Distributed Newton-Seeking for Resource Allocation

by:

Matthew Schultz

A thesis submitted to the Department of Chemical Engineering in conformity with the requirements for the degree of Master of Applied Science

Queen's University

Kingston, Ontario, Canada

August 2022

Abstract

Real-time optimization (RTO) is a category of closed-loop process control techniques that is focused on the improvement of process performance. To develop effective strategies, models of the systems are usually required. To circumvent this issue, model-free techniques such as Extremum-seeking control (ESC) have been developed that can accomplish the RTO task in the absence of any knowledge of the process dynamics and of the objective function.

Using this optimization methodology, this thesis develops an approach to solve resource allocation problems in a fully distributed fashion where the separable cost function is unknown with dynamics in discrete time. To accomplish this, we utilize a distributed extremum-seeking scheme that estimates the cost function's gradient and Hessian to implement Newton step dynamics. It is shown that the extremum-seeking approach yields asymptotic convergence of the primal and dual problem to a neighbourhood of the optimal values.

We propose two strategies to solve finite resource allocation problems. The first strategy tackles the globally convex cost function case. It requires a simultaneous solution of the primal problem and its associated dual problem. The second strategy utilizes a more general approach with an pseudo-Hessian for the solution of local saddle point problems. It allows a full separation of the primal and dual dynamics.

Two simulation case studies are performed for each strategy. The first case is a small-scale 3-agent problem highlighting the differences between the strategies.

The second is a 25-agent problem that mimics more realistic scenarios. Finally, a stability analysis is conducted to characterize the transient performance of the two strategies.

Acknowledgements

There are so many people I would like to thank for their support. First, I would like to thank my supervisor Dr. Martin Guay for the endless inspiration, encouragement and guidance. I would also like to thank the amazing teachers and professors I had growing up who taught me how to enjoy learning and strive for greatness.

To all my friends I've made along the way, I don't know if I could have done this without you. I know you guys are always there for me. I'm excited to see what the future holds for all of us and I can't wait to find out together.

Finally, I would like to say thank you to my family for their unwavering support and for pushing me in the correct direction, especially when I'm being stubborn. Your love and understanding has allowed me to pursue my passions and I am extremely grateful.

Its been a long journey but we made it!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The emergence of technologies and data-driven decision making strategies brought an abundance of new large-scale control strategies. One prevalent area of research has focused on the design of distributed optimization systems operated on multi-agent communication networks. In these systems, networks distribute tasks to individual agents that communicate to cooperatively minimize their local costs. An added benefit of this autonomy is an increase in the robustness of the system. These two types of systems are highlighted by Figures 1.1 and 1.2.

Significant research has been invested in Multi-Agent Systems (MAS) to understand their dynamics and assess their stability and performance for a wide range of problems. Such systems govern a collection of problems where a global convex cost function can be split into convex sub-problems and assigned to agents to optimize cooperatively and locally. The underlying dynamics of these problems

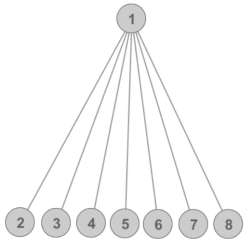

Figure 1.1: Centralized network

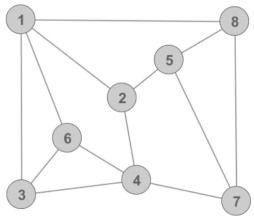

Figure 1.2: Distributed Network

are expressed using graph theoretical techniques, as described in Section 2.2.

1.1 Motivation

The focus of this thesis is a strategy to solve resource allocation problems using a distributed network that avoids the need for an initialization strategy. In this class of problems, interconnected agents cooperatively minimize the sum of their cost functions subject to a resource constraint on the network. This class of problems has been an active area of research in the study of multi-agent systems. Our secondary objective is to incorporate this model-free extremum-seeking control methodology to Newton's method on each node. This allows each agent to opti-

mize its locally unknown cost function, requiring only the exchange of neighbouring dual variable information. Consequently, each agent can preserve the privacy of its data allowing for a more secure network.

The approach proposed utilizes the Lagrangian of a local constrained optimization problem whose Lagrange multipliers are made to achieve consensus. In the first strategy, the primal and dual variables are coupled. In the second strategy, a pseudo-Hessian is used to decouple the dual variables from the primal problem using an alternative solution method for saddle-point problems. To find the optimal solution, a Newton consensus method is proposed in which the gradient and Hessian of the cost functions are estimated using an extremum seeking architecture. Finally, Lyapunov methods are applied to assess the stability of the optimum of the resource allocation problem using the distributed network.

1.2 Organization of the Thesis

This thesis is organized as follows.

In Chapter 2, background is presented to provide context for the foundation of this thesis. It covers developments in MAS, graph theory, decomposition methods, resource allocation problems, extremum seeking control and consensus algorithms.

The resource allocation problem is presented in Chapter 3, and the underlying assumptions of the solution strategy are stated. The decomposition strategy is presented to provide an understanding of the solution strategies along with a sum-

mary of how the extremum-seeking architecture are implemented in a distributed fashion. The two solution strategies are proposed.

Chapter 4 discusses the first solution strategy, the Newton-seeking Distributed Resource Allocation approach. The stability analysis of the algorithm is discussed in which a Lyapunov stability analysis is considered. Finally, both the model-free and model based formulations of the proposed algorithm are tested using two simulation examples.

In Chapter 5, the second solution strategy is analyzed. This approach generalizes the first strategy to handle both saddle-point problems and local minima/maxima problems. A stability analysis is presented for the new dynamics.

In Chapter 6, the conclusions of the performance and stability are presented, summarizing the work followed by the recommendations for future work.

Chapter 2

Background

This chapter provides background on fundamental material that is necessary to address the development, analysis and testing of the techniques proposed in this thesis. In Section 2.1, MAS are introduced. The objective is to provide a basic exposition of the class of problems that can be addressed in this framework. In Section 2.2, an introduction to graph theory is presented. In Section 2.3, the two most common decomposition methods for optimization problems are described with examples. Section 2.4 focuses on the analysis of resource allocation problems, providing applications of their use and a background on how these problems are formulated. In Section 2.5, extremum seeking control is introduced and the theory behind the technique is described. Distributed extremum seeking control is also presented. Finally, Section 2.6 reviews consensus algorithms and their importance in industry and academia, along with a review of some recent developments.

2.1 Multi-Agent Systems

A common approach to minimize the solution time of large-scale optimization problems is to split the problem into smaller sub-problems that can be solved locally using networked processors (agents). Such networks are known as MAS and are commonly used in the development of distributed solution techniques [1]. Key characteristics specific to the problems considered in this thesis include the ability of these agents to exchange local information with neighbouring agents. The connectivity of the communication network will impact the efficiency of the distributed agents. Furthermore, each agent is only aware of its own sub-problem without complete knowledge of the global problem. Three distinct architectures are prevalent in industry and academia.

Centralized networks encompass those where a single node (agent) is connected to every other agent in the network without any other links between nodes. With this architecture, the central node acts as a supervisor, creating a database of all state information and coordinating every other node to guarantee the optimal solution is found. This is often at the expense of solution time, as every node must relay its information to a single node. Furthermore, any failure of the master agents would lead to the failure of the entire network, leading to a loss of robustness [2].

Decentralized networks describe systems in which each agent only has access to its own local information and has the ability to set its local decision variables. In these networks, no information is exchanged and the dynamics of the system

are dictated by the aggregate decisions of the nodes. The lack of communication in these networks limit their ability to reach the optimum of a centralized problem.

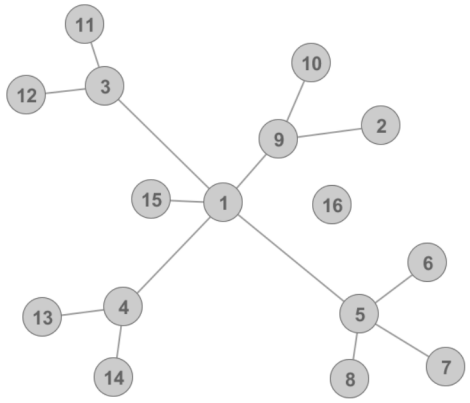

Figure 2.1: Decentralized Network

Distributed networks are peer-to-peer architectures where nodes are connected to a subset of the total nodes in the system [2]. Each node solves a local problem and checks their solution with adjacent nodes using a communication protocol. Each agent uses this exchange of information to update their own decision variables until the global problem is optimized. It is this class of networked systems that is considered throughout this thesis.

The class of problems commonly addressed by distributed networks are distributed optimization problems. A prevalent class of problems in distributed optimization are consensus problems where the objective is to achieve consensus on local decision variables that minimize the sum of all local costs. This class of problems has been well characterized in the literature [3–6]. Another important

class of problems that have been solved over distributed networks are resource allocation problems [7] [8]. These problems will be discussed in Section 2.6 and Section 2.4, respectively. The two main methodologies of how nodes optimize their sub-problem stems from ideas in game theory, namely cooperative and competitive games. Cooperative strategies encompass those where the nodes work together to solve the global problem. Agents work cooperatively by choosing decision variables that will both positively impact their own problem along with their neighbouring nodes' [9–11]. In contrast, competitive strategies implement agents with non-aligned goals that act selfishly. In these systems, nodes will choose the best input for their own problem at the expense of their neighbours; generally resulting in a Nash equilibrium between agents [12, 13]. Each strategy has its own strengths and weaknesses. Cooperative algorithms are more common in finding solutions to global problems while competitive games are used to simulate economical and societal settings [14].

To mirror more realistic conditions, additional environmental specifications can be made. Communication protocols can be operated using synchronous or asynchronous information updates. In synchronous problems, each agent will communicate and update their own decision variables at the same time. This specification is less realistic on a broader scale as the agents can operate on different time-scales. In contrast, asynchronous problems allow agents to iterate their own sub-problem as frequently as possible [15] [16]. Such problems can run into issues with cooperative strategies in which some agents can iterate at a faster rate. This generally leads to suboptimal global solutions [15].

2.2 Graph Theory

The distributed communication networks for MAS are usually expressed as graph topologies. This Section reviews some key elements from graph theory that are used to build the system dynamics throughout this thesis.

The communication network is represented by a graph G. This graph is an ordered pair $(V(G), E(G))$ consisting of a set $V(G)$ of vertices and a set $E(G)$ of edges. The elements $V(G)$ represent agents in the communication network while the elements of $E(G)$ identify the communication links between agents. The order of the graph is determined by the number of connected vertices $|V| = m$.

To describe the structure of the communication network, the three following matrices can be constructed. The adjacency matrix $A \in \mathbb{R}^{m \times m}$ describes how the agents are linked to one another. Elements are defined such that $a_{i,j} = 1$ whenever $(i, j) \in E$ for each $v_i, v_j \in V$ and $a_{ij} = 0$ otherwise. Depending on the topology of the network, the communication links can either be directed, indicating a one-way flow of information or undirected where both elements can exchange information freely [17]. In the case of an undirected graph, $(i, j) \in E$ whenever $(j, i) \in E$. Furthermore, if the graph is undirected, A is a symmetric matrix with $A^\top = A$.

The degree matrix $D \in \mathbb{R}^{m \times m}$ is a diagonal matrix with elements d_{ii} indicating how many nodes are incident to it, with $d_{ii} = \sum_{j=1}^{m} a_{ij}$. Weights can be assigned to communication links to prioritize specific channels, creating weighted

graphs. In this case, the degree matrix is then determined by $d_{ii} = \sum_{j=1}^{m} w_{ij}$.

The final matrix of interest is the Laplacian matrix $\mathcal{L} \in \mathbb{R}^{m \times m}$. This matrix is given by $\mathcal{L} = D - A$. A key characteristic of the Laplacian matrix of a communication network is that the sum of the elements in each row and column is equal to 0 for undirected graphs. This is important in the context of MAS as the Laplacian matrix has important properties that can be exploited in the stability analysis.

Figure 2.2: Connected unweighted undirected graph

As an example, we consider the simple, undirected, graph in Figure 2.2. The matrix representations of this graph is given by:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathcal{L} = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

Creating distributed dynamics that work with robust networks has proven to be a prevalent research area. Such challenges include working with switching [18], time-varying [15, 16, 19] , and plug and play topologies [20]. Furthermore, prob-

lems with restrictive communication protocols including asynchronous dynamics or time-delays are especially difficult as agents will be optimizing their local problem based on antiquated information. To circumvent this issue, many algorithms have implemented rules to ensure that, either topological information is instantaneously updated or not used at all.

2.3 Decomposition Methods

In this section, we present an overview on two common approaches to decompose convex optimization problems, using primal and dual decomposition. These decomposition techniques are used when sub-problems have coupled variables or constraints that cannot be solved directly.

2.3.1 Primal Decomposition

Consider the unconstrained optimization problem (2.1). It can be seen that u and v are local variables for f_1 and f_2, respectively. Furthermore, y appears in both f_1 and f_2, thereby coupling them. To optimize this problem, an iterative strategy is required. The problem is first split into a master problem, where the coupling variable y is held constant, and multiple sub-problems with functions of only local variables [21].

This strategy is shown in the following example from Boyd et al. [22]

$$\min_x f(x) = f_1(u, y) + f_2(v, y) \tag{2.1}$$

where $x = (u, v, y)$. The sub-problems are posed as

$$\phi_1(y) = \min_u f_1(u, y), \quad \phi_2(y) = \min_v f_2(v, y)$$

with the master problem

$$\min_y \phi_1(y) + \phi_2(y).$$

Once the local variables are optimized by the sub-problems, the master problem optimizes the complicating variable and the process is repeated. The iterative nature of these solutions stem from the relationship between the master and sub-problems and relies on a time-scale difference. While the sub-problems are constantly optimizing the local variables, the master problem has access to the optimal local variables only. This relationship is depicted in Figure 2.3.

