(3) = \{2,3,4\}, \qquad S(4) = \{2,3,4\}.$$

In this example, $S(i) = C_i \cup \{i\}$ for all i except $i = 2$. Recalling that the set $S(i)$ identifies those components of x^* that agent i estimates through the RCO update rule (6.12), we note that node 2 must maintain an estimate of x_3^* – not because node 3 affects its cost, but because node 2 must act as an information conduit between nodes 3 and 4, who interfere with one another.

We let

$$\xi_{i,j} = [\sigma_{j,1}^{(i)}(t), \phi_{j,1}^{(i)}(t), \sigma_{j,2}^{(i)}(t), \phi_{j,2}^{(i)}(t)]^T \in \mathbb{R}^4 \tag{7.13}$$

represent node i's estimate at time t of $x_j^* \in \mathbb{R}^4$, node j's optimal caching decision. Since the NMAS coordination problem (7.3)–(7.7) involves constraints, we apply the following update rule for the variable $\xi_{i,j}$:

$$\xi_{i,j}(t+1) = \begin{cases} \mathbf{P}_{X_0}\left[\sum_{k=1}^N a_{j,k}^{(i)} \xi_{i,j}(t) - \alpha \nabla_{x_j} J_i(x)|_{x=\xi_i}\right], & j \neq i \\ \mathbf{P}_{X_i}\left[\sum_{k=1}^N a_{j,k}^{(i)} \xi_{i,j}(t) - \alpha \nabla_{x_j} J_i(x)|_{x=\xi_i}\right], & j = i \end{cases} \qquad \forall j \in S(i), \tag{7.14}$$

$$x_i(t) = \xi_{i,i}(t), \tag{7.15}$$

where the weights $a_{j,k}^{(i)}$ are as shown in Figure 6.4 in Chapter 6, and the sets X_0 and X_i are as in (7.9) and (7.10).

With this update rule, the implemented caching decisions (7.12) are guaranteed to remain in the feasible region defined by the constraints (7.4) to (7.7). Moreover, node i does not need to know other nodes' cache sizes, since it only enforces its own cache capacity constraint.

7.5.1 Implementing the Projection Operation

There are several numerical methods available for the computation of the projection operations in (7.14). Since the sets X_0 and X_i, $\forall i \in \mathscr{V}$ defined in (7.9) and (7.10) are polytopes in $\mathbb{R}^4$, each defined as the intersection of finitely many halfspaces, the projection operation may be implemented using the algorithm described in [97], for example. That algorithm is implemented as follows. Let X be a polytope in $\mathbb{R}^m$, described as the intersection of h halfspaces – i.e.,

$$X = \{x \in \mathbb{R}^m \,|\, Hx \preceq b\}, \tag{7.16}$$

where $H \in \mathbb{R}^{h \times m}$, $b \in \mathbb{R}^h$ and "$\preceq$" denotes a component-wise inequality.

We denote by $(X)_j$ the jth halfspace

$$(X)_j = \{x \in \mathbb{R}^m \,|\, [H]_j x \leq b_j\}, \qquad j \in \{1, \ldots, h\}, \tag{7.17}$$

so that $X = \bigcap_{j=1}^m (X)_j$. Then, given some point $z \in \mathbb{R}^m$, its projection onto X can be iteratively computed as follows:

Algorithm 7.5.1 (Algorithm 1, [97])**:**

1. Let $p_1(0), \ldots, p_h(0)$ be arbitrary points in $\mathbb{R}^m$.
2. Let $x(0) = z - \sum_{j=1}^h p_j(0)$.
3. Let $q_j(0) = x(0) + \lambda p_j(0)$.
4. For $k \in \mathbb{N}$, let
   (a) $p_j(k) = \frac{1}{\lambda}\left[q_j(k-1) - \mathbf{P}_{(X)_j}(q_j(k-1))\right]$,
   $\forall j \in \{1, \ldots, h\}$.
   (b) $x(k) = z - \sum_{j=1}^h p_j(k)$.
   (c) $q_j(k) = x(k) + \lambda p_j(k)$.

where $\lambda > 0$ is a tuning parameter. According to Theorem 1 in [97],

$$\lim_{k\to\infty} x(k) = \mathbf{P}_X(z), \tag{7.18}$$

whenever λ is chosen to be larger than $h/2$.

In step 4a of this algorithm, we are required to compute the projection of each point $q_j(k-1),\ j \in \{1,\dots,h\}$ onto the halfspace $(X)_j$; fortunately, $\mathbf{P}_{(X)_j}(\cdot)$ can easily be derived in closed form (q.v. §8.1.1 in [22], for example). The expression for the projection of some point $q \in \mathbb{R}^m$ onto the halfspace $(X)_j$ is given by

$$\mathbf{P}_{(X)_j}(q) = \begin{cases} q + \frac{(b_j - [H]_j q)}{\|[H]_j\|^2}[H]_j^T, & \text{if } [H]_j q > 0, \\ q, & \text{otherwise} \end{cases}, \tag{7.19}$$

where $[H]_j$ and b_j are as in (7.17).

In our case the polytope X_0 is described as

$$X_0 = \{x \in \mathbb{R}^4 \,|\, H_0 x \preceq b_0\}, \tag{7.20}$$

where

$$H_0 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \tag{7.21}$$

and

$$b_0 = [0,0,0,0,1,1]^T. \tag{7.22}$$

Similarly, the polytopes $X_i,\ i \in \{1,2,3,4\}$ can be described as

$$X_i = \{x \in \mathbb{R}^4 \,|\, Hx \preceq b_i\}, \tag{7.23}$$

where

$$H = \begin{bmatrix} H_0 \\ \hdashline [H]_7 \end{bmatrix}, \qquad [H]_7 = [0, P_1, 0, P_2], \tag{7.24}$$

and

$$b_i = \begin{bmatrix} b_0 \\ \hdashline B_i \end{bmatrix}. \tag{7.25}$$

7.5.2 Simulation Results

In this section, we explore the qualitative network behavior under the decentralized coordination scheme derived using reduced consensus optimization. We simulate algorithm (7.14) with the costs $J_i(x)$ defined as in (7.11), for the network shown in Figure 7.4, and its segments and clusters assigned as indicated in Figure 7.4. We chose the following parameters to obtain the simulation results shown in this section:

Problem Parameters:

- Router cache sizes: $B_1 = B_2 = B_3 = B_4 = 1000$ (i.e. router i has a cache size of $B_i q$ data units, where q is the number of data units comprising one data packet)

- Content catalog: $\mathscr{F} = \{1,2\}$, with $P_1 = 1500$ and $P_2 = 1500$
- Transport efficiencies: $E_{tr,1} = E_{tr,2} = E_{tr,3} = E_{tr,4} = 1.5$ Joules per q units of data
- Caching efficiencies: $E_{ca,1} = E_{ca,2} = E_{ca,3} = E_{ca,4} = 50$ Watts per q units of data

Cost Function Parameters:

- Segmentation boundary terms: $b_{i,j} = 200$, for all $i \in \{1,2,3,4\}$ and $j \in \{1,2,3\}$
- Coverage terms: $c_1 = c_2 = c_3 = c_4 = 50$

Simulation Parameters:

- Step size: $\alpha = 4.8 \times 10^{-6}$
- Initial conditions: $\xi_{i,j}(0) = [0,0,0,0]^T$, for all $i \in \{1,2,3,4\}$ and $j \in S(i)$

To implement Algorithm 7.5.1, we took $\lambda = 5$ and generated h random vectors in $p_j(0) \in [0,1]^4$, $j \in \{1,\ldots,h\}$, where $h = 6$ for X_0 defined in (7.20) and $h = 7$ for the sets X_1 to X_4 defined in (7.23).

To investigate the adaptive capability of the proposed decentralized caching strategy, we changed the content demand rates $d_{i,k}(t)$ at time $t = 75$, as indicated in Figure 7.5. Initially all the demands are identical for both

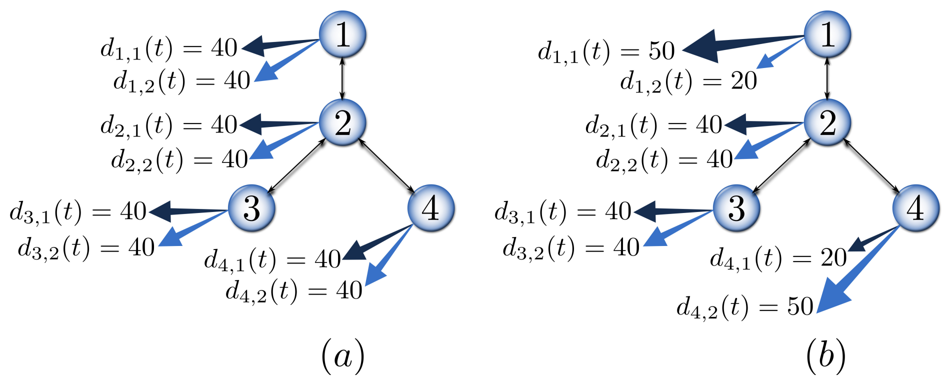

Figure 7.5: (a) Content demand rates for all $t \in \{1,\ldots,74\}$. (b) Content demand rates for all $t \in \{75,\ldots,150\}$.

content pieces and all nodes. At $t = 75$, node 1 experiences an increase in demand for content piece 1 and a decrease in demand for content piece 2, while node 4 experiences the exact opposite change in demand, and demands remain the same for other nodes.

The results are shown in Figures 7.6 and 7.7. Figure 7.6 shows the network caching configuration at $t = 75$ and $t = 150$, while Figure 7.7 shows the time evolution of the network configuration, and the evolution of nodes' estimates of others' optimal actions.

Since the transport and caching efficiencies and the cache sizes are identical for all nodes, and since the two content pieces are sized to fit within any three caches, we expect that after an initial transient period nodes 1, 2 and 4 should settle to a caching configuration in which they store approximately equal portions of each content piece, corresponding to their assigned segments (q.v. Figure 7.4). If the demands indicated in Figure 7.5 (a) are sufficiently large, we also expect that these three portions together should cover most of each content piece. Moreover, the caching pattern among these nodes should be identical for both content pieces. The same expectations apply to the nodes 2, 3 and 4, also due to the symmetry of the demand rates and the problem parameters. Figure 7.6 shows that the caching configuration at $t = 75$ – i.e., the iteration at which the demands change – is consistent with the desired operational principles, and conforms to our expectations.