Primal decomposition has many applications, notably in communication networks that utilize multi-input multi-output systems. Specifically in the design of MIMO transceivers [23] [24] and beamforming [25].

2.3.2 Dual Decomposition

Consider the constrained optimization problem (2.2). This problem presents a constraint that is imposed on both f_1 and f_2, coupling the functions [21]. To decompose this problem a dual decomposition strategy is utilized [22].

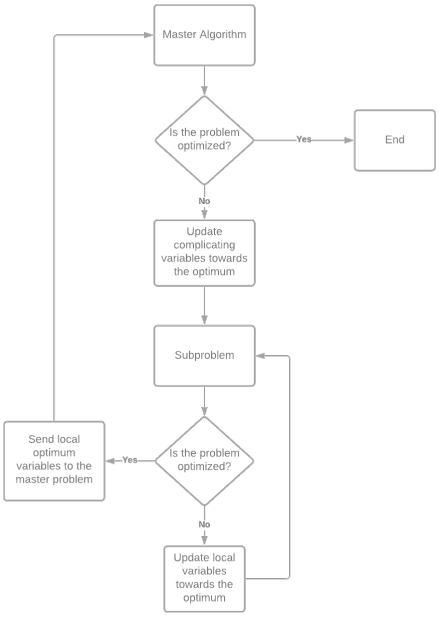

Figure 2.3: Example of algorithm flowchart for primal decomposition

$$\min_{x} f(x) = f_1(u, y_1) + f_2(v, y_2) \tag{2.2}$$

$$subject\ to \quad y_1 = y_2 \tag{2.3}$$

First, the method of Lagrange multipliers is applied, with ν as the dual variables associated with each constraint

$$L(u, y_1, v, y_2, \nu) = f_1(u, y_1) + f_2(v, y_2) + \nu^T y_1 - \nu^T y_2.$$

The associated Lagrangian can be fully separated with the dual function posed as

$$g(\nu) = g_1(\nu) + g_2(\nu)$$

where

$$g_1(\nu) = \inf_{u,y_1}(f_1(u, y_1) + \nu^T y_1), \quad g_2(\nu) = \inf_{v,y_2}(f_2(v, y_2) - \nu^T y_2).$$

The sub-problems are solved in parallel with their infimums relayed back to the master problem. The master problem is posed as

$$\max_{\nu} g(\nu)$$

and the problem can be solved iteratively.

These examples presented both a primal and dual decomposition where only a single decomposition is necessary. In both resource allocation problems and consensus problems, equality constraints are enforced resulting in many dual decomposition strategies [26] [27]. The distributed resource allocation strategy posed in this thesis utilizes consensus dynamics and two dual decompositions are required as shown in [28].

2.4 Resource Allocation Problems

As previously mentioned, researchers have taken an interest in MAS to solve distributed convex optimization problems. Resource allocation problems are one such problem with applications in power systems [29,30], transportation networks [31,32], communication networks [33,34] and supply chain management [35,36]. In this thesis, we pose distributed dynamics that asymptotically optimize resource

allocation problems.

2.4.1 Problem Description

These problems are of the form:

$$\min_{x \in \mathbb{R}^m} \quad f(x) = \sum_{i=1}^{m} f_i(x_i) \tag{2.4}$$

$$subject\ to \quad \sum_{i=1}^{m} x_i = R, \tag{2.5}$$

where $R \in \mathbb{R}$ is defined as the total resource that must be optimally allocated across each node, $x_i \in \mathbb{R}^m$.

Over the last few years, interest in resource allocation problems has grown leading to many developments in a wide range of problems. One common solution strategy is to develop primal-dual algorithms based on the Karush-Kuhn-Tucker conditions [37–40]. In these methodologies, the objective functions are decomposed and each agent is assigned its respective primal and dual variables. The agent's primal dynamics are used to optimize the primal variables, while it communicates the dual variables with neighbouring agents to cooperatively optimize the consensus and resource allocation sub-problems. In [37], the dual consensus and resource allocation dynamics can be freely communicated to neighbouring agents while protecting the privacy of each agent's primal variables.

In contrast, another method to ensure optimality is to implement penalty functions as done in [41]. The penalty terms added to the objective functions in these strategies restrict the feasible solution set and allow interior-point strategies to be

implemented. These strategies are prevalent in economic dispatch problems applications which are an application of resource allocation problems. Penalty function methodologies can face multiple drawbacks associated with noisy measurement signals and the initialization of parameters. In [42] parameter initialization is addressed for the finite-time problem by providing an alternative suboptimal solution. Further concerns using penalty methods stem from initialization of agent states based on their capacity constraints in dynamic communication networks. To address this problem, [20] and [43] developed initialization-free approaches for dynamic communication networks, allowing plug and play operation.

The directionality of the communication network dictates the flow of information in the system. In [44], a non-negative surplus-based algorithm is developed to converge to the global minimizer under the assumption that the time-varying digraph is jointly strongly connected. Distributed resource allocation for weight-balanced and weight-unbalanced digraphs were approached in [45] and [46], respectively. An alternative approach to Lagrangian and penalty function-based strategies was presented in [47], where the weights of the communication network were iteratively scaled. In their algorithm, each node updates its local variable based on the difference between its own cost function and its neighbours using a weighted gradient. In [48], this approach was extended using projection algorithms and LaSalle's invariance principle.

In [49], Alam et al. proposed a communication-efficient distributed multi-resource allocation strategy that took advantage of an additive-increase multiplicative-decrease (AIMD) stochastic algorithm. In this algorithm, connected nodes acquire

an increasing share of the global resource until the resource has been exceeded. After the threshold has been surpassed, each node responds in a probabilistic manner to reduce the demand.

Many of these optimization algorithms make use of game theory such that the nodes come to a Nash equilibrium at the optimal value [50]. These strategies can be developed in real-time as illustrated in [51] [52]. These problems are defined as dynamic problems as the cost functions are also functions of time.

2.5 Extremum Seeking Control

Extremum Seeking Control (ESC) was first proposed by M. Leblanc in 1922 to maximize the power transfer between a tram car and an electrical transmission line [53]. Following Leblanc's proposal, interest in ESC grew; however without any proof of stability for these control systems, interest tapered off with the onset of adaptive control. In 1997, Krstić and Wang were able to successfully prove the stability of ESC, providing the fundamental analysis that this technique required [54]. This real-time optimization (RTO) technique is a model-free control strategy that relies on the measurement of the objective function of interest. This technique has proven to be beneficial in practice when dealing with systems with complex or unknown dynamics.

2.5.1 Dynamics

For the Single-Input-Single-Output (SISO) case, the dynamics of the system are described by (2.6) with the associated objective function (2.7)

$$\dot{x} = f(x, u) \tag{2.6}$$

$$y = h(x) \tag{2.7}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}$ is the input, $y \in \mathbb{R}$ is the output. Furthermore, $f : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ and $h : \mathbb{R}^n \to \mathbb{R}$ are smooth. The goal of the ESC is to determine the optimal x^* and u^* such that our system is at equilibrium, $f(x^*, u^*) = 0$, with a minimized objective function $h(x^*)$. We assume that there exists a smooth control law

$$u = \alpha(x, \theta) \tag{2.8}$$

parameterized by a scalar parameter θ which takes the form

$$\theta = \hat{\theta} + asin(\omega t). \tag{2.9}$$

where $a\sin(\omega t)$ is defined as the dither signal with amplitude a. Using (2.8), the dynamics are written as

$$\dot{x} = f(x, \alpha(x, \theta)) \tag{2.10}$$

and the closed-loop system has an equilibria parameterized by θ. We make the following assumptions:

Assumption 1.1 *There exists a smooth function* $l : \mathbb{R} \to \mathbb{R}^n$ *such that*

$$f(x, \alpha(x, \theta)) = 0 \quad for\ x = l(\theta) \tag{2.11}$$

Assumption 1.2 *For each* $\theta \in \mathbb{R}$ *the equilibrium* $x = l(\theta)$ *of the system is locally exponentially stable.*

We can then assume the control law (2.8) is able to exponentially stabilize any equilibria that θ can produce. This assumption is non-restrictive in the sense that it can be readily relaxed, however when stated in this manner it must be assumed that the controller can stabilize the unknown non-linear system locally [54]. It further states that the controller does not need implicit understanding of the model $f(x, u)$ or $l(\theta)$.

Our final assumption is the existence of a unique optimum for the steady-state objective $y = h(l(\theta))$.

Assumption 1.3 *There exists* $\theta^* \in \mathbb{R}$ *such that*

$$(h \circ l)'(\theta^*) = 0$$
$$(h \circ l)''(\theta^*) < 0.$$

ESC has been applied to a wide range dynamic systems including bioreactors [55,56], mobile sensor networks [57,58], active breaking systems [59], DC motors

[60] and photovoltaic cells [61–63].

2.5.2 Controller Design

Through the injection of a sinusoidal dither signal, ESC strategies can check both sides of the estimated extremum point, confirming that a local minima is found. The perturbation input will create a periodic response that is either in phase or out of phase with the periodic response of the output. When the two sinusoids are in phase, this indicates that the current estimation of the extremum point is less than the optimal extremum point. Furthermore, if the sinusoids are out of phase, the current estimation of the extremum point is greater than the optimal solution [54]. Finally, the estimated extremum will match the optimal extremum when the two sinusoids annihilate one another. This relationship allows the control law to drive the estimate towards the optimal point, thereby optimizing the problem.

These perturbation inputs are generally slow so that the dynamics of the system will appear static. This allows the extremum point to be determined and tracked without impacting the dynamics. The block flow diagram for the classic extremum-seeking controller is shown in Figure 2.4.

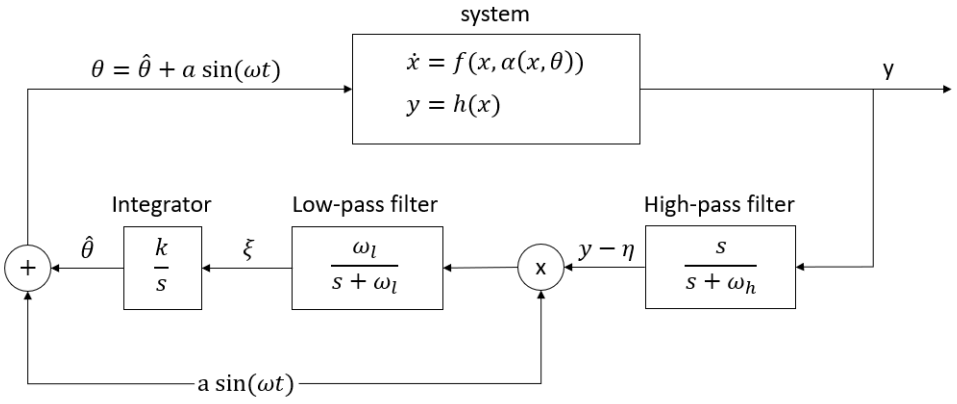

Figure 2.4: Architecture for classic extremum seeking control algorithms

The dynamics of the plant in Figure (2.4) can be summarized as follows

$$\dot{x} = f(x, \alpha(x, \hat{\theta} + asin(\omega t))$$
$$\dot{\hat{\theta}} = k\xi$$
$$\dot{\xi} = -\omega_l \xi + \omega_l (y - \eta) asin(\omega t)$$
$$\dot{\eta} = -\omega_h \eta + \omega_h y$$

where ξ is defined as the estimate of the gradient of the cost function. These dynamics can then be expressed in terms of the time scale by making the following substitution $\tau = \omega t$ and differentiating with respect to τ.

For systems with known input and output delays, [64] presents a feedback predictor based on perturbation estimates of an unknown model. In [65], an extremum seeking scheme for time-varying parameters was developed using a gen-

eralized averaging analysis of the perturbation signal in [66]. In systems with a time-varying extremum, [67] proposes a real-time gradient estimation along with gradient search subsystems. In [68] this problem is addressed with the use of high gain estimators and filters to generate an almost invariant manifold. This manifold is used in conjunction with an adaptive estimator to create a parameter update law.

In [69], Ghaffari et al. presents an approach to estimate the Hessian of a cost function, allowing for Newton-based optimization techniques with user assignable convergence rates. This approach was later applied in [70] to maximum power point tracking for the design of photovoltaic microconverter systems. This extremum seeking technique is applied to distributed optimization in this thesis.

2.5.3 Distributed Extremum Seeking Control

In the context of MAS, many distributed algorithms have incorporated model-free optimization techniques. In [71], an ES scheme was developed for convex optimization problems with inequality constraints based on saddle-point dynamics. In [72], a distributed dual-mode ESC was designed to address equality constraints and was applied to resource allocation problems.

Distributed ESC strategies have been shown to work in cooperative games. In [73], a cooperative ESC was used to optimize the physical spacing of satellites to ensure coverage of a communication networks. In non-cooperative games, a distributed ESC is utilized to develop a Nash seeking scheme [13]. In this approach, the convergence conditions coincided with that of model-based strategy.

Krstić and Wang's original analysis has been generalized in the discrete time

[74]. In discrete time, the technique proposed in [75] combines a dynamic average consensus algorithm with a proportional integral ESC to minimize an unmeasured total cost in a distributed setting.

2.6 Consensus algorithms

Consensus problems are another class of optimization problems that have sparked interest in recent years. These problems are defined by convex, fully separable objective function with solution spaces constrained such that each decision variable, x_i, is equal. In distributed settings, these algorithms allow agents to achieve consensus on the decision variables. These problems have a variety of applications in sensor networks [76–78], vehicle coordination [79, 80] and regression [81–83].