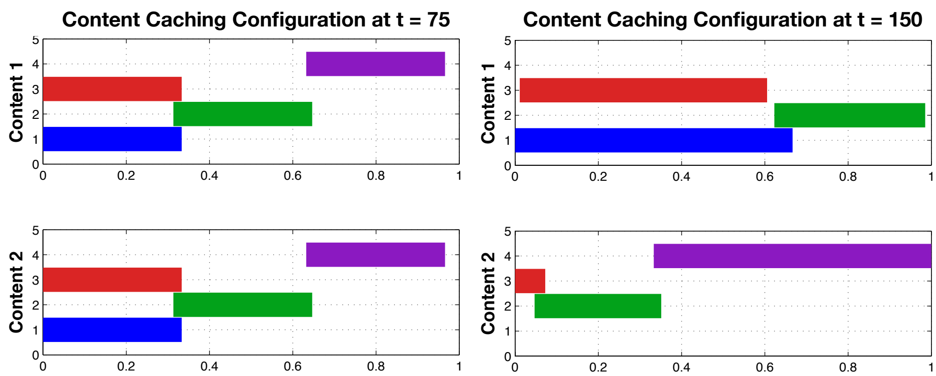

Figure 7.6: The content caching configurations at $t = 75$ and $t = 150$. The top and bottom plots indicate the starting location $\sigma_i^{(i)}(t)$ and the fraction $\phi_i^{(i)}(t)$ of the contiguous block of content that each node $i \in \{1,2,3,4\}$ caches, for content pieces 1 and 2, respectively. The nodes are labelled along the vertical axis.

When demands change at $t = 75$, we expect that node 1 should eventually cache less of content 2 and more of content 1, while node 4 should cache more of content 2 and less of content 1, in accordance with the first operational principle. The content caching configuration at $t = 150$ in Figure 7.6 shows that this is indeed the case.

We also note that the caching pattern for node 3 changes from that at $t = 74$, even though the demand that node 3 experiences remains the same throughout the simulation. This happens because $3 \in C_4$; although node 3 does not sense the change in content demand at node 4, its caching decisions are influenced by the "suggestions" that node 4 makes. In cooperating with node 4 (via node 2), node 3 changes its caching behavior so as to help cover those portions of content 1 that node 4 is no longer caching. This behavior is consistent with the third operational principle.

Figure 7.6 shows the set of actions implemented by the network nodes at two instants in time, while Figure 7.7 shows the complete time evolution of each variable updated in (7.14), including nodes' "suggestions" to one another. From Figure 7.7, we make two observations. First, since nodes' individual costs differ, the nodes are not initially in agreement as to what constitutes the optimal caching configuration. Eventually however, their estimates reach a consensus. This happens because the first term in the update law (7.14) is essentially a weighted average of node i's "opinion" and the "opinions" of his neighbors on $\mathscr{G}_j$, on what constitutes j's optimal action, for each $j \in S(i)$. The effect of repeated averaging can thus be intuited as a process by which nodes compromise with one another over individual objectives.

The second observation is that the constraints (7.4) to (7.7) are at no time violated throughout the evolution of the algorithm (7.14) (see Figure 7.8).

Remark 7.5.1 (The Merits of RCO)**:** In implementing the RCO-based update rule (7.14), the nodes collectively update a total of $2F\sum_{i\in\mathscr{V}}|S(i)|$ real values, and they exchange with their nearest neighbors a total of $2F\sum_{i\in\mathscr{V}}\sum_{j\in S(i)}|\mathscr{N}_j(i)|$ real values at each iteration. By contrast, in implementing an update rule based on al-

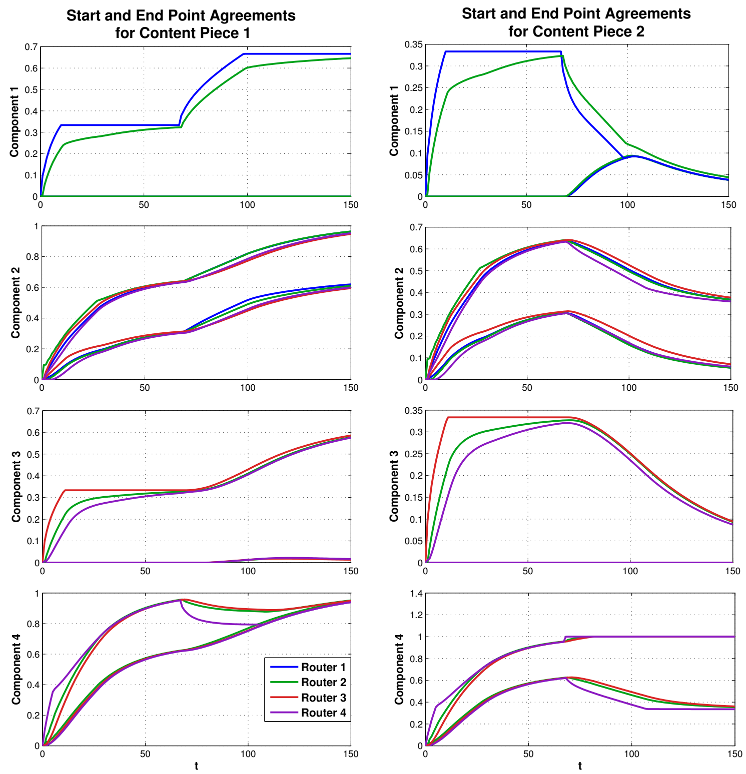

Figure 7.7: Time evolution of the content caching configuration. The plots in the first column pertain to content piece 1, while those in the second column pertain to content piece 2. The plots in row i ($i \in \{1,2,3,4\}$) correspond to agents' estimates of x_i^*. Each plot shows the emergence of consensus concerning where an agent's cached block ought to start (bottom set of lines), and where it ought to end (top set of lines).

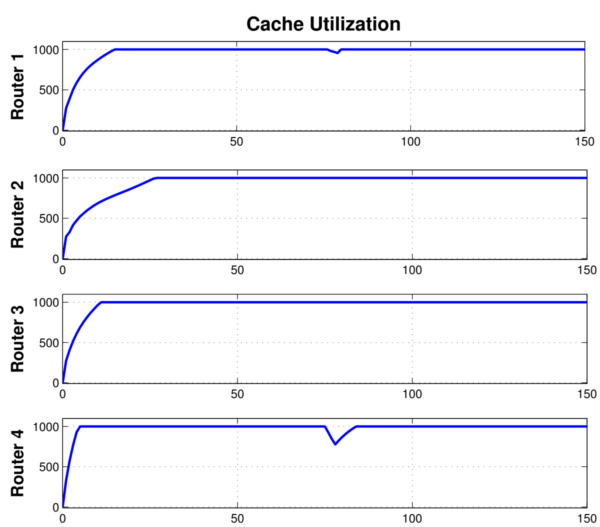

Figure 7.8: The cache capacity constraints are satisfied at each iteration.

gorithm (1.33), the nodes would need to update $2N^2F$ real values and exchange $2NF\sum_{i\in\mathscr{V}}|\mathscr{N}_C(i)|$ real values at each iteration, where $\mathscr{N}_C(i)$ denotes the set of node i's one-hop neighbors on the communication graph $\mathscr{G}_C$. In this example, the nodes collectively update a total of 52 real values, and exchange a total of 72 real values at each iteration. In implementing (1.33), they would update a total of 64 real values, and exchange a total of 96 real values at each iteration. In a large network with a sparse interference structure, implementing RCO instead of algorithm (1.33) can result in drastic reductions in communication and processing overhead associated with coordination among the nodes.

Another important benefit achieved by RCO is agents' nescience with regard to the operation of the collective. For example, with the costs defined as in (7.11), nodes 1 and 3 in this example need not be aware of each others' existence. In contrast, in implementing (1.33), each agent must know how many other agents are on the network because each agent maintains an estimate of the entire optimal network configuration. $\diamondsuit$

Remark 7.5.2 (The Scalability of RCO)**:** From (6.12), we observe that the number of real-valued variables updated by node i grows linearly in F and $|S(i)|$, where the set $S(i)$ identifies those subgraphs that node i belongs to. The cardinality of $S(i)$ depends on the way the clusters C_j are assigned to nodes j in the graphical vicinity of node i. Consequently, the maximum number of variables updated by any node within the network is independent of the network size N, and can be minimized by a judicious choice of cluster assignments. On the other hand, since nodes communicate only with their nearest neighbors, the communication overhead associated with coordination among nodes grows with node degree, rather than the size of the network.

The convergence rates of various consensus optimization schemes are studied in [114] and [116], for example. Convergence rates of these methods are known to be affected by the properties of the costs $J_i(\cdot)$, as well as the size of the Fiedler eigenvalue $\lambda_2(A)$ associated with the consensus matrix A (q.v. A3.3.1 and the remarks thereafter, in Chapter 3), whose value is affected by the network connectivity and the choice of link weights [167]. $\diamondsuit$

7.6 Performance Evaluation on the COST239

We evaluate the performance of an RCO-based content caching strategy (CCS) on a more realistic example. We compare its performance against that of the LFU cache policy, simulated on the eleven-node European optical backbone network COST239, which is shown in Figure 7.9. This network is also studied in [132], and the physical distances between the nodes (each of which represents a major European city) used here to compute various metrics of performance are taken from [132]. We have chosen to not make use of the two physically longest links in COST239 in our design. Our simulations are based on the parameters listed in §7.6.1. The demand rate signals $d_{i,k}(t)$ used in the simulations are intended to reflect a Zipf-like content request distribution; we describe our implementation of these signals in §7.6.3. Our implementation of the LFU caching policy is described in §7.6.4, and the performance metrics used to compare LFU against the proposed RCO-based CCS are described in §7.6.5. Finally, several simulation results are presented in §7.6.6.

7.6.1 Parameters

We chose the following set of parameters to obtain the simulation results presented in §7.6.6. Our choice of problem parameters and initial conditions is arbitrary. The cost function parameters were tuned in order to produce a visibly responsive algorithm behavior for the chosen set of problem parameters. The step-size is the last parameter tuned in order to obtain stable algorithm behavior.