2.6.1 Problem Description

Formally, these problems are posed as follows

$$\min_x f(x) = \sum_{i=1}^{m} f_i(x_i) \tag{2.12}$$

$$\text{s.t. } x_1 = \ldots = x_m \tag{2.13}$$

where $x_i \in \mathbb{R}^p$. As previously explored above, the coupling between the agents arises from the constraints. There are two general solution strategies for these optimization problems. Commonly, the Lagrangian of (**??**) is taken and iteratively minimized [84, 85]. Alternatively, distributed sub-gradient methods can

be utilized to optimize non-smooth cost functions [86]. The constraint in problem (2.12) can be reformulated using the structure of the communication network via its Laplacian. These optimization problems have been posed using a variety of approaches subject to a wide range of communication protocols. In [87], fixed step-size algorithms were proposed that achieve better convergence rates when compared to algorithms with decreasing step-sizes. In [26, 88], consensus is achieved with a dynamic objective function. For time-varying objective functions, the analysis presented in [26] showed that the corresponding time-varying optimal values converged up to a non vanishing steady-state bias that depends on the eigenvalues of the communication network's Laplacian. The alternating direction method of multipliers (ADMM) methodology was applied to consensus optimization problems in [84], this technique achieves linear convergence rates. In [89], a Newton-Raphson method was applied that can achieve convergence rates that are comparable to ADMM.

2.7 Summary

In this chapter, the elements required for the design of the proposed distributed Newton-seeking algorithm to tackle resource allocation problems were presented. The general concepts behind each element are explained. MAS with cooperative strategies are often preferred to solve large-scale optimization problems of the

form

$$\min_{x} \quad f(x) = \sum_{i=1}^{m} f_i(x_i)$$

where $x \in \mathbb{R}^p$. Such problems frequently utilize concepts from graph theory to highlight the main properties of their communication networks.

We seek to solve a class of resource allocation problems. These problems are usually encountered when addressing communication networks, transportation networks and power systems. Their efficient solution is an important problem that will be required in many smart city technologies and the Internet of Things.

To create a robust and secure algorithm, a peak seeking scheme is implemented to estimate both the gradient and Hessian of each agent's unknown cost function. By estimating both the gradient and the Hessian, Newton's method dynamics are utilized, which have proven to be more efficient when compared to sub-gradient and gradient descent algorithms [90].

Consensus algorithms are reviewed along with two common solution strategies that utilize the Lagrangian of the objective function or the sub-gradient of the system. Novel developments and their results were discussed in Section 2.6.

Chapter 3

Solution strategy methodology

In this chapter we develop the methodology to solve distributed resource allocation problems. The general form of the problem is discussed in Section 3.1 along with 2 assumptions that characterize permissible objective functions. The decomposition strategy and critical points are developed in Section 3.2 providing the framework for the dynamics explored in Chapters 4 and 5. Finally, the distributed extremum-seeking architecture is developed in Section 3.3.

In the proposed algorithms, it is assumed that the network connectivity is known while the local cost functions are unknown. Furthermore, each node is able to manipulate their decision variables and measure their cost functions. For each locally unknown cost function to be minimized, an extremum-seeking architecture is employed following Ghaffari [69] to approximate the function's gradient and Hessian. This allows the computation of a Newton update for the decision variables for each agent. Two dual decompositions are necessary. The solution

dynamics require a two time-scale approach between the computation of a Newton update and the implementation of a Newton consensus step.

3.1 Problem Description

The objective of this thesis is to develop dynamics to solve the following optimization problem in a fully distributed manner

$$\min_x f(x) = \sum_{i=1}^{m} f_i(x_i) \tag{3.1}$$

$$\text{s.t. } \sum_{i=1}^{m} x_i = R_j \tag{3.2}$$

where $x_i \in \mathbb{R}^p$, and each function $f_i : \mathbb{R}^p \to \mathbb{R}$ are twice continuously differentiable functions for $i = 1, \ldots, m$, and further, $R_j \in \mathbb{R}$ is the total resource to be allocated. We then make the following two assumptions about structure of the objective function.

Assumption 1 *The Hessian of function f_i is a positive definite constant matrix*

$$\alpha_1 I \leq \frac{\partial^2 f_i}{\partial x_i \partial x_i^T} \leq \alpha_2 I$$

for some constants α_1 and α_2 for all $x_i \in \mathbb{R}^p$.

This assumption specifies that the Hessians are bounded.

Assumption 2 *The global objective function*

$$f(x) = \sum_{i=1}^{m} f_i(x_i)$$

has a unique minimum at x_i^* *such that*

$$\frac{\partial f}{\partial x_i}|_{x_i^*} = 0.$$

The second assumption dictates that there is a vector x_i^* that globally minimizes the problem. In Section 5.2, we propose a strategy to tackle optimization problems with saddle-point structures.

To pose this problem in a distributed setting, we define a set of vectors $y_1, \ldots, y_p$, each in $\mathbb{R}^m$ that act as collectors for each dimension of the solution across every node

$$\begin{aligned} y_1 &= [x_1(1), x_2(1), \ldots, x_m(1)]^T, \\ y_2 &= [x_1(2), x_2(2), \ldots, x_m(2)]^T, \\ &\vdots \\ y_p &= [x_1(p), x_2(p), \ldots, x_m(p)]^T. \end{aligned}$$

Using this notation, the problem can be written as

$$\min_{y_1,\ldots,y_p} f(y) = \sum_{i=1}^{m} f_i(y_1(i),\ldots,y_p(i)) \tag{3.3}$$

$$subject\ to\ \sum_{i=1}^{m} y_j(i) = R_j \tag{3.4}$$

Using equation (3.3), each collection vector can then be treated as its own independent resource to be allocated. This allows multiple resources to be optimized within the same problem.

3.2 Decomposition Strategy

To create distributed dynamics, we utilize the Lagrangian of problem (3.3) through taking a dual decomposition and creating a primal-dual relationship similar to [37]. However, in [37] agents are split into groups and assigned two sets of decision variables for the local primal and dual resources. Instead, we further simplify the constraint (3.4) by assigning each node a fraction of the total available resource

$$y_j(i) - \frac{R_j}{m} = 0.$$

By posing the constraint in this manner, after the Lagrangian is taken, the optimization problem is posed as follows:

$$\min_{y} \max_{\lambda} = \sum_{i=1}^{m} f_i(y_1(i), \ldots, y_p(i)) + \sum_{j=1}^{p} (\sum_{i=1}^{m} y_j(i) - \frac{R_j}{m}) \lambda_j(i) \quad (3.5)$$

$$subject\ to\ \ \mathcal{L}\lambda_j = 0. \quad (3.6)$$

Where $\lambda_j \in \mathbb{R}^m$. The constraint (3.6) is known as a consensus constraint and for undirected communication networks, it will be satisfied for any vector $\boldsymbol{\lambda}$ once consensus among the local Lagrange multipliers is achieved. Without this constraint, the Lagrangian (3.5) will either be driven to the optimum at $y_j = \frac{R_j}{m}$ or the optimum of each $f_i(y_j(i))$, without finding a cooperative solution. We further decompose the problem to add the consensus constraint to the Lagrangian similar to [37]

$$L(y, \lambda, \mu) = \sum_{i=1}^{m} f_i(y_1(i), \ldots, y_p(i)) + \sum_{i=j}^{p} (\sum_{i=1}^{m} y_j(i) - \frac{R_j}{m}) \lambda_j(i) + \sum_{j=1}^{p} (\mathcal{L}\lambda_j) \mu_j^{\intercal}. \quad (3.7)$$

Where $\mu_j \in \mathbb{R}^m$. To create a parallelized form of this expression, such that each node has access to all resource allocation variables, λ_i, we define the matrix

$$\mathcal{M} = \mathcal{I}_{p \times p} \otimes \mathcal{L} \quad (3.8)$$

where $\mathcal{I}$ is defined as the identity matrix and $\mathcal{M} \in \mathbb{R}^{mp \times mp}$. The critical points of problem (3.7) are determined by differentiating the Lagrangian with respect to each of the three variables y, λ, μ as follows

$$\frac{\partial L}{\partial \mathbf{y}_j} = \frac{\partial f_i}{\partial \mathbf{y}_j} + \boldsymbol{\lambda}_j(i) \tag{3.9}$$

$$\frac{\partial L}{\partial \boldsymbol{\lambda}_j} = (\mathbf{y}_j - \frac{R_j}{m}) + \mathcal{L}\boldsymbol{\mu}_j \tag{3.10}$$

$$\frac{\partial L}{\partial \boldsymbol{\mu}_j} = \mathcal{L}\boldsymbol{\lambda}_j. \tag{3.11}$$

Using these derivatives, we define dynamics whose stable equilibrium is a critical point of the first order optimality conditions (3.9)-(3.11).

3.3 Distributed Extremum-Seeking

Model free techniques must be employed in the absence of known objective functions to estimate the local gradients and Hessians. In this section, we present the elements of the local Newton-seeking approach adapted from [69]. We define the perturbation signals for each agent as

$$\mathbf{s}^i(t) = \left[a_1 \sin(\omega_1^i t), \ldots, a_n \sin(\omega_n^i t)\right] \tag{3.12}$$

$$\mathbf{r}^i(t) = \left[\tfrac{2}{a_1} \sin(\omega_1^i t), \ldots, \tfrac{2}{a_n} \sin(\omega_n^i t)\right] \tag{3.13}$$

where the probing frequencies, $\begin{bmatrix} \omega_1^i & \omega_2^i & \dots & \omega_n^i \end{bmatrix}$ are assigned such that they obey the following rules:

1. $\omega_j^i \neq \omega_k^i$ for any $j \neq k$,
2. ω_j^i / ω_k^i are rational for any $j \neq k$,
3. $\omega_j^i + \omega_k^i \neq \omega_l^i$ for all j, k, l.

Where $i = 1, \dots, m$.We further define the matrix Υ to estimate the Hessians of each of our systems, following [69] as

$$\Upsilon_{jj}^i(i) = \frac{16}{a_j^2}(\sin^2(\omega_j^i t) - \frac{1}{2}) \tag{3.14}$$

$$\Upsilon_{jk}^i(i) = \frac{4}{a_j a_k} \sin(\omega_j^i t) \sin(\omega_k^i t). \tag{3.15}$$

The probing frequency rules are created to ensure that the product of the Υ perturbation signal and the feedback signal approach the true Hessian of the system as outlined in [69]. Through the averaging analysis, these estimations do not impact the dynamics of the system when taken over a finite period defined as follows

$$T(i) = 2\pi \times LCM\left(\frac{1}{\omega_j^i}\right).$$

Next, we define the distributed extremum-seeking dynamics for the primal

problem as follows

$$\dot{\hat{\mathbf{y}}}_j(i) = \rho_p \mathbf{v}_j(i) \tag{3.16}$$

$$\dot{\mathbf{v}}_j(i) = -\gamma_p(\sum_{j=1}^{n} \hat{H}_{jk}(i)\mathbf{v}_k(i) + \hat{\mathbf{g}}_j(i)) \tag{3.17}$$

$$\dot{\hat{\boldsymbol{\xi}}}_j(i) = -\omega_l(\hat{\boldsymbol{\xi}}_j(i) - \mathbf{r}_j^i(t)(f_i(\bar{\mathbf{y}}(i) + \mathbf{s}^i(t)) - \boldsymbol{\eta}^i)) \tag{3.18}$$

$$\dot{\hat{H}}_{jk}(i) = -\omega_l(\hat{H}_{jk}(i) - \Upsilon(t)(f_i(\bar{\mathbf{y}}(i) + \mathbf{s}^i(t)) - \boldsymbol{\eta}^i)) \tag{3.19}$$

$$\dot{\boldsymbol{\eta}}^i = -\omega_h(\boldsymbol{\eta}^i - f_i(\bar{\mathbf{y}}(i) + \mathbf{s}^i(t)) \tag{3.20}$$

where $\hat{\boldsymbol{\xi}}_j(i)$ estimates the gradient of the cost function f_i for each node and $\boldsymbol{\eta}^i$ is the high-pass filter that is used to estimate the cost function subject to the perturbation signal. The high pass filter, with gain, ω_h, is used to remove the DC component from the output signal. The low-pass filter, with gain, ω_h, attenuates the impact of high frequency signals arising from process noise or unmodelled dynamics. If either gain is too large, any disturbances in the signal can be amplified, leading to instability or a violation of the time-scale separation [91].

3.4 Conclusion

In this chapter, we presented an approach for the decomposition of a resource allocation problem to be solved in a distributed fashion. The resource allocation constraint is enforced by posing the optimization problem as a saddle-point problem; minimizing with respect to the primal variables and maximizing with respect

to the resource allocation Lagrange multiplier λ. Solving each agent's local problem then collectively solves the resource allocation problem when a consensus is achieved for the local Lagrange multipliers.

The strategies considered in the next two chapters rely on Newton's optimization method, requiring the gradient and Hessian of the objective functions. A model free extremum-seeking approach is employed to estimate the local gradients and Hessians when the cost functions are unknown. Finally, we present two resource allocation problems to test the strategies proposed in the next two chapters.

Chapter 4

Newton-Seeking Distributed Resource Allocation for Convex Cost Functions

In this chapter we develop the primal and dual dynamics to solve problem (3.7) in discrete-time. In Section 4.1, we develop the primal-dual algorithm. A stability analysis of the dynamics is presented in Section 4.2. Finally, the algorithm is tested using the two simulation examples presented in Section 4.3.