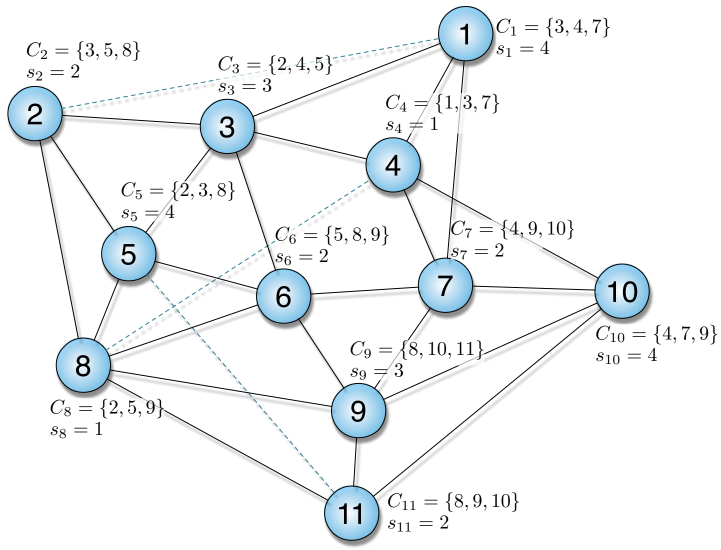

Figure 7.9: The European optical backbone network, COST239. Each content piece is assumed to be segmented into $M = 4$ parts. The segment and cluster assignments used to design the RCO-based CCSs are indicated next to each node. The dashed lines indicate links that are not used to coordinate the nodes.

Problem Parameters

We select the following problem parameters:

- Router cache sizes: $B_i = 100$ packets, for all $i \in \mathscr{V}$
- Content catalog: $\mathscr{F} = \{1, \ldots, 20\}$, with $P_k = 50$ packets, for all $k \in \mathscr{F}$.
- Transport efficiencies: $E_{tr,i} = 15$ Joules per q units of data, for all $i \in \mathscr{V}$ (each data packet is assumed to be q units of data in size)
- Caching efficiencies: $E_{ca,i} = 2$ Joules per q units of data, for all $i \in \mathscr{V}$.

We note that with $F = 20$ content pieces, the search space associated with the optimization problem (7.3) to (7.7) has a dimension of $2NF = 440$ real-valued variables.

Cost Function Parameters

To develop the RCO-based CCS for this example, we apply the design methodology proposed in §7.4. Specifically, we segment each content piece into $M = 4$ parts, and we assign the segments and clusters as shown in Figure 7.9. The tunable parameters $b_{i,j}$ and $c_{i,j}$ are selected as follows:

- Segmentation boundary terms: $b_{i,j} = 1300$, for all $i \in \mathscr{V}$ and $j \in \{1,2,3,4\}$.
- Coverage terms: $c_i = 500$, for all $i \in \mathscr{V}$.

Simulation Parameters

The RCO step-size and agent i's initial estimate of x_j^* are selected as follows:

- Step size: $\alpha = 5 \times 10^{-6}$

- Initial conditions: $\xi_{i,j}(0) = [0,\ldots,0]^T \in \mathbb{R}^{40}$, for all $i \in \mathscr{V}$ and $j \in S(i)$.

7.6.2 The RCO-Based CCS for COST239

With all relevant parameters selected as in §7.6.1, the RCO-based CCS takes the form (7.14), where the cost function $J_i(\cdot)$ is given by (7.11), the sets X_0 and X_i are given by (7.9) and (7.10) respectively, the link weights $a^{(i)}_{j,k}$ are assigned according to (6.13), and the caching decision implemented by node i at iteration t is given by (7.15), with x_i carrying the meaning in (7.2). The projection operations in (7.14) are implemented according to Algorithm 7.5.1.

We observe that the convergence of the said algorithm does not appear to be sensitive to the choice of initial conditions. Moreover, simulation experience suggests that finding a "sufficiently small" step size is generally not challenging. We tuned the step-size α once all other parameters were selected. A suitable α was found after only three simulation trials, even though the problem parameters and the cost function parameters were selected in a mostly arbitrary manner.

7.6.3 Zipf-like Demand Rate Signals

From the analysis of several Web proxy traces, it is found in [23] that requests arriving at a Web cache tend to be distributed according to Zipf's law, which states that the relative probability of a request for the rth most popular Web page is proportional to $\frac{1}{r}$. In particular, given a catalogue of F pages, with page k being the r_kth most popular page, [23] posits

$$P_F(k) = \frac{\left(\sum_{k=1}^{F} \frac{1}{k^{\rho}}\right)^{-1}}{r_k^{\rho}} \tag{7.26}$$

– with typical values of the exponent ρ ranging from 0.6 to 0.8 – as a model for the probability that the next request that arrives at a given cache is a request for page k. For exponents in this range, the Zipf-distribution implies that a vast majority of all requests are made only for only a small number of the most popular Web pages. Though [23] proposes this Zipf-like model in the context of Web caching, the model is also relevant to caching in the context of CDNs and CCNs [60].

To reflect the features of this model, we generate the demand rate signals according to

$$d_{i,k}(t) = m(t) \cdot \frac{\left(\sum_{k\in\mathscr{F}} \frac{1}{k^{0.75}}\right)^{-1}}{r_k(t)^{0.75}}, \tag{7.27}$$

where $r_k(t)$ is the popularity rank of content piece k at time t and $m(t)$ is the magnitude.

To see how the RCO-based CCS recovers from a disturbance, we let $r_k(t) = k$ and $m(t) = 150$ for the first 1000 iterations. For the next 1000 iterations, the popularities of the files are randomly re-assigned, but are taken to be the same across all nodes, while $m(t)$ is increased to 195.

7.6.4 LFU Caching

We compare the performance of the RCO-based CCS against that of the least-frequently-used (LFU) cache policy, which is known to outperform most local caching policies under the assumption that arriving requests are i.i.d. [23]. It is noted in [23] that LFU performs particularly well when the demands are Zipf-distributed.

Under LFU, nodes evict the least frequently accessed content in order to make room for new content. We approximate this behavior by a placement policy in which each node caches as much of the most popular content as possible at each t. Since we have chosen $B_i = 2P_k$, for all $i \in \mathscr{V}$ and $k \in \mathscr{F}$, node i implements LFU by caching contents k_1 and k_2, whose demand rate signals $d_{i,k_1}(t)$ and $d_{i,k_2}(t)$ are largest in magnitude at each time t. We allow each node implementing LFU to instantaneously acquire any content piece in response to $d_{i,k}(t)$.

7.6.5 Performance Metrics

We compare the performance of the LFU caching policy described in §7.6.4 against that of the RCO-based CCS described in §7.6.2, along several efficiency-related, network-wide performance metrics. In formulating these metrics, we make the assumption that each node is able to request and acquire individual packets comprising each content piece from the nearest node that caches them. We describe the adopted metrics in the sequel.

Network Transport Cost (NTC)

The Network Transport Cost (NTC) is intended to reflect the total actual energy cost associated with delivering uncached content. This metric sums the cost of transporting each uncached packet, in proportion to the rate at which requests arrive for the content piece to which the packet belongs. Specifically, we assume that if a content piece k is requested N_r times at node i, then each packet of k not cached by node i needs to be transported N_r times to node i, from the nearest node in the network that caches the required packet. The cost of transporting each packet is assumed to consist of two components: one that accounts of the energy required for intermediate nodes to route the packet, and another that accounts for the energy expended per unit distance that the packet travels along any network link. We define the NTC as follows:

$$\begin{aligned} NTC(t) = \cdot \sum_{i\in\mathscr{V}} \sum_{k\in\mathscr{F}} \sum_{p\in\overline{C}_{i,k}(t)} d_{i,k}(t) \cdot (E^{tr,H} H_{i,k,p}(t) \\ + E^{tr,D} D_{i,k,p}(t)), \end{aligned} \tag{7.28}$$

where $\overline{C}_{i,k}(t)$ is the set of all packets comprising content piece k that node i does not cache at time t, $E^{tr,H}$ is amount of energy that a node requires in order to route one packet, $H_{i,k,p}(t)$ is the number of hops that packet p of file k must traverse in order to reach node i from the closest node that caches p at time t, $E^{tr,D}$ is the amount of energy required to transport one packet over a distance of one kilometer, and $D_{i,k,p}(t)$ distance that packet p of file k must travel in order to reach node i from the closest node that caches p at time t. The link distances used in this metric are taken from [132]. If node i requires a packet p of content k which is not cached anywhere within the network at time t, then the maximum distance and hop-wise penalties of $D_{i,k,p}(t) = 2000$km and $H_{i,k,p}(t) = 10$ are incurred.

Average Hops Travelled by Un-cached Packets (AHT)

This metric measures the average number of hops that an uncached packet at node i needs to travel in order to arrive at i from the nearest node that caches it. This average is itself then averaged over all nodes in the network. We define the Average Hops Travelled (AHT) as follows:

$$AHT(t) = \frac{1}{N} \sum_{i\in\mathscr{V}} \frac{\sum_{k\in\mathscr{F}} d_{i,k}(t) \cdot \left(\sum_{p\in\overline{C}_{i,k}(t)} H_{i,k,p}\right)}{TPRS_i(t)}, \tag{7.29}$$

where $TPRS_i(t)$ is the total number of packet requests sent out by node i at time t – i.e.,

$$TPRS_i(t) = \sum_{i \in \mathscr{F}} d_{i,k}(t) \cdot |\overline{C}_{i,k}(t)|. \tag{7.30}$$

Network Caching Cost (NCC)

The Network Caching Cost (NCC) reflects the total amount of energy consumed by all the caches in the network. We assume that caching energy is proportional to the amount of content cached [10], and we therefore formulate the NCC metric as

$$NCC(t) = \sum_{i \in \mathscr{V}} \sum_{k \in \mathscr{F}} E_i^{ca,T} \cdot |C_{i,k}^0(t)|, \tag{7.31}$$

where $E_i^{ca,T}$ is the amount of energy required by node i to cache one packet for T seconds, T is the duration of one algorithm iteration, and $|C_{i,k}(t)|$ is the number of packets comprising content piece k that node i caches at time t.

Node-Averaged Cache Hit Ratios (ACHRs)

The *cache hit ratio* (CHR) is a standard metric in the literature on caching. In essence, this metric indicates the fraction of requests received that a cache is successfully able to serve. We adapt this notion to the present setting by considering cache hit ratios within an h-hop radius of a given node. To reflect the network-wide CHR performance, we average these h-hop radius CHRs across all nodes in the network.

We define the h-hop CHR at node i at time t (denoted as $CHR_i^h(t)$) as follows. Let $C_{i,k}^h(t)$ be the set of all packets pertaining to content piece $k \in \mathscr{F}$, cached within an h-hop radius of node i at iteration t. That is, $p \in C_{i,k}^h(t)$ implies that at least one node that is at most h hops away from node i is caching packet p. Also, let $TPRR_i(t)$ denote the total number of packet requests received by node i between iteration $t-1$ and t – i.e.,

$$TPRR_i(t) = \sum_{k \in \mathscr{F}} d_{i,k}(t) P_k. \tag{7.32}$$

Then, the h-hop CHR at node i is given by

$$CHR_i^h(t) = \frac{\sum_{k \in \mathscr{F}} d_{i,k}(t) |C_{i,k}^h(t)|}{TPRR_i(t)}, \quad h = 0, 1, 2, \ldots, \tag{7.33}$$

where $h = 0$ corresponds to the usual notion of CHR at node i, and $h = \infty$ corresponds to a network-wide CHR. Next, we define the Node-Averaged h-hop CHR (ACHR) as

$$ACHR^h(t) = \frac{1}{N} \sum_{i \in \mathscr{V}} CHR_i^h(t), \tag{7.34}$$

and the network-wide, node-averaged CHR as

$$ACHR^\infty(t) = \frac{1}{N} \sum_{i \in \mathscr{V}} \frac{\sum_{k \in \mathscr{F}} d_{i,k}(t) |C_{i,k}^\infty(t)|}{TPRR_i(t)}. \tag{7.35}$$

We assess the performance of the proposed RCO-based CCS described in §7.6.2 against that of the LFU caching policy described in §7.6.4 along the four CHR metrics $ACHR^0(t)$, $ACHR^1(t)$, $ACHR^2(t)$ and $ACHR^\infty(t)$.

7.6.6 Simulation Results

The RCO-based CCS and the LFU caching algorithms were run for 2000 iterations. The performance of the RCO-based CCS is compared to that of LFU along the seven metrics defined in §7.6.5, and the results are plotted in Figures 7.10 and 7.11.

For simplicity, we assume that all nodes have identical transport efficiency profiles, and we set the related energy efficiency parameters as $E_1^{ca,T} = \cdots = E_N^{ca,T} = E^{tr,D} = E^{tr,H} = 1$.

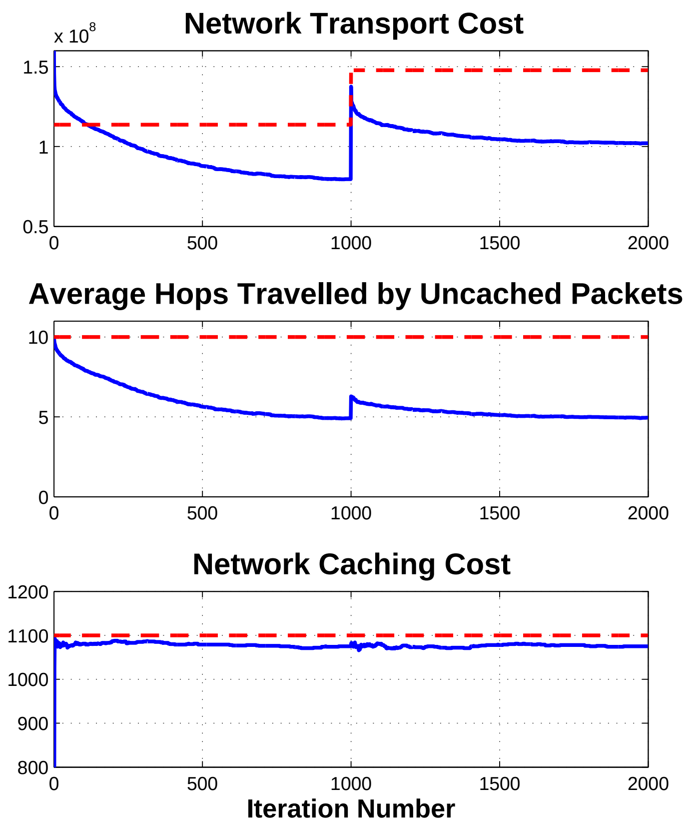

Figure 7.10: Performance of the RCO-based CCS (solid lines) compared to that of LFU (dashed lines) along the three metrics calculated using expressions (7.28), (7.29) and (7.31).

Since the RCO-based CCS is initialized with no content cached at any node, its NTC in Figure 7.10 is much higher at first. However, in coordinating their caching decisions, nodes eventually achieve a lower network transport cost as indicated by the NTC and AHT plots. The NCC plot indicates that for the valuation of problem parameters in §7.6.1, the RCO-based CCS maintains a full cache at each node for most of the time; this may not happen when the cost function parameters are set such that the $E_{tr,i}$ are lower relative to the $E_{ca,i}$.

Even though in both halves of the simulation the LFU caching policy results in each node caching exactly two of the most popular content pieces, the increase in network transport cost observed in the second half of the simulation occurs because the magnitude $m(t)$ of the demand rate signals is increased (q.v. §7.6.3).

From Figure 7.11 we notice that LFU outperforms the RCO-based CCS for the $ACHR^0(t)$ metric, through-

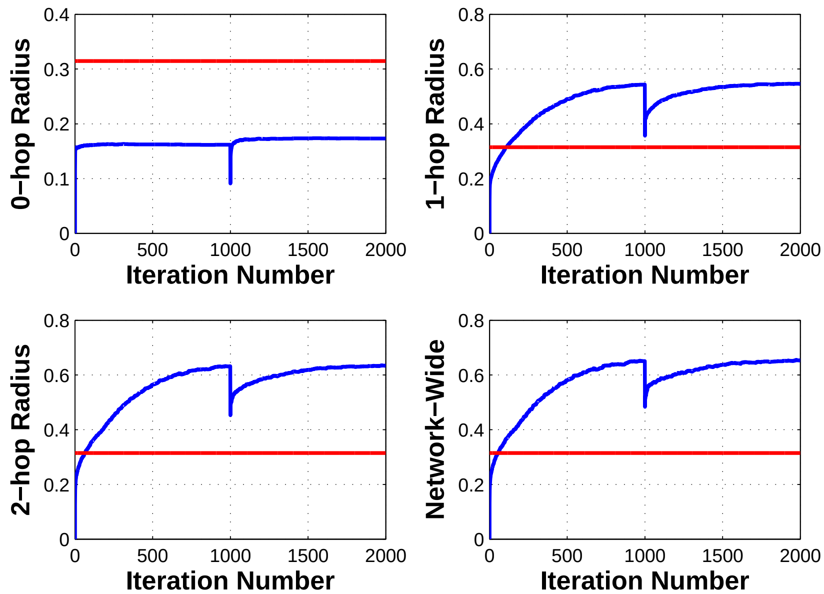

Figure 7.11: Performance of the RCO-based CCS (blue) compared to that of LFU (red) along the four Node-averaged CHR metrics $ACHR^0(t)$, $ACHR^1(t)$, $ACHR^2(t)$ and $ACHR^\infty(t)$ given by (7.34) and (7.35).

out the entire simulation. This is explained by the fact that LFU caching is especially effective when content demand exhibits a Zipf-like distribution [23]. In our implementation of LFU, each node caches exactly two of the locally most popular content pieces in the catalogue, thus ensuring its maximal individual CHR. By contrast, the RCO-based CCS entails what may be interpreted as the nodes' tendency to compromise maximizing their individual performance for the sake of improving network-wide performance. Indeed, the averaged CHR metrics $ACHR^1(t)$, $ACHR^2(t)$ and $ACHR^\infty(t)$ indicate that the network-wide CHR of the RCO-based CCS eventually becomes superior to our implementation of LFU.

It is also interesting to note that the h-hop radius CHRs for this example are not significantly improved for $h > 1$. This is a reflection of our particular choice of cluster and segment assignments indicated in Figure 7.9. In particular, the clusters are all comprised of 1-hop neighbors in this case. However in general, clusters may be assigned to include more than a subset of a node's 1-hop neighbors. The price potentially paid is that the number of variables that each node needs to update and exchange with its neighbors at each iteration may increase. The choice of clustering and segmentation as described in §7.4.1 represent an important degree of freedom in the design of RCO-based CCSs, and its effect on the efficacy of the resulting CCSs warrants further investigation.

The metrics plotted in Figures 7.10 and 7.11 suggest that RCO-based CCSs may potentially realize significant performance gains relative to caching policies that do not involve coordination among the cache-enabled nodes. Moreover, RCO-based CCSs can exhibit good adaptivity to changes in time-varying problem parameters such as content demand rates.

7.7 Final Remarks

In this chapter we developed a flexible methodology for the design of decentralized, adaptive content caching strategies. In response to real-time changes in content demand, a network node implementing the proposed strategy coordinates with its nearest neighbors and updates its caching decisions based on locally available information only. Collectively, the nodes achieve a network caching configuration that best meets a desired set of network-level performance criteria.

We focused on the problem of energy-efficient content delivery over networks with capabilities such as those of CCN. We provided a detailed design example illustrating the application of our methodology, which involves designing node costs in order to balance the trade-off between energy expended in caching content and that of transporting uncached content.

Our work opens many avenues for future investigation. First, the impact of segment and cluster assignments on network-wide performance metrics has not been investigated. There is a potentially interesting connection between the number of segments (i.e. the number of nodes in a cluster) and the hop-wise distance between content replicas. Previous work which attempts to quantify optimal hop-wise distances between content replicas (q.v. [60], for example) may help inform best practices in making segment and cluster assignments. Second, better designs for the cost functions are certainly possible. For example, one might consider designing these functions in a way that avoids redundant caching among disjoint clusters. Also, though caching and transport are the primary energy-consuming functions within a network they may not the only ones; one may consider encoding a more elaborate set of operational principles, which take into account the energy consumed by other network elements. Fourth, although we focused on energy efficiency, other performance objectives can just as easily be incorporated. Fifth, it would be interesting to investigate the algorithms's ability to cope with topological network changes. Specifically, in the present work, cluster and segment assignments are envisioned as initially being assigned offline. Once a network is up and running the proposed CCS, newly added nodes can be assigned segments in a way that maximizes the segment diversity in their graphical neighbourhood, and cluster mates in a way that minimizes the node's hop-wise cluster radius.