4.1 Solution Dynamics

The first strategy utilizes a Newton step in the primal to maximize the collection vectors, while letting the resource allocation vector, $\boldsymbol{\lambda}$, follow a gradient ascent

method to its optimum, ensuring that the sum of resources allocated is the total resource available. We also implement a Newton step for the dual consensus, $\boldsymbol{\mu}$, problem to ensure that the $\boldsymbol{\lambda}$ vector achieves a consensus among all the agents of the network. This creates a trade-off between maximizing the collection vectors using the primal problem and ensuring that the total resource is allocated using the dual problem.

Following [92], the primal and dual resource allocation dynamics for the Lagrangian (3.7) can be reduced to the following system of equations. From the primal critical point, $\boldsymbol{\lambda}$ is defined as the gradient of the local problem

$$\boldsymbol{\lambda}_j = -\frac{\partial f_i}{\partial \mathbf{y}_j}. \tag{4.1}$$

Furthermore, from the resource allocation critical point, for a given $\boldsymbol{\mu}$, our primal states are defined as

$$\mathbf{y}_j = \frac{R_j}{m} - \mathcal{L}\boldsymbol{\mu_i}. \tag{4.2}$$

Through this system of equations, both the primal and resource allocation states can be calculated directly. This method reduces the proposed primal-dual dynamics to optimizing the dual problem as defined in Section 4.1.2.

4.1.1 Primal Problem

To create a primal-dual algorithm to optimize Lagrangian (3.7), in this thesis we propose creating primal dynamics through the utilization of a Newton step using the partial derivative of the Lagrangian at fixed values of the Lagrange multipliers $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$

$$\mathbf{g} = \frac{\partial L}{\partial \mathbf{y}_j} = \frac{\partial f_i}{\partial \mathbf{y}_j} + (\boldsymbol{\lambda}_j)_i$$

along with its Hessian

$$\frac{\partial^2 L}{\partial \mathbf{y}_j \partial \mathbf{y}_k} = \frac{\partial^2 f_i}{\partial \mathbf{y}_j \partial \mathbf{y}_k} = H_{jk}(i).$$

Each agent, i, updates their local primal variables as follows:

$$\dot{\mathbf{v}}_j(i) = -\gamma_p(\sum_{k=1}^{n} H_{jk}\mathbf{v}_k(i) + \mathbf{g}_j(i)) \tag{4.3}$$

$$\dot{\mathbf{y}}_j(i) = \rho_p \mathbf{v}_j(i) \tag{4.4}$$

Furthermore, $\mathbf{v}$ acts as an auxiliary variable with a unique equilibrium at $\mathbf{v} = \mathbf{0}$. The equilibrium of each agent's gradient is

$$\frac{\partial f_i}{\partial \mathbf{y}_j} = -(\boldsymbol{\lambda}_j)_i. \tag{4.5}$$

satisfying the critical point (3.9). Furthermore each node is able to optimize its primal problem with only the knowledge of its local primal variable, $\mathbf{y}$, and dual $\boldsymbol{\lambda}$ without the complete knowledge of the system [93].

It is imperative to keep a time-scale separation between the dynamics such that the primal problem is solved with respect to current dual variables. To achieve this, we employed two strategies. To ensure that the primal variables update faster than their respected dual variables, we incorporate dynamic gains and step-sizes of varying magnitudes into the update steps. To increase the convergence rate, we further allow the dual problem to iterate numerous times before updating the primal problem.

4.1.2 Dual Problem

Using (3.7), the dual problem is taken as

$$\nu(\mu) = \sum_{i=1}^{m} \inf_{y_1,\ldots,y_p} f_i(y_1(i),\ldots,y_p(i)) + \sum_{j=1}^{p}(\sum_{i=1}^{m} y_j(i) - \frac{R_j}{m})\lambda_j + \sum_{j=1}^{p}\sum_{i=1}^{m}(\mathcal{L}\lambda_j)\mu_i \tag{4.6}$$

The gradient and Hessian of (4.6) following [93],

$$\nabla\nu(\boldsymbol{\mu}) = \mathcal{M}\boldsymbol{\lambda}(\boldsymbol{\mu}) \tag{4.7}$$

$$H(\boldsymbol{\mu}) = \mathcal{M}(\nabla^2 f(\mathbf{y}(\boldsymbol{\mu}))\mathcal{M}. \tag{4.8}$$

The consensus dynamics are developed in a distributed manner. To drive the system to its equilibrium position at the critical point, we introduce the auxiliary variables $\mathbf{w}, \mathbf{d}$ and $\mathbf{q}$ each in $\mathbb{R}^{m\times p}$. The dynamics of the auxiliary variables are given by:

$$\dot{\mathbf{w}}_r(k) = -\gamma_d(\sum_{j=1}^{m} \mathcal{L}_{k,j}\mathbf{w}_r(j) - \sum_{j=1}^{m} \mathcal{L}_{kj}\boldsymbol{\lambda}_r(j)). \tag{4.9}$$

This system has a unique equilibrium position at $\mathcal{M}\mathbf{w} = \mathcal{M}\boldsymbol{\lambda}$, which is identified as the gradient of the Lagrangian with respect to $\boldsymbol{\mu}$. By the diagonally dominant property of the Laplacian, the vector $\mathbf{w}$ reaches an ϵ close approximation of $\boldsymbol{\lambda}$. An ϵ-approximation is defined as:

Definition: $\boldsymbol{\epsilon}$-approximation [94]. *Let $\boldsymbol{x}^* \in \mathbb{R}^n$ be the solution to the symmetric diagonally dominant system of the form $M\boldsymbol{x}_0 = \boldsymbol{b}_0$. The vector $\tilde{\boldsymbol{x}} \in \mathbb{R}^n$ is defined as an ϵ-approximation, bounded by the M norm, to the exact solution, $\boldsymbol{x}^*$, if:*

$$\|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}\|_M \leq \epsilon\|\boldsymbol{x}^*\|_M.$$

By collecting the dual variable $\boldsymbol{\lambda}$'s from the neighbouring agents, this computation can be performed locally by each agent, i. We then define the distributed Newton step in the dual using equations (4.7) and (4.8) as

$$\dot{\mathbf{d}}_r(k) = -\gamma_d(\sum_{j=1}^{m} \mathcal{L}_{kj}\mathbf{d}_r(j) + \sum_{j=1}^{n} H_{rj}^{-1}\mathbf{w}_j(k)) \tag{4.10}$$

The dynamics of (4.10) has an equilibrium point at $\mathcal{M}\mathbf{d} = -H^{-1}\mathbf{w}$, where $\mathbf{d}$ is defined as a newton step for each fixed value of the vector $\boldsymbol{\mu}$. This computation can be performed locally as each agent collects neighbouring, $\boldsymbol{\lambda}$s, and without complete knowledge of each node's Hessian. Finally, as this Newton step reaches its equilibrium position, the consensus step is defined using $\mathbf{q}$ and $\boldsymbol{\mu}$ as

$$\dot{\mathbf{q}}_r(k) = -\gamma_d(\sum_{j=1}^{m} \mathcal{L}_{kj}\mathbf{q}_r(j) - \sum_{j=1}^{m} \mathcal{L}_{kj}\mathbf{d}_r(j)) \tag{4.11}$$

$$\dot{\boldsymbol{\mu}}_r(k) = \rho_d\mathbf{q}_r(k) \tag{4.12}$$

such that if $\mathbf{q}$ is at equilibrium, $\mathbf{q} = \mathbf{0}$, $\boldsymbol{\mu}$ will be in equilibrium and the dual problem will be optimized. To ensure this equilibrium point is approached, a gain close to 0 is applied to the forward-euler discretization method.

The dynamics in (4.10) can be used directly when the cost functions are known. This is due to the fact that the Hessians can be inverted directly by locally inverting each local Hessian matrix. In contrast, when the distributed extremum-seeking scheme outlined in Section 3.3 is utilized, some instabilities my arise if one considers explicit inversion of the local estimated Hessian. In [69], this problem is circumvented by inverting the local Hessian using a differential Riccati equation. In the unknown cost function case, we propose an alternative indirect Newton step to create stable dynamics that avoids the explicit inversion of the Hessian.

The resource allocation dynamics for $\boldsymbol{\lambda}$ employ a gradient based approach

directly following critical point (3.10)

$$\dot{\boldsymbol{\lambda}}_r(k) = \frac{\partial L}{\partial \boldsymbol{\lambda}_r(k)} = \mathbf{y}_r(k) - \frac{R_r}{m} + \sum_{j=1}^{m} \mathcal{L}_{kj} \boldsymbol{\mu}_r(j) \tag{4.13}$$

such that we can drive the dynamics in (3.10) to its equilibrium. Posing the dynamics in this manner allows each node to allocate more or less than their fraction, $\frac{R_r}{m}$, of the total resource. The consensus term, $\mathcal{M}\boldsymbol{\mu}_k$, compensates for the difference, ensuring that $\dot{\boldsymbol{\lambda}}_r(k) = \mathbf{0}$.

4.2 Stability Analysis

In this section, we assess the stability of the equilibrium points through conducting a Lyapunov analysis [95]. To accomplish this, the system is expressed in deviation form under the time-scale $\tau = \frac{t}{\epsilon}$. The dual consensus Lagrange multipliers are updated in the slowest time-scale, allowing the primal and dual resource allocation dynamics to achieve equilibrium for any given $\boldsymbol{\mu}$. We then apply theorem 1 from [92].

Theorem 1 *Consider the Newton resource allocation and consensus algorithm expressed in error. Assume that the communication network can be represented by an undirected graph. Let $\gamma > 0$. Then there exists a constant ϵ^* such that for any $0 < \epsilon < \epsilon^*$, the origin of the error system is a globally exponentially stable equilibrium.*

Proof. We first define the vector valued functions $\mathbf{y}(\boldsymbol{\mu})$, $\boldsymbol{\lambda}(\boldsymbol{\mu})$, $\mathbf{w}(\boldsymbol{\mu})$, $\mathbf{d}(\boldsymbol{\mu})$

and $\mathbf{q}(\boldsymbol{\mu})$ as the equilibrium for a fixed $\boldsymbol{\mu}$. In deviation form, the dynamics are

$$\frac{d\tilde{\mathbf{y}}}{d\tau} = \rho_p \tilde{\mathbf{v}} - \frac{\partial \mathbf{y}(\boldsymbol{\mu})}{\partial \boldsymbol{\mu}} \frac{d\boldsymbol{\mu}}{d\tau} \tag{4.14}$$

$$\frac{d\mathbf{v}}{d\tau} = -\gamma_p((\nabla^2_{yy} L(\mathbf{y}, \boldsymbol{\lambda}, \boldsymbol{\mu}))\mathbf{v} + \nabla_y L(\mathbf{y}, \boldsymbol{\lambda}, \boldsymbol{\mu})) \tag{4.15}$$

$$\frac{d\tilde{\boldsymbol{\lambda}}}{d\tau} = (\tilde{\mathbf{y}} - \frac{R}{m}) + \mathcal{M}\tilde{\boldsymbol{\mu}} - \frac{\partial \boldsymbol{\lambda}(\boldsymbol{\mu})}{\partial \boldsymbol{\mu}} \frac{d\boldsymbol{\mu}}{d\tau} \tag{4.16}$$

$$\frac{d\tilde{\mathbf{w}}}{d\tau} = -\gamma_d(\mathcal{M}\tilde{\mathbf{w}} - \mathcal{M}\tilde{\boldsymbol{\lambda}}) - \frac{\partial \mathbf{w}(\boldsymbol{\mu})}{\partial \boldsymbol{\mu}} \frac{d\boldsymbol{\mu}}{d\tau} \tag{4.17}$$

$$\frac{d\tilde{\mathbf{d}}}{d\tau} = -\gamma_d(\mathcal{M}\tilde{\mathbf{d}} + (\nabla^2 f(\mathbf{y}(\boldsymbol{\mu})))^{-1}\tilde{\mathbf{w}}) - \frac{\partial \mathbf{d}(\boldsymbol{\mu})}{\partial \boldsymbol{\mu}} \frac{d\boldsymbol{\mu}}{d\tau} \tag{4.18}$$

$$\frac{d\tilde{\mathbf{q}}}{d\tau} = -\gamma_d(\mathcal{M}\tilde{\mathbf{q}} - \mathcal{M}\tilde{\mathbf{d}}) - \frac{\partial \mathbf{q}(\boldsymbol{\mu})}{\partial \boldsymbol{\mu}} \frac{d\boldsymbol{\mu}}{d\tau} \tag{4.19}$$

$$\frac{d\tilde{\boldsymbol{\mu}}}{d\tau} = \epsilon\rho_d \mathbf{q}(\boldsymbol{\mu}) + \epsilon\rho\tilde{\mathbf{q}} \tag{4.20}$$

We then apply a singular perturbation analysis by defining the boundary layer dynamics by setting $\epsilon = 0$. This yields

$$\frac{d\tilde{\mathbf{y}}}{d\tau} = \rho_p \tilde{\mathbf{v}} \tag{4.21}$$

$$\frac{d\mathbf{v}}{d\tau} = -\gamma_p((\nabla^2_{yy} L(\mathbf{y}, \boldsymbol{\lambda}, \boldsymbol{\mu}))\mathbf{v} + \nabla_y L(\mathbf{y}, \boldsymbol{\lambda}, \boldsymbol{\mu})) \tag{4.22}$$

$$\frac{d\tilde{\boldsymbol{\lambda}}}{d\tau} = (\tilde{\mathbf{y}} - \frac{R}{m}) + \mathcal{M}\tilde{\boldsymbol{\mu}} \tag{4.23}$$

$$\frac{d\tilde{\mathbf{w}}}{d\tau} = -\gamma_d(\mathcal{M}\tilde{\mathbf{w}} - \mathcal{M}\tilde{\boldsymbol{\lambda}})) \tag{4.24}$$

$$\frac{d\tilde{\mathbf{d}}}{d\tau} = -\gamma_d(\mathcal{M}\tilde{\mathbf{d}} + (\nabla^2 f(\mathbf{y}(\boldsymbol{\mu})))^{-1}\tilde{\mathbf{w}})) \tag{4.25}$$

$$\frac{d\tilde{\mathbf{q}}}{d\tau} = -\gamma_d(\mathcal{M}\tilde{\mathbf{q}} - \mathcal{M}\tilde{\mathbf{d}}) \tag{4.26}$$

As previously mentioned, the primal and dual resource allocation dynamics can be reduced to the following system of equations for any fixed value of μ:

$$\boldsymbol{\lambda}(\boldsymbol{\mu}) = -\frac{\partial f(\mathbf{y}(\boldsymbol{\mu}))}{\partial y} \tag{4.27}$$

$$\mathbf{y}(\boldsymbol{\mu}) = \frac{R}{m} - \mathcal{M}\boldsymbol{\mu}. \tag{4.28}$$

In what follows, it is assumed that the local decision variables and Lagrange multipliers can be taken as the solution of the system of equations. This will be generalized in the next Chapter.