Chapter 8

Concluding Remarks

The accelerated advance of information and communication technologies is giving rise to disruptive technological trends characterized by the ubiquitous presence of geographically distributed sensors and actuators. Large-scale systems such as power grids, communication networks and logistics operations are increasingly integrating intelligent components that are capable of autonomously processing and influencing their local environment, as well as interacting with other such components. It is widely believed that the proliferation of these capabilities will lead to greater interdependence among systems spanning previously disparate application domains, thereby enabling the realization of unprecedented efficiency gains in their joint operation. Harnessing the full potential of these technological trends is contingent in part on our ability design individual component behaviors in such a way that the ongoing interactions among multitudes of these components lead to their predictable, reliable and optimal collective behavior.

In this thesis we have considered the general synthesis problem of decentralized coordination control for multiagent systems in which the agents are either static, or dynamic entities. We have formulated several classes of decentralized coordination control problems (DCCPs), and we have proposed a number of variants of a decentralized optimization method known as *consensus optimization* (CO) as a foundation for solving them. The individual contributions of this thesis are unified by an overall effort to assess the utility and efficacy of this approach, and to enable its rigorous analysis.

Our contributions can be summarized as follows. We proposed a novel framework for the convergence analysis of a class of CO algorithms (q.v. Chapter 3). The perspective taken leads to the relaxation of several standard assumptions on CO algorithms. This framework easily accommodates the study of continuous-time variants of CO (q.v. Chapter 4), but more importantly, it facilitates the study of DCCP scenarios in which the agents executing the CO algorithm are themselves dynamic entities, whose dynamics interact with those of the CO algorithm and thereby affect its performance (q.v. Chapter 5). We have also proposed a streamlined version of the standard form of CO, which we have called *reduced consensus optimization* (RCO) (q.v. Chapter 6). This variant is especially suitable in the context of decentralized coordination control, and under certain conditions may involve significantly less coordination overhead relative to CO. We considered the application problem of designing decentralized content caching strategies for information centric networks, which we have cast as a static DCCP, and addressed by means of RCO (q.v. Chapter 7).

Several new directions are made possible by the work in this thesis. An immediate direction in which the results of Chapter 3 ought to be extended would aim at relaxing the assumptions on the agents' communication graph (q.v. Remark 3.3.3). Even as early as in the work of Tsitsiklis [154] it was known that the convergence of discrete-time consensus algorithms does not require the communication graph to be connected at each iteration (c.f. our discussion in §2.2 and §2.3.1). In fact, the communication graph need only be "connected in the long run", which means, quite remarkably, that it need not be connected at *any* iteration of the algorithm. Moreover, the estimate exchanges among agents need not be synchronous. Apart from relaxing the connectivity and synchrony assumptions implicitly made in Chapter 3, it would be equally important to relax the requirement that agents exchange information in a bidirectional manner. Within the last decade there has been a significant effort to accommodate agent communication over directed graphs, and much has been learned. It is left to future work to investigate how some of these results can be incorporated into the proposed analytic framework.

In the literature on distributed optimization, one finds a great deal of emphasis on developing methods with fast convergence rates. In particular, attaining convergence rates on the order of those achievable by second-order (i.e. Newton's) methods in the distributed context is currently of great interest. In showing how to apply our analytic techniques to consensus-decentralized weighted gradient methods, we made a conjecture that the same techniques may well accommodate the case in which the gradient matrix weights $H_i(t)$ differ for each agent i, and are such that $(H_i(t))_{t=0}^{\infty} \to (H(t))_{t=0}^{\infty}$, where for each t, $H(t)$ is symmetric and satisfies the bounds $m \leq \|H(t)\| \leq M$, for some positive, real numbers m and M (q.v. Remark 3.8.2). In such CO schemes agents would update not only their primal optimization variables, but also maintain an estimate of some approximation of the inverse Hessian associated with the collective cost function. Our techniques would then effectively lead to a set of convergence conditions for a class of decentralized quasi-Newton methods.

The work presented in Chapter 5 can be extended in at least three obvious ways. Of immediate interest are the conditions that would guarantee the practically asymptotically stable behavior of a decision updating process involving search directions $\hat{s}_i(\cdot)$ formed not from the transient-corrupted measurements of the gradients of agents' cost functions, but rather from the measurements of their state-dependent cost functions themselves (q.v. Example 1.4.3). A second potential extension is to consider decision updating rules that enforce actuator constraints. A third potential extension is to incorporate the ideas in Chapter 6, in order to develop RCO-based decentralized extremum-seeking schemes.

The work presented in Chapter 7 has left us with at least one unanswered theoretical question. How does one characterize the class of exogenous environmental conditions $d_i(t)$ (which, in this case represent the demand rate signals measured by node i in the network) to which an RCO-based coordination control strategy is able to effectively respond? Intuitively, it is clear that for "sufficiently slowly varying" signals $d_i(t)$ that affect the agents' cost functions, an optimization algorithm like RCO should be able to "keep up". However, it is not clear what "sufficiently slowly varying" actually means. One potential way to address this question is to attempt to quantify temporal changes in the cost functions by somehow incorporating time-derivative data (if it is available) into the decision updating process $\mathscr{D}_i$ – either directly, or by means of setting appropriate conditions on the algorithm's tunable parameters. On the other hand, it seems that this question is not unlike those that led to the discovery of the so-called *internal model principle* for linear systems [51]. In particular, another potentially interesting approach to addressing this question is to consider $d(t)$ that are generated by some hypothetical dynamical exosystem whose dynamics can be modelled. Then, one might consider designs of decision updating processes that directly incorporate or account for the model of this exosystem.

Throughout our doctoral research adventures, we have, from the start, been primarily motivated by the prospect of engineering simplistic individual behaviors and interaction rules that lead to purposive, sophisticated collective behavior in the absence of centralized coordination mechanisms. Especially intriguing is the possibility of designing such rules in a way that maximally leverages the synergism of the collective, endowing it with functionalities that extend far beyond those of the modules from which it is comprised.

In studying consensus optimization in the context of DCCPs, we have developed the general impression that decentralized optimization methods represent an excellent starting point for the design of coordination control strategies, because they are both general and practical. They are general because a wide variety of coordination objectives can be easily expressed in the language of convex programming. They are practical because from practitioner's point of view, this language is easy to understand, and it is generally easy to implement.

Nature is a master designer of emergence phenomena in complex systems. The variety and the apparent efficacy of naturally evolved multiagent collectives seem to taunt the engineer with the obvious challenge. Although we are convinced that CO is a viable foundation for achieving modestly sophisticated collective behaviors through the design of local interaction rules, it is not clear to what extent this design approach is able to maximize the "distance" between the functional capabilities of the individual, and those of the collective.

Another way of framing this question is to consider CO as a generic model for the seemingly purposive, coordinated behavior of many naturally occurring multiagent collectives. Are animal collectives, such as bird flocks for example, really only optimizing some specific collective objective function? How many varieties of complex behaviors can be expressed within the language of optimization? Is it always just a matter of finding the right function, or are there tangible limits to the predictive powers of behavioral models based on CO? Is the general structure of the consensus optimization dynamics rich enough to produce behavior exhibiting phase transitions and other critical phenomena, and if so, how many different flavors of these phenomena are possible for CO?

Our understanding of these important topics is very far from complete, and this thesis is already far too long. It is therefore with some reluctance, that this author leaves off all further philosophical speculations to Tortoise.

Appendix A

Notation and Basic Definitions

The following notation and definitions are used throughout this thesis:

- The Euclidean vector norm is denoted by $\|\cdot\|$, while $\|\cdot\|_1$ denotes the vector 1-norm.
- The entry in the ith row and jth column of a matrix A is denoted $[A]_{i,j}$.
- If $A \in \mathbb{R}^{n\times n}$, then $\sigma(A)$ is its set of eigenvalues.
- A matrix $A \in \mathbb{R}^{N\times N}$ is *stochastic* if each of its entries is non-negative, and $\forall i \in \{1,\ldots,N\}$, $\sum_{j=1}^{N}[A]_{i,j} = 1$. A matrix A is *doubly stochastic* if both A and A^T are stochastic.
- An n-dimensional vector whose entries are all equal to 1 is denoted by $\mathbf{1}_n$. Similarly, $\mathbf{0}_n$ denotes an n-dimensional vector whose entries are all equal to zero.
- An n by n identity matrix is denoted by I_n.
- The Kronecker product of matrices A and B is denoted by $A \otimes B$.
- The set of non-negative real numbers is denoted by $\mathbb{R}_+$, while the set of positive real numbers is denoted by $\mathbb{R}_{++}$.
- For a point $x_o \in \mathbb{R}^n$, and $r \in \mathbb{R}_{++}$, $\bar{B}_r^n(x_o) = \{x \in \mathbb{R}^n : \|x - x_o\| \leq r\}$ and $B_r^n(x_o) = \{x \in \mathbb{R}^n : \|x - x_o\| < r\}$. For a compact set $S \subset \mathbb{R}^n$, $\bar{B}_r^n(S) = \{x \in \mathbb{R}^n \mid \|x - \mathbf{P}_S(x)\| \leq r\}$, while $B_r^n(S) = \{x \in \mathbb{R}^n \mid \|x - \mathbf{P}_S(x)\| < r\}$. In both preceding cases, the dimension "n" will be dropped when there is no ambiguity.
- For $S \subset \mathbb{R}^n$, $\mathrm{co}(S)$ is its convex hull.
- For some set S, $|S|$ denotes its cardinality.
- The set of continuous functions from $\mathbb{R}^n$ into $\mathbb{R}^m$ is denoted by $C^0[\mathbb{R}^n, \mathbb{R}^m]$.
- We say that a function $V \in C[\mathbb{R}^n, \mathbb{R}_+]$ is *positive definite* with respect to a closed set S on a set $\Omega \supset S$ if, $V(S) = \{0\}$, and $V(x) > 0$ for all $x \in \Omega \backslash S$.
- A function $V : x \mapsto V(x)$ on $\mathbb{R}^n$ is *radially unbounded* with respect to a closed set $S \subset \mathbb{R}^n$, if for any $B \in \mathbb{R}$, there exists an $r \in \mathbb{R}_{++}$ such that $V(x) > B$, for all $x \in \mathbb{R}^n \backslash \bar{B}_r(S)$.