In the slow time-scale, the consensus of the resource allocation Lagrange multipliers is determined. The equilibrium point that indicates that consensus has been achieved occurs at $\mathbf{q} = \mathbf{0}$. To assess the stability of this origin, we first change coordinate systems using the nullspace of the Laplacian. We consider the m-dimensional vector $\mathbf{r}$ and a $m \times (m-1)$ matrix $\mathbf{R}$ such that:

$$\mathbf{r} = \frac{1}{\sqrt{m}}\mathbf{1}_m, \mathbf{r}^T\mathbf{R} = 0,$$

$$\mathbf{R}^T\mathbf{R} = I_{m-1}, \mathbf{R}\mathbf{R}^T = I_m - \frac{1}{m}\mathbf{1}_m\mathbf{1}_n^T.$$

The coordinate transformation then takes the following form:

$$\tilde{\mathbf{w}} = ([\mathbf{r}, \mathbf{R}] \otimes I_n)\mathbf{W}, \tilde{\mathbf{d}} = ([\mathbf{r}, \mathbf{R}] \otimes I_n)\mathbf{D}$$

$$\tilde{\mathbf{q}} = ([\mathbf{r}, \mathbf{R}] \otimes I_n)\mathbf{Q}, \tilde{\boldsymbol{\lambda}} = ([\mathbf{r}, \mathbf{R}] \otimes I_n)\mathbf{K}$$

$$\tilde{\mathbf{y}} = ([\mathbf{r}, \mathbf{R}] \otimes I_n)\mathbf{Y}, \tilde{\mathbf{v}} = ([\mathbf{r}, \mathbf{R}] \otimes I_n)\mathbf{V}$$

$$\tilde{\boldsymbol{\mu}} = ([\mathbf{r}, \mathbf{R}] \otimes I_n)\boldsymbol{\Pi}$$

and we can express the dynamics of the consensus steps (4.24)-(4.26) in the new coordinate system. The dynamics of $\mathbf{W}$ are:

$$\dot{\mathbf{W}} = -\gamma_d \begin{bmatrix} \mathbf{r}^T\mathcal{L}\mathbf{r} \otimes I_n & \mathbf{r}^T\mathcal{L}\mathbf{R} \otimes I_n \\ \mathbf{R}^T\mathcal{L}\mathbf{r} \otimes I_n & \mathbf{R}^T\mathcal{L}\mathbf{R} \otimes I_n \end{bmatrix} \begin{bmatrix} \mathbf{W}_1 - \mathbf{K}_1 \\ \mathbf{W}_2 - \mathbf{K}_2 \end{bmatrix}.$$

By the property of the Laplacian, one can write the dynamics of $\mathbf{W}$ as:

$$\dot{\mathbf{W}}_1 = 0$$

$$\dot{\mathbf{W}}_2 = -\gamma_d(\mathbf{R}^T\mathcal{L}\mathbf{R} \otimes I_n)(\mathbf{W}_2 - \mathbf{K}_2).$$

where $W_1 \in \mathbb{R}^m$ and $W_2 \in \mathbb{R}^m$. Similarly, we express (4.25) in terms of $\mathbf{D}$ as

$$\dot{\mathbf{D}} = -\gamma_d \left(\begin{bmatrix} \mathbf{r}^T\mathcal{L}\mathbf{r} \otimes I_n & \mathbf{r}^T\mathcal{L}\mathbf{R} \otimes I_n \\ \mathbf{R}^T\mathcal{L}\mathbf{r} \otimes I_n & \mathbf{R}^T\mathcal{L}\mathbf{R} \otimes I_n \end{bmatrix} \begin{bmatrix} \mathbf{D}_1 \\ \mathbf{D}_2 \end{bmatrix} + \left(\frac{\partial^2 f}{\partial \mathbf{Y} \partial \mathbf{Y}^T} \right)^{-1} \begin{bmatrix} \mathbf{W}_1^T & \mathbf{W}_2^T \end{bmatrix} \right).$$

To express the Hessian in the new coordinate system, we write

$$\frac{\partial^2 f}{\partial \mathbf{Y} \partial \mathbf{Y}^T} = (\begin{bmatrix} \mathbf{r}^T \\ \mathbf{R}^T \end{bmatrix} \otimes I_n) \nabla^2 f(\mathbf{y}(\boldsymbol{\mu}))([\mathbf{r}\ \mathbf{R}] \otimes I_n)$$

This allows us to state the dynamics of $\mathbf{D}$ as

$$\dot{\mathbf{D}}_1 = -\gamma_d(\mathbf{r}^T \otimes I_n)(\nabla^2 f(\mathbf{y}(\boldsymbol{\mu})))^{-1}([\mathbf{r}\ \mathbf{R}] \otimes I_n)\mathbf{W}_2$$
$$\dot{\mathbf{D}}_2 = -\gamma_d(\mathbf{R}^T \mathcal{L} \mathbf{R} \otimes I_n)\mathbf{D}_2 - \gamma_d(\mathbf{R}^T \otimes I_n)(\nabla^2 f(\mathbf{y}(\boldsymbol{\mu})))^{-1}([\mathbf{r}\ \mathbf{R}] \otimes I_n)\mathbf{W}_2.$$

where it is assumed without loss of generality that $W_1(0) = 0$. Finally, $\mathbf{Q}$ can be expressed as

$$\dot{\mathbf{Q}} = -\gamma_d \begin{bmatrix} \mathbf{r}^T \mathcal{L} \mathbf{r} \otimes I_n & \mathbf{r}^T \mathcal{L} \mathbf{R} \otimes I_n \\ \mathbf{R}^T \mathcal{L} \mathbf{r} \otimes I_n & \mathbf{R}^T \mathcal{L} \mathbf{R} \otimes I_n \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 - \mathbf{D}_1 \\ \mathbf{Q}_2 - \mathbf{D}_2 \end{bmatrix}$$

and the dynamics of $\mathbf{Q}$ are

$$\dot{\mathbf{Q}}_1 = 0$$
$$\dot{\mathbf{Q}}_2 = -\gamma_d(\mathbf{R}^T \mathcal{L} \mathbf{R} \otimes I_n)(\mathbf{Q}_2 - \mathbf{D}_2).$$

To create the Lypanuov function candidate for the consensus dynamics, we utilize the fact that the matrix $\mathbf{R}^T \mathcal{L} \mathbf{R} \otimes I_n$ is positive definite, allowing us to bound it by

the two constants $0 < \lambda_1 < \lambda_2$ as follows:

$$\lambda_1 |\tilde{\mathbf{w}}_2|^2 \leq \tilde{\mathbf{w}}_2^T (\mathbf{R}^T \mathcal{L} \mathbf{R} \otimes I_n) \tilde{\mathbf{w}}_2 \leq \lambda_2 |\tilde{\mathbf{w}}_2|^2.$$

Using these dynamics, we pose the quadratic Lyapunov function candidate

$$V_{b1} = \frac{\alpha}{2} \mathbf{W}_2^T \mathbf{W}_2 + \frac{\beta}{2} \mathbf{D}_2^T \mathbf{D}_2 + \frac{1}{2} \mathbf{Q}_2^T \mathbf{Q}_2. \tag{4.29}$$

The derivative of the Lyapunov function candidate is as follows

$$\begin{aligned} \dot{V}_{b1} = & -\gamma_d \alpha \mathbf{W}_2^T (\mathbf{R}^T \mathcal{L} \mathbf{R} \otimes I_n)(\mathbf{W}_2 - \mathbf{K}_2) - \gamma_d \beta \mathbf{D}_2^T (\mathbf{R}^T \mathcal{L} \mathbf{R} \otimes I_n) \mathbf{D}_2 \\ & - \gamma_d \beta \mathbf{D}_2^T (\mathbf{R}^T \otimes I_n)(\nabla^2 f(\mathbf{y}(\boldsymbol{\mu})))^{-1} ([\mathbf{r}\ \mathbf{R}] \otimes I_n) \mathbf{W}_2 \\ & - \gamma_d \mathbf{Q}_2^T (\mathbf{R}^T \mathcal{L} \mathbf{R} \otimes I_n)(\mathbf{Q}_2 - \mathbf{D}_2) \end{aligned} \tag{4.30}$$

Since the Hessian is assumed to be bounded, one can write that $|\frac{\partial^2 f}{\partial Y_2 \partial Y_2^T}| \geq M$ for some positive constant M. We apply Young's inequality to compute an upper

bound for the right hand side of the last inequality. This yields:

$$\begin{aligned}\dot{V}_{b1} \leq & -\gamma_d\alpha\mathbf{W}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\mathbf{W}_2 - \gamma_d\mathbf{Q}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\mathbf{Q}_2 \\ & -\gamma_d\beta\mathbf{D}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\mathbf{D}_2 + \frac{\gamma_d\alpha}{2k_3}\mathbf{W}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\mathbf{W}_2 \\ & +\frac{k_3\alpha\gamma_d}{2}\mathbf{K}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\mathbf{K}_2 + \frac{\gamma_d}{2k_2}\mathbf{Q}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\mathbf{Q}_2 \\ & +\frac{k_2\gamma_d}{2}\mathbf{D}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\mathbf{D}_2 + \frac{\gamma_d\beta}{2k_1}\mathbf{D}_2^T(\frac{\partial^2 f}{\partial\mathbf{Y}\partial\mathbf{Y}^T})^{-1}\mathbf{D}_2 \\ & +\frac{k_1\gamma_d\beta}{2}\mathbf{W}_2^T(\frac{\partial^2 f}{\partial\mathbf{Y}\partial\mathbf{Y}^T})^{-1}\mathbf{W}_2\end{aligned} \tag{4.31}$$

We then collect terms to obtain the following inequality:

$$\begin{aligned}\dot{V}_{b1} \leq & -(\alpha\gamma_d\lambda_1 - \frac{\alpha\gamma_d\lambda_2}{2k_3} - \frac{\beta\gamma_d k_1}{2M})\mathbf{W}_2^T\mathbf{W}_2 - (\beta\gamma_d\lambda_1 - \frac{\gamma_d\lambda_2 k_2}{2} - \frac{\beta\gamma_d}{2k_1 M})\mathbf{D}_2^T\mathbf{D}_2 \\ & -(\gamma_d\lambda_1 - \frac{\gamma_d\lambda_2}{2k_2})\mathbf{Q}_2^T\mathbf{Q}_2 + \frac{\alpha\gamma_d\lambda_2 k_3}{2}\mathbf{K}_2^T\mathbf{K}_2\end{aligned} \tag{4.32}$$

By setting $k_1 = \frac{1}{\lambda_1 M}, k_2 = \frac{\lambda_2}{\lambda_1}, k_3 = \frac{\lambda_2}{\lambda_1}, \alpha = \frac{\lambda_2^2}{\lambda_1^4 M^2}$ and $\beta = \frac{\lambda_2^2}{\lambda_1^2}$, we obtain:

$$\dot{V}_{b1} \leq -\frac{\gamma_d\lambda_2^2}{2\lambda_1^3 M^2}\mathbf{W}_2^T\mathbf{W}_2 - \frac{3\gamma_d\lambda_2^2}{2\lambda_1}\mathbf{D}_2^T\mathbf{D}_2 - \frac{\gamma_d\lambda_1}{2}\mathbf{Q}_2^T\mathbf{Q}_2 + \frac{\gamma_d\lambda_2^4}{2\lambda_1^5 M^2}\mathbf{K}_2^T\mathbf{K}_2. \tag{4.33}$$

From equations (4.29) and (4.33), the dual system has an asymptotically stable in absence of external inputs, $\mathbf{K}_2$. Since we assume that the local Lagrange multiplier vector $\boldsymbol{\lambda}$ is at the critical point then $\tilde{\boldsymbol{\lambda}} = \mathbf{0}$ and therefore, $\mathbf{K}_2 = \mathbf{0}$. We can then conclude that the boundary layer has an exponentially stable equilibrium at the origin.