- A function $\gamma :\in C^0[\mathbb{R}_+,\mathbb{R}_+]$ is a *class-$\mathscr{K}$* function if it is strictly monotonically increasing, and $\gamma(0) = 0$. If $\gamma(\cdot) \in \mathscr{K}$ is radially unbounded, then it also belongs to the $\mathscr{K}_\infty$ class of functions (q.v. Definition 4.2, [80]).

- A function $\sigma \in C^0[\mathbb{R}_+,\mathbb{R}_+]$ is a *class-$\mathscr{L}$* function if it is strictly monotonically decreasing, and $\lim_{t\to\infty} \sigma(t) = 0$ (q.v. Definition 2.6, [62]).

- A function $\beta \in C^0[\mathbb{R}_+ \times \mathbb{R}_+,\mathbb{R}_+]$ is a *class-$\mathscr{KL}$* function if $\forall t \in \mathbb{R}_+$, $s \mapsto \beta(s,t) \in \mathscr{K}$, while $\forall s \in \mathbb{R}_+$, $t \mapsto \beta(s,t)$ is monotonically decreasing (but not necessarily strictly monotonically decreasing; q.v. Definition 24.2, [62]).

- A *sequence* is a function in $(\mathbb{R}^n)^{\mathbb{N}}$, denoted by $(x_k)_{k=1}^\infty$ or $(x_k)_{k\in\mathbb{N}}$.

- For some set $S = \{k_1,\ldots,k_{|S|}\} \subset \mathbb{N}$ with $k_i < k_j$ whenever $i < j$, $(x_k)_{k\in S}$ denotes the ordered $|S|$-tuple of numbers $(x_{k_1},\ldots,x_{k_{|S|}})$.

- If $(t_k)_{k=1}^\infty$ is an increasing sequence and $q : \mathbb{R}_+ \to \mathbb{R}^n$, then $q(t_k)$ is often denoted as q, while $q(t_{k+1})$ is dented as q^+. Moreover, $\Delta q := q^+ - q$.

- For some quantity $q : \mathbb{R}_+ \to \mathbb{R}^n$, $q(t_k^-) = \lim_{t\uparrow t_k} q(t)$.

- The gradient of a differentiable function $J : \mathbb{R}^n \to \mathbb{R}$ is denoted by $\nabla J(\cdot)$.

- The *subdifferential* of a convex function $J : \mathbb{R}^n \to \mathbb{R}$ at a point $x \in \mathbb{R}^n$ is defined as the set of all vectors $g \in \mathbb{R}^n$ satisfying the inequality

$$J(z) \geq J(x) + g^T(z-x), \quad \forall z \in \mathbb{R}^n, \tag{A.1}$$

  and it is denoted by $\partial J(x)$ (q.v. §4.2, [14]).

- If $S \subset \mathbb{R}^n$ is closed and $x_o \in \mathbb{R}^n$, then

$$\mathbf{P}_S(x_o) = \arg\min_{x\in S} \|x_o - x\| \tag{A.2}$$

  is the Euclidean projection of x_o onto S.

Bibliography

[1] Annual report. Technical report, Pacific Northwest Smart Grid Demonstration Project, 2013.

[2] *IEEE Control Systems Magazine*, volume 34, August 2014.

[3] R.P. Agarwal. *Difference equations and inequalities - theory, methods and applications*. Marcel Dekker, 2000.

[4] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman. A survey of information-centric networking. *IEEE Communications Magazine*, 50:26–36, 2012.

[5] P.W. Anderson. More is different - broken symmetry and the nature of the hierarchical structure of science. *Science*, 177:393–397, 1972.

[6] M. Aoki. Control of large-scale dynamic systems by aggregation. *IEEE Transactions on Automatic Control*, AC-13:246–253, 1968.

[7] K.B. Ariyur and M. Krstić. *Real-time optimization by extremum-seeking control*. Wiley, 2003.

[8] H. Arrow, L. Hurwicz, and H. Uzawa. *Studies in linear and non-linear programming*. Stanford University Press, 1958.

[9] H. Bai, M. Arcak, and J. Wen. *Cooperative control design: a systematic, passivity-based approach*. Springer, 2011.

[10] L.A. Barroso and U. Holzle. The case for engergy-proportional computing. *IEEE Computer*, 40:33–37, 2007.

[11] T. Basar and G.J. Olsder. *Dynamic Noncooperative Game Theory*. SIAM, 1995.

[12] A. Bemporad, M. Heemels, and M. Johansson, editors. *Networked control systems*. Springer, 2010.

[13] A. Berman and R.J. Plemmons. *Nonnegative matrices in the mathematical sciences*. SIAM, 1979.

[14] D. Bertsekas, A. Nedić, and A. Ozdaglar. *Convex analysis and optimization*. Athena Scientific, 2003.

[15] D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[16] P. Bianchi and J. Jakubowicz. Convergence of multi-agent projected stochastic gradient algorithm for non-convex optimization. *IEEE Transactions on Automatic Control*, 58:391–405, 2013.

[17] E. Bonabeau, G. Theraulaz, J.-L. Deneubourg, N.R. Franks, O. Rafelsberger, J.-L. Joly, and S. Blanco. A model for the emergence of pillars, walls and royal chambers in termite nests. *Philosophical Transactions of the Royal Society of London*, 353:1561–1576, 1998.

[18] J.T. Bonner. *The social amoebae: the biology of cellular slime molds*. Princeton University Press, 2009.

[19] V.S. Borkar. *Stochastic approximation: a dynamical systems viewpoint*. Cambridge University Press, 2008.

[20] S. Borst, V. Gupta, and A. Walid. Distributed caching algorithms for content distribution networks. In *Proceedings of the 29th IEEE International Conference on Computer Communications (INFOCOM)*, 2010.

[21] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3:1–122, 2010.

[22] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[23] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web caching and zipf-like distributions: evidence and implications. *Proceedings of the IEEE Infocom*, pages 126–134, 1999.

[24] M.E. Broucke. Reach control on simplices by continous state feedback. *Proceedings of the American Control Conference*, pages 1154–1159, 2009.

[25] F. Bullo, J. Cortes, and S. Martinez. *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*. Princeton Series in Applied Mathematics, 2006.

[26] T. Campbell, C. Williams, O. Ivanovna, and B. Garrett. Could 3d printing change the world? technologies, potential, and implications of additive manufacturing. Technical report, Atlantic Council Strategic Foresight Report, 2011.

[27] R.L.G. Cavalcante and S. Stanczak. A distributed subgradient method for dynamic convex optimization problems under noisy information exchange. *IEEE Journal of Selected Topics in Signal Processing*, 7:243–256, 2013.

[28] H. Che, Y. Tung, and Z. Wang. Hierarchical web caching systems: modeling, design and experimental results. *IEEE Journal on Selected Areas in Communications*, 20:1305–1314, 2002.

[29] A.I. Chen and A. Ozdaglar. A fast distributed proximal-gradient method. *Proceedings of the Fiftieth Annual Allerton Conference, Monticello, IL*, pages 601–609, 2012.

[30] J. Chen and A.H. Sayed. Diffusion adaptation strategies for distributed optimization and learning over networks. *IEEE Transactions on Signal Processing*, 60:4289–4306, 2012.

[31] M. Chiang, S.H. Low, A.R. Calderbank, and J.C. Doyle. Layering as optimization decomposition: a mathematical theory of network architectures. *Proceedings of the IEEE*, 95:255–312, 2007.

[32] K. Cho, M. Lee, K. Park, T.T. Kwon, Y. Choi, and S. Pack. Wave: popularity-based and collaborative in-network caching for content-oriented networks. In *Proceedings of the 31st IEEE International Conference on Computer Communications (INFOCOM)*, 2011.

[33] N. Choi, K. Guan, D. Kilper, and G. Atkinson. In-network caching effect on optimal energy consumption in content-centric networking. In *Proceedings of the IEEE ICC'12 Next Generation Network Symposium*, 2012.

[34] P.D. Christofides, J. Liu, and D. Munoz de la Pena. *Networked and distributed predictive control*. Springer, 2011.

[35] D. Cohen, M. Sargeant, and K. Somers. 3-d printing takes shape. Technical report, McKinsey Quarterly, 2014.

[36] I. Couzin. Collective minds. *Nature*, 445:715, 2007.

[37] I. Couzin. Collective cognition in animal groups. *Trends in Cognitive Sciences*, 13:36–44, 2008.

[38] R. Cuevas, N. Laoutaris, X. Yang, G. Siganos, and P. Rodriguez. Deep diving into bittorrent locality. In *Proceedings of the IEEE INFOCOM, Shanghai*, 2011.

[39] G.B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8:101–111, 1960.

[40] E.J. Davison. A method for simplifying linear dynamic systems. *IEEE Transactions on Automatic Control*, AC-11:93–102, 1966.

[41] E.J. Davison. The robust decentralized control of a general servomechanism problem. *IEEE Transactions on Automatic Control*, AC-21:14–24, 1976.

[42] L Dong, D. Zhang, Y. Zhang, and D. Raychaudhuri. Optimal caching with content broadcast in cache-and-forward networks. In *Proceedings of the IEEE International Conference on Communications (ICC)*, 2011.

[43] P.M. Dower, P.M. Farrell, and D. Nešić. Extremum seeking control of cascaded raman optical amplifiers. *IEEE Transactions on Control Systems Technology*, 16:396–407, 2008.

[44] J.C. Duchi, A.Agarwal, and M.J. Wainwright. Dual averaging for distributed optimization: convergence analysis and network scaling. *IEEE Transactions on Automatic Control*, 57:592–606, 2012.

[45] H.-B. Dürr, M.S. Stanković, and K.H. Johansson. Distributed positioning of autonomous mobile sensors with application to coverage control. *Proceedings of the 2011 American Control Conference, San Francisco, USA*, pages 4822–4827, 2011.

[46] A. Dussutour, T. Latty, M. Beekman, and S.J. Simpson. Amoeboid organism solves complex nutritional challenges. *Proceedings of the National Academy of Sciences*, 107:4607–4611, 2010.

[47] X. Fan, T. Alpcan, M. Arcak, T.J. Wen, and T. Başar. A passivity based approach to game-theoretic cdma power control. *Automatica*, 42:1837–1847, 2006.

[48] D. Feijer and F. Paganini. Krasovskii's method in the stability of network control. In *Proceedings of the American Control Conference, St. Louis, MO, USA*, 2009.

[49] M. Fiedler. Algebraic connectivity of graphs. *Chzechoslovak Mathematical Journal*, 23:298–305, 1973.

[50] W. Findeisen. Decentralized and hierarchical control under consistency or disagreement of interests. *Automatica*, 18:647–664, 1982.

[51] B.A. Francis and W.M. Wonham. The internal model principle of control theory. *Automatica*, 12:457–465, 1976.

[52] A. Garfinkel. The slime mold dictyostelium as a model of self-organization in social systems. In *Self-organizing systems: the emergence of order*. Plenum Press, 1987.

[53] B. Gharesifard and J. Cortes. Continuous-time distributed convex optimization on weight-balanced digraphs. *Proceedings of the* 51^{st} *IEEE Conference on Decision and Control*, pages 7451–7456, 2012.

[54] B. Gharesifard and J. Cortes. Distributed continuous-time convex optimization on weight-balanced digraphs. *IEEE Transactions on Automatic Control*, page To appear, 2013.

[55] N. Ghods, P. Frihauf, and M. Krstić. Multi-agent deployment in the plane using stochastic extremum seeking. *49th IEEE Conference on Decision and Control, Atlanta, USA*, pages 5505–5510, 2010.

[56] N. Ghods and M. Krstić. Multi-agent deployment around a source in one dimension by extremum seeking. *Proceedings of the 2010 American Control Conference, Baltimore, USA*, pages 4794–4799, 2010.

[57] J. Giles. Internet encyclopedias go head to head: Jimmy Wales' Wikipedia comes colose to Britannica in terms of the accuracy of its science entries, a *Nature* investigation finds. *Nature*, 438, 2005.

[58] D.M. Gordon. *Ant encounters: interaction networks and colony behavior*. Princeton University Press, 2010.

[59] D. Grundel, R. Murphey, P. Pardalos, and O. Prokopyev, editors. *Cooperative systems: control and optimization*. Springer, 2007.

[60] K. Guan, G. Atkinson, D.C. Kilper, and E. Gulsen. On the energy efficiency of content delivery architectures. In *Proceedings of the 2011 IEEE International Conference on Communications (ICC)*, 2011.

[61] J.M. Guerrero, J.C. Vasquez, J. Matas, L. Garcia de Vicuna, and M. Castilla. Hierarchical control of droop-controlled AC and DC microgrids – a general approach toward standardization. *IEEE Transactions on Industrial Electronics*, 58:158–172, 2011.

[62] W. Hahn. *Stability of motion*. Springer, 1967.

[63] H. Haken. *Synergetics: an introduction. Nonequilibrium phase ttransition and self-organization in physics, chemistry and biology*. Springer-Verlag, 1977.

[64] P. Hande, S. Rangan, M. Chiang, and X. Wu. Distributed uplink power control for optimal SIR assignment in cellular data networks. *IEEE Transactions on Networking*, 16:1420–1433, 2008.

[65] W.V. Harris. Termite mound building. *Insectes Sociaux*, 3:261–268, 1956.

[66] D.R. Hofstadter. *Gödel, Escher, Bach: an Eternal Golden Braid*. Basic Books, 1999.

[67] R.A. Horn and C.R. Johnson. *Matrix analysis*. Cambridge University Press, 1985.

[68] CISCO Systems Inc. The zettabyte era (cisco vni report). Technical report, available on-line: `http://www.cisco.com/web/solutions/sp/vni/vni_forecast_highlights/index.html`, 2012.

[69] A. Isidori. *Nonlinear control systems*. Springer, 1995.

[70] V. Jacobson, D.K Smetters, J.D. Thornton, M.F. Plass, N.H. Briggs, and R.L. Braynard. Networking named content. In *Proceedings of the 5th ACM International Conference on Emerging Networking Experiments and Technologies, Rome, Italy*, 2009.

[71] A. Jadbabaie, J. Lin, and S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48:988–1001, 2003.

[72] D. Jakovetić, J.M.F. Xavier, and J.M.F. Moura. Convergence rates of distributed nesterov-like gradient methods on random networks. *IEEE Transactions on Signal Processing*, 62:868–882, 2014.

[73] A. Jiang and J. Bruck. Optimal content placement for en-route web caching. In *Proceedings of the Second IEEE International Symposium on Network Computing and Applications*, 2003.

[74] S. Jin and A. Bestavaros. popularity-aware greedydual-size web proxy caching algorithms. *Proceedings of ICDCS*, pages 254–261, 2000.

[75] B. Johansson, T. Keviczky, M. Johansson, and K.H. Johansson. Subgradient methods and consensus algorithms for solving convex optimization problems. *Proceedings of the* 47^{th}*IEEE Conference on Decision and Control, Cancun, Mexico*, pages 4185–4191, 2008.

[76] N.R. Sandell Jr., P. Varaiya, M. Athans, and M.G. Safonov. Survey of decentralized control methods for large scale systems. *IEEE Transactions on Automatic Control*, AC-23:108–129, 1978.

[77] R. Kalman. Discovery and invention: the Newtonian revolution in systems technology. *Journal of Guidance, Control, and Dynamics*, 26:833–888, 2003.

[78] H. Karimi, E.J. Davison, and R. Iravani. Multivariable servomechanism controller for autonomous operation of a distributed generation unit: design and performance evaluation. *IEEE Transactions on Power Systems*, 25:853–865, 2010.

[79] F.P. Kelly, A.K. Maulloo, and D.K.H. Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operations Research Society*, 49:237–252, 1998.

[80] H.K. Khalil. *Nonlinear systems*. Prentice Hall, 1996.

[81] D.C. Kilper, G. Atkinson, S.K. Korotky, S. Goyal, P. Vetter, D. Suvakovic, and O. Blume. Power trends in communication networks. *IEEE Journal of Selected Topics in Quantum Electronics*, 17:275–285, 2011.

[82] J. Korb. Thermoregulation and ventilation of termite mounds. *Naturwissenschaften*, 90:212–219, 2003.

[83] M.R. Korupolu and M. Dahlin. Coordinated placement and replacement for large-scale distributed caches. *IEEE Transactions on Knowledge and Data Engineering*, 14:1317–1329, 2002.

[84] E. Koutsoupias and C. Papadimitriou. Worst case equilibria. *Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science*, pages 404–413, 1999.

[85] M. Krstić, I. Kanellakopoulos, and P. Kokotović. *Nonlinear and Adaptive Control Design*. John Wiley and Sons, New York, NY, 1995.

[86] M. Krstić and H-H. Wang. Stability of extremum seeking feedback for general nonlinear dynamic systems. *Automatica*, 36:595–601, 2000.

[87] V. Kumar, N. Leonard, and A.S. Morse, editors. *Cooperative control*. Springer, 2005.

[88] K. Kvaternik, J. Llorca, D. Kilper, and L. Pavel. A decentralized coordination strategy for networked multiagent systems. In *Proceedings of the* 50^{th} *Annual Allerton Conference on Communication, Control and Computing*, 2012.

[89] K. Kvaternik, J. Llorca, D. Kilper, and L. Pavel. Decentralized caching strategies for energy-efficient content delivery. In *Proceedings of the IEEE International Conference on Communications, Sydney, Australia*, 2014.

[90] K. Kvaternik and L. Pavel. Interconnection conditions for the stability of nonlinear sampled-data extremum seeking schemes. In *Proceedings of the 50th Conference on Decision and Control, and the European Control Conference (CDC-ECC)*, 2011.

[91] K. Kvaternik and L. Pavel. Lyapunov analysis of a distributed optimization scheme. In *Proceedings of the International IEEE Conference on Network Games, Control and Optimization (NetGCOOP)*, 2011.

[92] K. Kvaternik and L. Pavel. Analysis of decentralized extremum seeking schemes. Technical Report 1201, University of Toronto, Systems Control Group, 2012.

[93] K. Kvaternik and L. Pavel. An analytic framework for decentralized extremum-seeking control. In *Proceedings of the American Control Conference, Montreal, Canada*, 2012.

[94] K. Kvaternik and L. Pavel. A continuous-time decentralized optimization scheme with positivity constraints. In *Proceedings of the* 51^{st} *IEEE Conference on Decision and Control*, 2012.

[95] U. Lee, I. Rimac, D. Kilper, and V. Hilt. Toward energy efficient content dissemination. *IEEE Network*, 25:14–19, 2011.

[96] Z. Li, G. Simon, and A. Gravey. Caching policies for in-network caching. In *Proceedings of the 21st International Conference on Computer Communications and Networks (ICCCN)*, 2012.

[97] B. Llanas and C. Moreno. Finding the projection on a polytope: an iterative method. *Computers and Mathematics with Applications*, 32:33–39, 1996.

[98] J. Llorca, A.M. Tulino, K. Guan, J. Esteban, M. Varvello, N. Choi, and D.C. Kilper. Dynamic in-network caching for energy efficient content delivery. In *Proceedings of the 32nd IEEE International Conference on Computer Communications (INFOCOM)*, 2013, (to appear).

[99] S.H. Low and D.E. Lapsley. Optimization flow control–i: basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, 7:861 – 874, 1999.

[100] J. Lu and C.Y. Tang. Zero-gradient-sum algorithms for distributed convex optimization: the continuous-time case. *IEEE Transactions on Automatic Control*, 57:2348–2354, 2012.

[101] A. Mehrizi-Sani and R. Iravani. Potential-function based control of a microgrid in islanded and grid-connected modes. *IEEE Transactions on Power Systems*, 25:1883–1891, 2010.

[102] M.D. Mesarović, D. Macko, and Y. Takahara. *Theory of hierarchical, multilevel systems*. Academic Press, 1970.

[103] H. Miao, X. Liu, B. Huang, and L. Getoor. A hypergraph-partitioned vertex programming approach for large-scale consensus optimization. In *Proceedings of the IEEE International Conference on Big Data*, 2013.

[104] A.N. Michel and R.K. Miller. *Qualitative analysis of large scale dynamical systems*. Academic Press, 1977.

[105] S. Michel, K. Nguyen, A. Rosenstein, L. Zhang, S. Floyd, and V. Jacobson. Adaptive web caching: towards a new global caching architecture. *Computer Networks and ISDN Systems*, 30:2169–2177, 1998.

[106] M. Mitchell. Complex systems: network thinking. *Artificial Intelligence*, 170:1194–1212, 2006.

[107] B. Mohar. *Some applications of Laplace eigenvalues of graphs*, pages 225–275. Kluwer Academic Publishers, 1997.

[108] L. Moreau. Stability of multiagent systems with time-dependent communication link. *IEEE Transactions on Automatic Control*, 50:160–182, 2005.