Next we consider the reduced order system. The reduced order system as-

sumes that the fast dynamics are at the equilibrium. This leads to the following dynamics:

$$\frac{d\boldsymbol{\mu}}{d\tau} = \epsilon\rho_d(\mathbf{q} + \hat{\mathbf{q}}) \tag{4.34}$$

where $\mathbf{q}$ is a Newton step for the consensus problem. The Newton step is applied to the boundary layer system, in deviation form, to create the following dynamics

$$\frac{d\tilde{\mathbf{y}}}{d\tau} = \rho_p\tilde{\mathbf{v}} + \epsilon\rho_d\mathcal{M}(\mathbf{q} + \hat{\mathbf{q}}) \tag{4.35}$$

$$\frac{d\mathbf{v}}{d\tau} = -\gamma_p((\nabla^2_{yy}L(\mathbf{y}, \boldsymbol{\lambda}, \boldsymbol{\mu}))\mathbf{v} + \nabla_y L(\mathbf{y}, \boldsymbol{\lambda}, \boldsymbol{\mu})) \tag{4.36}$$

$$\frac{d\tilde{\boldsymbol{\lambda}}}{d\tau} = (\tilde{\mathbf{y}} - \frac{R}{m}) + \mathcal{M}\tilde{\boldsymbol{\mu}} - \epsilon\rho_d\mathcal{M}(\nabla^2 f(\mathbf{y}))(\mathbf{q} + \hat{\mathbf{q}}) \tag{4.37}$$

$$\frac{d\tilde{\mathbf{w}}}{d\tau} = -\gamma_d(\mathcal{M}\tilde{\mathbf{w}} - \mathcal{M}\tilde{\boldsymbol{\lambda}}) - \epsilon\rho_d\mathcal{M}(\nabla^2 f(\mathbf{y}))(\mathbf{q} + \hat{\mathbf{q}}) \tag{4.38}$$

$$\frac{d\tilde{\mathbf{d}}}{d\tau} = -\gamma_d(\mathcal{M}\tilde{\mathbf{d}} + (\nabla^2 f(\mathbf{y}))^{-1}\tilde{\mathbf{w}}) + \epsilon\rho_d\mathcal{M}(\mathbf{q} + \hat{\mathbf{q}}) \tag{4.39}$$

$$\frac{d\tilde{\mathbf{q}}}{d\tau} = -\gamma_d(\mathcal{M}\tilde{\mathbf{q}} - \mathcal{M}\tilde{\mathbf{d}}) + \epsilon\rho_d\mathcal{M}(\mathbf{q} + \hat{\mathbf{q}}) \tag{4.40}$$

$$\frac{d\boldsymbol{\mu}}{d\tau} = \epsilon\rho_d\mathbf{q} + \epsilon\rho_d\hat{\mathbf{q}}. \tag{4.41}$$

To show that the reduced order system (4.35)-(4.41) is asymptotically stable, we first change coordinate systems using the nullspace of the Laplacian. In the

new coordinate system, the dynamics of $\mathbf{Y}$ are expressed as

$$\dot{\mathbf{Y}} = \rho_p \mathbf{V} + \epsilon \rho_d \begin{bmatrix} \mathbf{r}^T \mathcal{L} \mathbf{r} \otimes I_n & \mathbf{r}^T \mathcal{L} \mathbf{R} \otimes I_n \\ \mathbf{R}^T \mathcal{L} \mathbf{r} \otimes I_n & \mathbf{R}^T \mathcal{L} \mathbf{R} \otimes I_n \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 + \hat{\mathbf{Q}}_1 \\ \mathbf{Q}_2 + \hat{\mathbf{Q}}_2 \end{bmatrix}$$

The dynamics of $\mathbf{Y}$ are written as

$$\begin{aligned} \dot{\mathbf{Y}}_1 &= \rho_p \mathbf{V}_1 \\ \dot{\mathbf{Y}}_2 &= \rho_p \mathbf{V}_2 + \epsilon \rho_d (\mathbf{R}^T \mathcal{L} \mathbf{R} \otimes I_n)(\mathbf{Q}_2 + \hat{\mathbf{Q}}_2) \end{aligned}$$

The dynamics of $\mathbf{V}$ are unchanged in the new coordinate system. Furthermore, because $\tilde{\mathbf{v}} = \mathbf{v}$, we state the dynamics as

$$\begin{aligned} \dot{\mathbf{V}}_1 &= -\gamma_p ((\nabla^2_{yy} L(\mathbf{y}, \boldsymbol{\lambda}, \boldsymbol{\mu})) \mathbf{V}_1 + \mathbf{g}_1) \\ \dot{\mathbf{V}}_2 &= -\gamma_p ((\nabla^2_{yy} L(\mathbf{y}, \boldsymbol{\lambda}, \boldsymbol{\mu})) \mathbf{V}_2 + \mathbf{g}_2) \end{aligned}$$

In the new coordinate system, the dynamics of $\boldsymbol{\lambda}$ are expressed as

$$\begin{aligned} \dot{\mathbf{K}} =& \mathbf{Y} - ([\mathbf{r}, \mathbf{R}] \otimes I_n) \frac{R}{m} + \begin{bmatrix} \mathbf{r}^T \mathcal{L} \mathbf{r} \otimes I_n & \mathbf{r}^T \mathcal{L} \mathbf{R} \otimes I_n \\ \mathbf{R}^T \mathcal{L} \mathbf{r} \otimes I_n & \mathbf{R}^T \mathcal{L} \mathbf{R} \otimes I_n \end{bmatrix} \begin{bmatrix} \boldsymbol{\Pi}_1 \\ \boldsymbol{\Pi}_2 \end{bmatrix} \\ & - \epsilon \rho_d \left(\frac{\partial^2 f}{\partial \mathbf{Y} \partial \mathbf{Y}^T} \right) \begin{bmatrix} \mathbf{r}^T \mathcal{L} \mathbf{r} \otimes I_n & \mathbf{r}^T \mathcal{L} \mathbf{R} \otimes I_n \\ \mathbf{R}^T \mathcal{L} \mathbf{r} \otimes I_n & \mathbf{R}^T \mathcal{L} \mathbf{R} \otimes I_n \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 + \hat{\mathbf{Q}}_2 \\ \mathbf{Q}_2 + \hat{\mathbf{Q}}_2 \end{bmatrix} \end{aligned}$$

where $\dot{\mathbf{K}}_1, \dot{\mathbf{K}}_2$ are defined as

$$\dot{\mathbf{K}}_1 = \mathbf{Y}_1 - ([\mathbf{r}, \mathbf{R}] \otimes I_n)\frac{R}{m}$$
$$\dot{\mathbf{K}}_2 = \mathbf{Y}_2 - ([\mathbf{r}, \mathbf{R}] \otimes I_n)\frac{R}{m} + (\mathbf{R}^T\mathcal{L}\mathbf{R} \otimes I_n)\mathbf{\Pi}_2 - \epsilon\rho_d \left(\frac{\partial^2 f}{\partial \mathbf{Y}_2 \partial \mathbf{Y}_2^T}\right)(\mathbf{R}^T\mathcal{L}\mathbf{R} \otimes I_n)(\mathbf{Q}_2 + \hat{\mathbf{Q}}_2)$$

The dynamics of $\mathbf{W}$ are:

$$\begin{aligned}\dot{\mathbf{W}} = &- \gamma_d \begin{bmatrix} \mathbf{r}^T\mathcal{L}\mathbf{r} \otimes I_n & \mathbf{r}^T\mathcal{L}\mathbf{R} \otimes I_n \\ \mathbf{R}^T\mathcal{L}\mathbf{r} \otimes I_n & \mathbf{R}^T\mathcal{L}\mathbf{R} \otimes I_n \end{bmatrix} \begin{bmatrix} \mathbf{W}_1 - \mathbf{K}_1 \\ \mathbf{W}_2 - \mathbf{K}_2 \end{bmatrix} \\ &- \epsilon\rho_d \left(\frac{\partial^2 f}{\partial \mathbf{Y} \partial \mathbf{Y}^T}\right) \begin{bmatrix} \mathbf{r}^T\mathcal{L}\mathbf{r} \otimes I_n & \mathbf{r}^T\mathcal{L}\mathbf{R} \otimes I_n \\ \mathbf{R}^T\mathcal{L}\mathbf{r} \otimes I_n & \mathbf{R}^T\mathcal{L}\mathbf{R} \otimes I_n \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 + \hat{\mathbf{Q}}_1 \\ \mathbf{Q}_2 + \hat{\mathbf{Q}}_2 \end{bmatrix}\end{aligned}$$

Similar to the previous dynamics, one can write $\mathbf{W}$ as:

$$\dot{\mathbf{W}}_1 = 0$$
$$\dot{\mathbf{W}}_2 = -\gamma_d(\mathbf{R}^T\mathcal{L}\mathbf{R} \otimes I_n)(\mathbf{W}_2 - \mathbf{K}_2) - \epsilon\rho_d \left(\frac{\partial^2 f}{\partial \mathbf{Y}_2 \partial \mathbf{Y}_2^T}\right)(\mathbf{R}^T\mathcal{L}\mathbf{R} \otimes I_n)(\mathbf{Q}_2 + \hat{\mathbf{Q}}_2)$$

The dynamics of $\mathbf{D}$ in equation (4.39) are expressed as

$$\begin{aligned}\dot{\mathbf{D}} = &- \gamma_d \left(\begin{bmatrix} \mathbf{r}^T\mathcal{L}\mathbf{r} \otimes I_n & \mathbf{r}^T\mathcal{L}\mathbf{R} \otimes I_n \\ \mathbf{R}^T\mathcal{L}\mathbf{r} \otimes I_n & \mathbf{R}^T\mathcal{L}\mathbf{R} \otimes I_n \end{bmatrix} \begin{bmatrix} \mathbf{D}_1 \\ \mathbf{D}_2 \end{bmatrix} + \left(\frac{\partial^2 f}{\partial \mathbf{Y} \partial \mathbf{Y}^T}\right)^{-1} \begin{bmatrix} \mathbf{W}_1^T & \mathbf{W}_2^T \end{bmatrix} \right) \\ &+ \epsilon\rho_d \begin{bmatrix} \mathbf{r}^T\mathcal{L}\mathbf{r} \otimes I_n & \mathbf{r}^T\mathcal{L}\mathbf{R} \otimes I_n \\ \mathbf{R}^T\mathcal{L}\mathbf{r} \otimes I_n & \mathbf{R}^T\mathcal{L}\mathbf{R} \otimes I_n \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 + \hat{\mathbf{Q}}_1 \\ \mathbf{Q}_2 + \hat{\mathbf{Q}}_2 \end{bmatrix}.\end{aligned}$$

The dynamics are then written as:

$$\begin{aligned}
\dot{\mathbf{D}}_1 &= -\gamma_d(\mathbf{r}^T \otimes I_n)(\nabla^2 f(\mathbf{y}))^{-1}([\mathbf{r}\ \mathbf{R}] \otimes I_n)\mathbf{W}_2 \\
\dot{\mathbf{D}}_2 &= -\gamma_d(\mathbf{R}^T\mathcal{L}\mathbf{R} \otimes I_n)\mathbf{D}_2 - \gamma_d(\mathbf{R}^T \otimes I_n)(\nabla^2 f(\mathbf{y}))^{-1}([\mathbf{r}\ \mathbf{R}] \otimes I_n)\mathbf{W}_2 \\
&\quad + \epsilon\rho_d(\mathbf{R}^T\mathcal{L}\mathbf{R} \otimes I_n)(\mathbf{Q}_2 + \hat{\mathbf{Q}}_2).
\end{aligned}$$

The dynamics of $\mathbf{Q}$ becomes:

$$\begin{aligned}
\dot{\mathbf{Q}} = &-\gamma_d \begin{bmatrix} \mathbf{r}^T\mathcal{L}\mathbf{r} \otimes I_n & \mathbf{r}^T\mathcal{L}\mathbf{R} \otimes I_n \\ \mathbf{R}^T\mathcal{L}\mathbf{r} \otimes I_n & \mathbf{R}^T\mathcal{L}\mathbf{R} \otimes I_n \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 - \mathbf{D}_1 \\ \mathbf{Q}_2 - \mathbf{D}_2 \end{bmatrix} \\
&+ \epsilon\rho_d \begin{bmatrix} \mathbf{r}^T\mathcal{L}\mathbf{r} \otimes I_n & \mathbf{r}^T\mathcal{L}\mathbf{R} \otimes I_n \\ \mathbf{R}^T\mathcal{L}\mathbf{r} \otimes I_n & \mathbf{R}^T\mathcal{L}\mathbf{R} \otimes I_n \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 + \hat{\mathbf{Q}}_1 \\ \mathbf{Q}_2 + \hat{\mathbf{Q}}_2 \end{bmatrix}
\end{aligned}$$

It is assumed without loss of generality that $\mathbf{W}_1(0) = 0$, this allows $\mathbf{Q}$ to be expressed as:

$$\begin{aligned}
\dot{\mathbf{Q}}_1 &= 0 \\
\dot{\mathbf{Q}}_2 &= -\gamma_d(\mathbf{R}^T\mathcal{L}\mathbf{R} \otimes I_n)(\mathbf{Q}_2 - \mathbf{D}_2) + \epsilon\rho_d(\mathbf{R}^T\mathcal{L}\mathbf{R} \otimes I_n)(\mathbf{Q}_2 + \hat{\mathbf{Q}}_2).
\end{aligned}$$

Finally, the dynamics for $\mathbf{\Pi}$ are

$$\begin{aligned}
\dot{\mathbf{\Pi}}_1 &= 0 \\
\dot{\mathbf{\Pi}}_2 &= \epsilon\rho_d\mathbf{Q}_2 + \epsilon\rho_d\hat{\mathbf{Q}}_2
\end{aligned}$$