[109] T. Motzkin. Sur quelques propriétés caractéristiques des ensembles convexes. *Atti della Accademia Nazionale dei Lincei. Classe di Scienze Fisiche, Matematiche e Naturali. Rend. Cl. Sci. Fis. Mat. Natur.*, 21:562–567, 1935.

[110] R. Murphey and P.M. Pardalos, editors. *Cooperative control and optimization*. Kluwer Academic Publishers, 2002.

[111] T. Nakagaki, H. Yamada, and Á. Tóhl. Maze-solving by an amoeboid organism. *Nature*, 407:470, 2000.

[112] A. Nedić. Asynchronous broadcast-based convex optimization over a network. *IEEE Transactions on Automatic Control*, 56:1337–1352, 2011.

[113] A. Nedić and A. Ozdaglar. On the rate of convergence of distributed subgradient methods for multi-agent optimization. *Proceedings of the* 46^{th} *IEEE Conference on Decision and Control, New Orleans, LA*, pages 4711–4717, 2007.

[114] A. Nedić and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54:48–60, 2009.

[115] A. Nedić and A. Ozdaglar. Subgradient methods for saddle-point problems. *Journal of Optimization Theory and Applications*, 142:205–228, 2009.

[116] A. Nedić, A. Ozdaglar, and P.A. Parrilo. Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, 55:922–938, 2010.

[117] R.R. Negenborn and J.M. Maestre. Distributed model predictive control. *IEEE Control Systems Magazine*, 34:87–97, 2014.

[118] G. Nicolis and I. Prigogine. *Self-organization in nonequilibrium systems*. John Wiley & Sons, 1977.

[119] F. Paganini. A unified approach to congestion control and node-based multipath routing. *IEEE Transactions on Networking*, 17:1413–1426, 2009.

[120] D. Pais and N.E. Leonard. Adaptive network dynamics and evolution of leadership in collective migration. (to appear).

[121] D.P. Palomar and M. Chiang. Tutorial on decomposition methods for network utlity maximization. *IEEE Journal on Selected Areas in Communications*, 24:1439–1451, 2006.

[122] Y. Pan. *A game theoretical approach to constrained OSNR optimization problems in optimal networks*. PhD thesis, University of Toronto, 2008.

[123] S. Podlipnig and L. Boszormenyi. A survey of web cache replacement strategies. *ACM Computing Surveys*, 35:374–398, 2003.

[124] B.T. Polyak. *Introduction to optimization*. Optimization Software, Inc., 1987.

[125] D. Popović, M. Janković, S. Manger, and A.R. Teel. Extremum seeking methods for optimization of variable cam timing engine operation. *IEEE Transactions on Control Systems Technology*, 14:398–407, 2006.

[126] I. Psaras, W.K. Chai, and G. Pavlou. Probabilistic in-network caching for information-centric networks. In *Proceedings of the second edition of the ICN workshop on Information-centric networking*, 2012.

[127] M. Raginsky and J. Bouvrie. Continuous-time stochastic mirror descent on a network: variance reduction, consensus, convergence. *Proceedings of the* 51^{st} *Conference on Decision and Control, Maui, HI*, pages 6793–6800, 2012.

[128] W. Ren. and R.W. Beard. *Distributed consensus in multi-vehicle cooperative control*. Springer, 2008.

[129] W. Ren and Y. Cao. *Distributed coordination of multi-agent networks*. Springer, 2011.

[130] J. Rifkin. *The third industrial revolution: how lateral power is transforming energy, the ecomoomy and the world*. Palgrave Macmillan, 2011.

[131] J. Rifkin. *The zero-marginal const society: the internet of things, the collaborative commons, and the eclipse of captialism*. Palgrave Macmillan, 2014.

[132] G. Rizzelli, A. Morea, M. Tornatore, and O. Rival. Energy-efficient traffic-aware design of on-off multi-layer translucent optical networks. *Computer Networks*, 56:2443–2455, 2012.

[133] A.W. Roberts and D.E. Varberg. Another proof that convex functions are lolocally lipschitz. *The American Mathematical Monthly*, 81:1014–1016, 1974.

[134] J. Robinson. An iterative method of solving a game. *Annals of Mathematics*, 54:296 – 301, 1951.

[135] P. Rodriguez, C. Spanner, and E.W. Biersack. Analysis of web caching architectures: hierarchical and distributed caching. *IEEE/ACM Transactions on Networking*, 9:404–418, 2001.

[136] R.O. Saber, J.A. Fax, and R.M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95:215–233, 2007.

[137] T. Samad and A.M. Annaswamy (Eds.). The impact of control technology: control for renewable energy and smart grids, 2011.

[138] J.S. Shamma, editor. *Cooperative control of distributed multi-agnent systems*. Wiley, 2007.

[139] G. Shi, K.H. Johansson, and Y. Hong. Reaching an optimal consensus: dynamical systems that compute intersections of convex sets. *IEEE Transactions on Automatic Control*, 58:610–622, 2013.

[140] G. Shi, A. Proutiere, and K.H. Johansson. Distributed optimization: convergence conditions from a dynamical system perspective. *arXiv:1210.6685v1*, 2012.

[141] A. Simonetto. *Distributed estimation and control for robotic networks*. PhD thesis, Delft University of Technology, 2012.

[142] R.D. Smith. The dynamics of internet traffic: self-similarity, self-organization and complex phenomena. *Advances in complex systems*, 14:905–949, 2011.

[143] E.D. Sontag. Comments on integral variants of ISS. *Systems and Control Letters*, 34:93–100, 1998.

[144] M.S. Stanković, K.H. Johansson, and D.M. Stipanović. Distributed seeking of nash equilibria in mobile sensor networks. In *49th IEEE Conference on Decision and Control, GA, USA*, 2010.

[145] S.S. Stanković, M.S. Stanković, and D.M. Stipanović. Decentralized parameter estimation by consensus based stochastic approximation. *IEEE Transactions on Automatic Control*, 56:531–543, 2011.

[146] H.E. Stanley, L.A.N. Amaral, S.V. Buldyrev, P. Gopikrishnan, V. Plerou, and M.A. Salinger. Self-organized complexity in economics and finance. *Proceedings of the National Academy of Sciences*, 99:2561–2565, 2002.

[147] N. Stefanović. Robust L_2 nonlinear control of EDFA with amplified spontaneous emission. Master's thesis, University of Toronto, 2005.

[148] A. Teel and L. Praly. Global stabilizability and observability imply semiglobal stabilizability by output feedback. *Systems and Control Letters*, 22:313–325, 1994.

[149] A.R. Teel, J. Peuteman, and D. Aeyels. Semi-global practical asymptotic stability and averaging. *Systems and Control Letters*, 37:329–334, 1999.

[150] A.R. Teel and D. Popović. Solving smooth and nonsmooth multivariable extremum seeking problems by the methods of nonlinear programming. In *Proceedings of the American Control Conference, VA, USA*, 2001.

[151] A. Tero, S. Takagi, T. Saigusa, K. Ito, D.P. Bebber, M.D. Fricker, K. Yumiki, R. Koboayashi, and T. Nakagaki. Rules for biologically inspired adaptive network design. *Science*, 327:430–442, 2010.

[152] K.I. Tsianos, S. Lawlor, and M.G. Rabbat. Consensus-based distributed optimization: practical issues and applications in large-scale machine learning. *Proceedings of the Fiftieth Annual Allerton Conference, Monticello, IL*, pages 1543–1551, 2012.

[153] K.I. Tsianos and M.G. Rabbat. Consensus-based distributed online prediction and estimation. *Proceedings of the Global Conference on Signal and Information Processing*, pages 807–810, 2013.

[154] J.N. Tsitsiklis. *Problems in decentralized decision making and computation*. PhD thesis, Massachusetts Institute of Technology, 1984.

[155] J.N. Tsitsiklis, D.P. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, AC-31:803–812, 1986.

[156] A.M. Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London*, 237:37–72, 1952.

[157] O. Vermesan and P. Friess, editors. *Internet of things - converging technologies for smart environments and integrated ecosystems*. River Publishers, 2013.

[158] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet. Novel type of phase transition in a system of self-driven particles. *Physical Review Letters*, 75:1226–1229, 1995.

[159] A. Vishwanath, J. Zhu, K. Hinton, R. Ayre, and R. Tucker. Estimating the energy consumption for packet processing, storage and switching in optical-ip routers. In *Proceedings of the Optical Fiber Communication Conference*, 2013.

[160] P. Vixie. What DNS is not. *ACM Queue*, 7:10–15, 2009.

[161] D.D. Šiljak. *Large-scale dynamic systems*. Elsevier North-Holland, 1978.

[162] D.D. Šiljak. *Decentralized control of complex systems*. Academic Press, 1991.

[163] J. Wang and N. Elia. A control perspective for centralized and distributed convex optimization. *Proceedings of the 50thIEEE Conference on Decision and Control and the European Control Conference, Orlando, Florida*, pages 3800–3806, 2011.

[164] S.-H. Wang and E.J. Davison. On the stabilization of decentralized control systems. *IEEE Transactions on Automatic Control*, AC-18:473–478, 1973.

[165] E. Wei and A. Ozdaglar. Distributed alternating direction method of multipliers. *Proceedings of the 51st Conference on Decision and Control, Maui, HI*, page 5445, 2012.

[166] H.S. Witsenhausen. A counterexample in stochastic optimum control. *SIAM Journal of Control*, 6:131–148, 1968.

[167] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks*, 2005.

[168] F.E. Yates, editor. *Self-organizing systems: the emergence of order*. Plenum Press, 1987.

[169] D. Yuan, S. Xu, and H. Zhao. Distributed primal-dual subgradient methods for multiagent optimization via consensus algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, 41:1715–1723, 2011.

[170] A.I. Zečević and D.D. Šiljak. *Control of complex systems: structural constraints and uncertainty*. Springer, 2010.

[171] C. Zheng and R. Ordonez. *Extremum-seeking control and applications - a numerical optimization-based approach*. Springer, 2012.

[172] M. Zhu and S. Martinez. On distributed optimization under inequality constraints via lagrangian primal-dual methods. In *Proceedings of the 2010 American Control Conference, Baltimore, USA*, 2010.

[173] M. Zhu and S. Martinez. On distributed convex optimization under inequality and equality constraints. *IEEE Transactions on Automatic Control*, 57:151–164, 2012.