The equilibrium sets formed by the solution of the fast boundary layer dynamics allow us to form the following two relationships. Through the equation (4.24)

$$\mathcal{M}\boldsymbol{\lambda} = \mathcal{M}\mathbf{w} \tag{4.42}$$

$$\mathcal{M}\mathbf{d} = -\nabla^2 f(\mathbf{y})\mathbf{w} \tag{4.43}$$

Therefore, we state

$$-\mathcal{M}\mathbf{d}(\nabla^2 f(\mathbf{y}))^{-1}\mathcal{M} = \mathcal{M}\boldsymbol{\lambda}. \tag{4.44}$$

This allows us to define the following dynamics for $\boldsymbol{\lambda}$ in the new coordinates

$$\dot{\bar{\mathbf{K}}}_2 = -\rho_d(\frac{\partial^2 f}{\partial \mathbf{Y}_2^T \partial \mathbf{Y}_2})^{-1}(\mathbf{R}^T\mathcal{L}\mathbf{R} \otimes I_n)\bar{\mathbf{D}}_2 \tag{4.45}$$

where $\mathbf{D}_2 = \bar{\mathbf{D}}_2 - \mathbf{D}_2^*$. Furthermore, by defining $\mathbf{E}_2 = \mathbf{W}_2 - \mathbf{K}_2$, we create the following dynamics

$$\dot{\bar{\mathbf{E}}}_2 = -\gamma_d((\mathbf{R}^T\mathcal{L}\mathbf{R} \otimes I_n)\bar{\mathbf{E}}_2 - \dot{\bar{\mathbf{K}}}_2 - \epsilon\rho_d(\mathbf{R}^T\mathcal{L}\mathbf{R} \otimes I_n)\frac{\partial^2 f}{\partial \mathbf{Y}_2^T \partial \mathbf{Y}_2}(\mathbf{Q}_2 + \hat{\mathbf{Q}}_2) \tag{4.46}$$

using (4.45), $\dot{\bar{\mathbf{E}}}_2$ can be written as

$$\begin{aligned}\dot{\bar{\mathbf{E}}}_2 =& -\gamma_d((\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\bar{\mathbf{E}}_2 + \rho_d(\frac{\partial^2 f}{\partial \mathbf{Y}_2^T\partial\mathbf{Y}_2})^{-1}(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\bar{\mathbf{D}}_2 \\ &- \epsilon\rho_d(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\frac{\partial^2 f}{\partial \mathbf{Y}_2^T\partial\mathbf{Y}_2}(\mathbf{Q}_2+\hat{\mathbf{Q}}_2)\end{aligned} \tag{4.47}$$

and the dynamics of $\bar{\mathbf{D}}_2$ are stated as

$$\dot{\bar{\mathbf{D}}}_2 = -\gamma_d(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\bar{\mathbf{D}}_2 - \gamma_d\frac{\partial^2 f}{\partial \mathbf{Y}_2^T\partial\mathbf{Y}_2}\bar{\mathbf{E}}_2 + \gamma_d\frac{\partial^2 f}{\partial \mathbf{Y}_2^T\partial\mathbf{Y}_2}\bar{\mathbf{K}}_2 \tag{4.48}$$

$$+ \epsilon\rho_d(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)(\mathbf{Q}_2+\hat{\mathbf{Q}}_2). \tag{4.49}$$

The optimal $\mathbf{D}_2$ is defined as

$$\mathbf{D}_2^* = (\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)^{-1}\frac{\partial^2 f}{\partial \mathbf{Y}_2^T\partial\mathbf{Y}_2}\bar{\mathbf{K}}_2. \tag{4.50}$$

Using (4.50), we redefine the dynamics of $\mathbf{K}_2$ as

$$\dot{\bar{\mathbf{K}}}_2 = -\rho_d(\frac{\partial^2 f}{\partial \mathbf{Y}_2^T\partial\mathbf{Y}_2})^{-1}(R^T\mathcal{L}R\otimes I_n)\mathbf{D}_2 - \rho_d\bar{\mathbf{K}}_2 \tag{4.51}$$

and we redefine the dynamics of $\mathbf{E}_2$

$$\dot{\bar{\mathbf{E}}}_2 = -\gamma_d((\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\bar{\mathbf{E}}_2 + \rho_d(\frac{\partial^2 f}{\partial \mathbf{Y}_2^T\partial\mathbf{Y}_2})^{-1}(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\mathbf{D}_2 + \rho_d\bar{\mathbf{K}}_2. \tag{4.52}$$

Finally, we consider the dynamics of $\mathbf{D}_2$

$$\begin{aligned}\dot{\mathbf{D}}_2 =& -\gamma_d(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\mathbf{D}_2 - \gamma_d\frac{\partial^2 f}{\partial \mathbf{Y}_2^T\partial\mathbf{Y}_2}\bar{\mathbf{E}}_2 - \rho_d\mathbf{D}_2 \\ &- \rho_d(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)^{-1}\frac{\partial^2 f}{\partial \mathbf{Y}_2^T\partial\mathbf{Y}_2}\bar{\mathbf{K}}_2 + \epsilon\rho_d(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)(\mathbf{Q}_2+\hat{\mathbf{Q}}_2)\end{aligned} \tag{4.53}$$

Using these dynamics, we pose the second half of the boundary layer dynamics' Lyapunov candidate function as follows

$$\begin{aligned}V_{b2} =& \frac{1}{2}\mathbf{Y}_2^T\mathbf{Y}_2 + \frac{\psi}{2}\mathbf{Q}_2^T\mathbf{Q}_2 + \frac{\sigma}{2}\mathbf{V}_2^T\mathbf{V}_2 + \frac{1}{2}\mathbf{g}_2^T\mathbf{V}_2 \\ &+\frac{1}{2}\mathbf{g}_2^T\mathbf{g}_2 + \frac{1}{2}\mathbf{E}_2^T\mathbf{E}_2 + \frac{\beta}{2}\mathbf{D}_2^T\mathbf{D}_2 + \frac{\alpha}{2}\bar{\mathbf{K}}_2^T\bar{\mathbf{K}}_2\end{aligned} \tag{4.54}$$

The derivative of the Lyapunov candidate function is

$$
\begin{aligned}
\dot{V}_{b2} =& \rho_p \mathbf{Y}_2^T \mathbf{V}_2 + \epsilon \rho_d \mathbf{Y}_2^T (\mathbf{R}^T \mathcal{L} \mathbf{R} \otimes I_n)(\mathbf{Q}_2 + \hat{\mathbf{Q}}_2) - \psi \gamma_d \mathbf{Q}_2^T (\mathbf{R}^T \mathcal{L} \mathbf{R} \otimes I_n)(\mathbf{Q}_2 - \mathbf{D}_2) \\
&+ \psi \epsilon \rho_d \mathbf{Q}_2^T (\mathbf{R}^T \mathcal{L} \mathbf{R} \otimes I_n)(\mathbf{Q}_2 + \hat{\mathbf{Q}}_2) - \gamma_p \sigma \mathbf{V}_2^T \frac{\partial^2 f}{\partial \mathbf{Y}_2^T \partial \mathbf{Y}_2} \mathbf{V}_2 - \gamma_p \sigma \mathbf{g}_2^T \mathbf{V}_2 + \rho_p \mathbf{g}_2^T \frac{\partial^2 f}{\partial \mathbf{Y}_2^T \partial \mathbf{Y}_2} \mathbf{V}_2 \\
&+ \rho_p \mathbf{V}_2^T \frac{\partial^2 f}{\partial \mathbf{Y}_2^T \partial \mathbf{Y}_2} \mathbf{V}_2 - \gamma_p \mathbf{g}_2^T \frac{\partial^2 f}{\partial \mathbf{Y}_2^T \partial \mathbf{Y}_2} \mathbf{V}_2 - \gamma_p \mathbf{g}_2^T \mathbf{g}_2 - \gamma_d \mathbf{E}_2^T (\mathbf{R}^T \mathcal{L} \mathbf{R} \otimes I_n) \mathbf{E}_2 \\
&+ \rho_d \mathbf{E}_2^T (\frac{\partial^2 f}{\partial \mathbf{Y}_2^T \partial \mathbf{Y}_2})^{-1} (\mathbf{R}^T \mathcal{L} \mathbf{R} \otimes I_n) \mathbf{D}_2 + \rho_d \mathbf{E}_2^T \bar{\mathbf{K}}_2 - \alpha \rho_d \bar{\mathbf{K}}_2^T \bar{\mathbf{K}}_2 \\
&- \beta \gamma_d \mathbf{D}_2^T (\mathbf{R}^T \mathcal{L} \mathbf{R} \otimes I_n) \mathbf{D}_2 - \beta \gamma_d \mathbf{D}_2^T (\frac{\partial^2 f}{\partial \mathbf{Y}_2^T \partial \mathbf{Y}_2}) \mathbf{E}_2 - \beta \rho_d \mathbf{D}_2^T \mathbf{D}_2 \\
&- \beta \rho_d \mathbf{D}_2^T (\mathbf{R}^T \mathcal{L} \mathbf{R} \otimes I_n)^{-1} (\frac{\partial^2 f}{\partial \mathbf{Y}_2^T \partial \mathbf{Y}_2}) \bar{\mathbf{K}}_2 + \alpha \rho \bar{\mathbf{K}}_2^T (\mathbf{R}^T \mathcal{L} \mathbf{R} \otimes I_n) (\frac{\partial^2 f}{\partial \mathbf{Y}_2^T \partial \mathbf{Y}_2})^{-1} \mathbf{D}_2 \\
&- \epsilon \rho_d \mathbf{E}_2^T (\mathbf{R}^T \mathcal{L} \mathbf{R} \otimes I_n) (\frac{\partial^2 f}{\partial \mathbf{Y}_2^T \partial \mathbf{Y}_2}) (\mathbf{Q}_2 + \hat{\mathbf{Q}}_2) + \epsilon \rho_d \mathbf{D}_2^T (\mathbf{R}^T \mathcal{L} \mathbf{R} \otimes I_n)(\mathbf{Q}_2 + \hat{\mathbf{Q}}_2) \\
&+ \epsilon \rho_d \psi \mathbf{Q}_2^T (\mathbf{R}^T \mathcal{L} \mathbf{R} \otimes I_n)(\mathbf{Q}_2 + \hat{\mathbf{Q}}_2)
\end{aligned}
\tag{4.55}
$$

One can create an upper bound such that $\mathbf{Y}_2^T \mathbf{Y}_2 \leq \xi \mathbf{g}_2^T \mathbf{g}_2$, for some positive con-

stant ξ, allowing us to create the following inequality:

$$\begin{aligned}
\dot{V}_{b2} \leq & \frac{\rho_p \xi}{2k_1}\mathbf{g}_2^T\mathbf{g}_2 + \frac{\rho_p k_1}{2}\mathbf{V}_2^T\mathbf{V}_2 + \frac{\xi\epsilon\rho_d}{2k_2}\mathbf{g}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\mathbf{g}_2 + \frac{\epsilon\rho_d k_2}{2}\mathbf{Q}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\mathbf{Q}_2 \\
& - \psi\gamma_d\mathbf{Q}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\mathbf{Q}_2 + \frac{\psi\gamma_d}{2k_3}\mathbf{Q}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\mathbf{Q}_2 + \frac{\psi\gamma_d k_3}{2}\mathbf{D}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\mathbf{D}_2 \\
& + \psi\epsilon\rho_d\mathbf{Q}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\mathbf{Q}_2 - (\gamma_p\sigma-\rho_p)\mathbf{V}_2^T\frac{\partial^2 f}{\partial\mathbf{Y}_2^T\partial\mathbf{Y}_2}\mathbf{V}_2 + \frac{\gamma_p\sigma k_{16}}{2}\mathbf{V}_2^T\mathbf{V}_2 \\
& + \frac{\gamma_p\sigma}{2k_{16}}\mathbf{g}_2^T\mathbf{g}_2 - \frac{1}{2k_{17}}(\gamma_p\sigma-\rho_p)\mathbf{g}_2^T\frac{\partial^2 f}{\partial\mathbf{Y}_2^T\partial\mathbf{Y}_2}\mathbf{g}_2 - \frac{k_{17}}{2}(\gamma_p\sigma-\rho_p)\mathbf{V}_2^T\frac{\partial^2 f}{\partial\mathbf{Y}_2^T\partial\mathbf{Y}_2}\mathbf{V}_2 \\
& - \gamma_p\mathbf{g}_2^T\mathbf{g}_2 - \gamma_d\mathbf{E}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\mathbf{E}_2 + \frac{\rho}{2k_4}\mathbf{E}_2^T(\frac{\partial^2 f}{\partial\mathbf{Y}_2^T\partial\mathbf{Y}_2})^{-1}(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\mathbf{E}_2 \\
& + \frac{\rho k_4}{2}\mathbf{D}_2^T(\frac{\partial^2 f}{\partial\mathbf{Y}_2^T\partial\mathbf{Y}_2})^{-1}(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\mathbf{D}_2 + \frac{\rho}{2k_5}\mathbf{E}_2^T\mathbf{E}_2 + \frac{\rho k_5}{2}\bar{\mathbf{K}}_2^T\bar{\mathbf{K}}_2 - \alpha\rho\bar{\mathbf{K}}_2^T\bar{\mathbf{K}}_2 \\
& - \beta\gamma_d\mathbf{D}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\mathbf{D}_2 + \frac{\beta\gamma_d}{2k_{10}}\mathbf{D}_2^T(\frac{\partial^2 f}{\partial\mathbf{Y}_2^T\partial\mathbf{Y}_2})\mathbf{D}_2 + \frac{\beta\gamma_d k_{10}}{2}\mathbf{E}_2^T(\frac{\partial^2 f}{\partial\mathbf{Y}_2^T\partial\mathbf{Y}_2})\mathbf{E}_2 - \beta\rho\mathbf{D}_2^T\mathbf{D}_2 \\
& + \frac{\beta\rho}{2k_6}\mathbf{D}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)^{-1}(\frac{\partial^2 f}{\partial\mathbf{Y}_2^T\partial\mathbf{Y}_2})\mathbf{D}_2 + \frac{\beta\rho k_6}{2}\bar{\mathbf{K}}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)^{-1}(\frac{\partial^2 f}{\partial\mathbf{Y}_2^T\partial\mathbf{Y}_2})\bar{\mathbf{K}}_2 \\
& + \frac{\alpha\rho}{2k_7}\bar{\mathbf{K}}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)(\frac{\partial^2 f}{\partial\mathbf{Y}_2^T\partial\mathbf{Y}_2})^{-1}\bar{\mathbf{K}}_2 + \frac{\alpha\rho k_7}{2}\mathbf{D}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)(\frac{\partial^2 f}{\partial\mathbf{Y}_2^T\partial\mathbf{Y}_2})^{-1}\mathbf{D}_2 \\
& + \frac{\epsilon\rho_d}{2k_8}\mathbf{E}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)(\frac{\partial^2 f}{\partial\mathbf{Y}_2^T\partial\mathbf{Y}_2})\mathbf{E}_2 + \frac{\epsilon\rho_d k_8}{2}\mathbf{Q}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)(\frac{\partial^2 f}{\partial\mathbf{Y}_2^T\partial\mathbf{Y}_2})\mathbf{Q}_2 \\
& + \frac{\epsilon\rho_d}{2k_9}\mathbf{D}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\mathbf{D}_2 + \frac{\epsilon\rho_d k_9}{2}\mathbf{Q}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\mathbf{Q}_2 \\
& + \epsilon\rho_d\psi\mathbf{Q}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\mathbf{Q}_2 + \frac{\epsilon\rho_d}{2k_{15}}\mathbf{g}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\mathbf{g}_2 + \frac{\epsilon\rho_d k_{15}}{2}\hat{\mathbf{Q}}_2(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\hat{\mathbf{Q}}_2 \\
& + \frac{\psi\epsilon\rho_d}{2k_{11}}\mathbf{Q}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\mathbf{Q}_2 + \frac{\psi\epsilon\rho_d k_{11}}{2}\hat{\mathbf{Q}}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\hat{\mathbf{Q}}_2 \\
& + \frac{\epsilon\rho_d}{2k_{12}}\mathbf{E}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)(\frac{\partial^2 f}{\partial\mathbf{Y}_2^T\partial\mathbf{Y}_2})\mathbf{E}_2 + \frac{\epsilon\rho_d k_{12}}{2}\hat{\mathbf{Q}}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)(\frac{\partial^2 f}{\partial\mathbf{Y}_2^T\partial\mathbf{Y}_2})\hat{\mathbf{Q}}_2 \\
& + \frac{\epsilon\rho_d}{2k_{13}}\mathbf{D}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\mathbf{D}_2 + \frac{\epsilon\rho_d k_{13}}{2}\hat{\mathbf{Q}}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\hat{\mathbf{Q}}_2 \\
& + \frac{\epsilon\rho_d\psi}{2k_{14}}\mathbf{Q}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\mathbf{Q}_2 + \frac{\epsilon\rho_d\psi k_{14}}{2}\hat{\mathbf{Q}}_2^T(\mathbf{R}^T\mathcal{L}\mathbf{R}\otimes I_n)\hat{\mathbf{Q}}_2
\end{aligned} \tag{4.56}$$

After collecting terms, we obtain the inequality:

$$\begin{aligned}
\dot{V}_{b2} \leq & -(\gamma_p - \frac{\gamma_p \sigma}{2k_{16}} - \frac{\gamma_p M}{4k_{17}} - \frac{\rho_p \xi}{2k_1} - \frac{\epsilon \rho_d \lambda_2 \xi}{2k_2} - \frac{\epsilon \rho_d \lambda_2}{2k_{15}}) \mathbf{g}_2^T \mathbf{g}_2 \\
& -(\gamma_p \sigma M - \frac{\gamma_p \sigma k_{16}}{2} - \frac{k_{17} M}{4} - \rho_p M - \frac{k_1 \rho_p}{2}) \mathbf{V}_2^T \mathbf{V}_2 \\
& -(\gamma_d \lambda_1 - \frac{\rho \lambda_2}{2k_4 M} - \frac{\rho}{2k_5} - \frac{\beta \gamma_d M k_{10}}{2} - \frac{\epsilon \rho_d M \lambda_2}{2k_8} - \frac{\epsilon \rho_d M \lambda_2}{2k_{12}}) \mathbf{E}_2^T \mathbf{E}_2 \\
& -(\beta \lambda_1 \gamma_d + \beta \rho - \frac{\rho \lambda_2 k_4}{2M} - \frac{\beta \gamma_d M}{2k_{10}} - \frac{\rho \beta M}{2\lambda_2 k_6} - \frac{\alpha \rho \lambda_2 k_7}{2M} + \frac{\epsilon \rho_d \lambda_2}{2k_9} - \frac{\epsilon \rho_d \lambda_2}{2k_{13}} - \frac{\psi \gamma_d \lambda_2 k_3}{2}) \mathbf{D}_2^T \mathbf{D}_2 \\
& -(\rho \alpha - \frac{\rho k_5}{2} - \frac{\rho \beta k_6 M}{2\lambda_2} - \frac{\alpha \rho \lambda_2}{2M k_7}) \mathbf{K}_2^T \mathbf{K}_2 \\
& -(\psi \gamma_d \lambda_1 - \frac{\psi \gamma_d \lambda_2}{2k_3} - \frac{\epsilon \rho_d \psi \lambda_2}{2k_{14}} - \frac{\epsilon \rho_d \psi \lambda_2}{2k_{11}}) \mathbf{Q}_2^T \mathbf{Q}_2 \\
& +(\frac{\epsilon \rho_d \lambda_2 \xi k_2}{2} + \frac{\epsilon \rho_d M \lambda_2 k_8}{2} + \frac{\epsilon \rho_d \lambda_2 k_9}{2} + \epsilon \psi \rho_d \lambda_2) \mathbf{Q}_2^T \mathbf{Q}_2 \\
& +(\frac{\epsilon \rho_d k_{15} \lambda_2}{2} + \frac{\psi \epsilon \rho_d k_{11} \lambda_2}{2} + \frac{\epsilon \rho_d k_{12} \lambda_2 M}{2} + \frac{\epsilon \rho_d k_{13} \lambda_2}{2} + \frac{\epsilon \rho_d \psi k_{14} \lambda_2}{2}) \hat{\mathbf{Q}}_2^T \hat{\mathbf{Q}}_2
\end{aligned} \tag{4.57}$$

By setting $k_1 = M, k_2 = \frac{16\epsilon\rho_d\lambda_2}{\gamma_p}, k_3 = \frac{\lambda_2}{\lambda_1}, k_4 = \frac{\beta M}{8\lambda_2}, k_5 = \frac{\alpha}{4}, k_6 = \frac{M}{\lambda_2}, k_7 = \frac{\lambda_2}{M}, k_8 = \frac{4\epsilon\rho_d M\lambda_2}{\gamma_d\lambda_1}, k_9 = \frac{8\epsilon\rho_d\lambda_2}{\rho\beta}, k_{10} = \frac{M}{\lambda_1}, k_{11} = \frac{4\epsilon\rho_d\lambda_2}{\gamma_d\lambda_1}, k_{12} = \frac{4\epsilon\rho_d M\lambda_2}{\gamma_d\lambda_1}, k_{13} = \frac{2\epsilon\rho_d\lambda_2}{\lambda_1\beta\gamma_d}, k_{14} = \frac{4\epsilon\rho_d\lambda_2}{\gamma_d\lambda_1}, k_{15} = \frac{4\epsilon\rho_d}{\lambda_2\gamma_p}\alpha = \frac{\lambda_1^2}{2\lambda_2^2}, k_{16} = M, k_{17} = M, \sigma = M, \beta = \frac{\lambda_1^2}{8M^2}, \psi = \frac{\lambda_1^6}{8\lambda_2^4M^2}, \rho = \frac{\lambda_1^3\gamma_d}{8\lambda_2^2}, \xi = 1$, and $\rho_p = \frac{\gamma_p M}{8}$, we obtain the gradient of the

Lyapunov candidate function:

$$\begin{aligned}\dot{V}_{b2} \leq & -(\frac{1\gamma_p}{32})\mathbf{g}_2^T\mathbf{g}_2 - (\frac{1\gamma_p M^2}{16})\mathbf{V}_2^T\mathbf{V}_2 - (\frac{\gamma_d\lambda_1}{16})\mathbf{E}_2^T\mathbf{E}_2 \\ & -(\frac{\beta\lambda_1\gamma_d}{4} + \frac{\rho\beta}{16})\mathbf{D}_2^T\mathbf{D}_2 - (\frac{\rho\alpha}{4})\mathbf{K}_2^T\mathbf{K}_2 - (\frac{3\psi\gamma_d\lambda_1}{4})\mathbf{Q}_2^T\mathbf{Q}_2 \\ & +(\frac{8\epsilon^2\rho_d^2\lambda_2^2}{\gamma_p} + \frac{2\epsilon^2\rho_d^2\lambda_2^2M^2}{\gamma_d\lambda_1} + \frac{4\epsilon^2\rho_d^2\lambda_2^2}{\beta\rho} + \epsilon\psi\rho_d\lambda_2)\mathbf{Q}_2^T\mathbf{Q}_2 \\ & +(2\epsilon^2\rho_d^2 + \frac{2\psi\epsilon^2\rho_d^2\lambda_2^2}{\gamma_d\lambda_1} + \frac{2\epsilon^2\rho_d^2\lambda_2^2M^2}{\gamma_d\lambda_1} + \frac{\epsilon^2\rho_d^2\lambda_2^2}{2\lambda_1\beta\gamma_d} + \frac{2\epsilon^2\rho_d^2\psi\lambda_2^2}{\lambda_1\gamma_d})\hat{\mathbf{Q}}_2^T\hat{\mathbf{Q}}_2\end{aligned} \tag{4.58}$$

where the last two terms defines the interconnection between the reduced and boundary layer dynamics. The second last term can be upper bounded by some positive constant θ as follows

$$(\frac{8\epsilon^2\rho_d^2\lambda_2^2}{\gamma_p} + \frac{2\epsilon^2\rho_d^2\lambda_2^2M^2}{\gamma_d\lambda_1} + \frac{4\epsilon^2\rho_d^2\lambda_2^2}{\beta\rho} + \epsilon\psi\rho_d\lambda_2)\mathbf{Q}_2^T\mathbf{Q}_2 \leq \epsilon^2\theta\mathbf{Q}_2^T\mathbf{Q}_2.$$

We see that there exists a positive ϵ that makes the term $\mathbf{Q}_2^T\mathbf{Q}_2$ term negative definite

$$\epsilon \leq (\frac{\psi\gamma_d\lambda_1}{\theta})^{1/2} = \epsilon_1 \tag{4.59}$$

The last positive term only appears when the boundary layer and reduced systems are combined. We can upper bound this term by $\mathbf{\Pi}_2$ as follows

$$(2\epsilon^2\rho_d^2 + \frac{2\psi\epsilon^2\rho_d^2\lambda_2^2}{\gamma_d\lambda_1} + \frac{2\epsilon^2\rho_d^2\lambda_2^2M^2}{\gamma_d\lambda_1} + \frac{\epsilon^2\rho_d^2\lambda_2^2}{2\lambda_1\beta\gamma_d} + \frac{2\epsilon^2\rho_d^2\psi\lambda_2^2}{\lambda_1\gamma_d})\hat{\mathbf{Q}}_2^T\hat{\mathbf{Q}}_2 \leq \epsilon^2\sigma\mathbf{\Pi}_2^T\mathbf{\Pi}_2$$

Finally, we pose the Lyapunov candidate function for the Newton step dynamics

as

$$V_r = \frac{1}{2}\mathbf{\Pi}_2^T\mathbf{\Pi}_2. \tag{4.60}$$

The gradient of the reduced system's Lyapunov function candidate becomes

$$\dot{V}_r = \epsilon\rho_d\mathbf{\Pi}_2^T\mathbf{Q}_2 + \epsilon\rho_d\mathbf{\Pi}_2^T\hat{\mathbf{Q}}_2 \tag{4.61}$$

where $\mathbf{Q}_2$ is defined as the Newton's step of the dual problem. Furthermore, $\hat{\mathbf{Q}}_2$ can be upper bound as follows $\|\hat{\mathbf{\Pi}}_2^T\hat{\mathbf{Q}}_2\| < -\omega\|\mathbf{\Pi}_2^T\mathbf{\Pi}_2\|$. This yields the following inequality

$$\dot{V}_r \leq -(\epsilon\rho_d\omega - \frac{\epsilon\rho_d}{2k_{18}})\mathbf{\Pi}_2^T\mathbf{\Pi}_2 + \frac{\epsilon\rho_d k_{18}}{2}\mathbf{Q}_2^T\mathbf{Q}_2 \tag{4.62}$$

By setting $k_{18} = \frac{1}{\omega}$, we obtain

$$\dot{V}_r \leq -\frac{\epsilon\rho_d\omega}{2}\mathbf{\Pi}_2^T\mathbf{\Pi}_2 + \frac{\epsilon\rho_d}{2\omega}\mathbf{Q}_2^T\mathbf{Q}_2 \tag{4.63}$$

where the reduced system is asymptotically stable when the external input $\mathbf{Q}_2$ is omitted. Furthermore, as $\mathbf{Q}_2 = \mathbf{0}$ is an equilibrium point, we can state that the reduced system is exponentially stable.

We conclude the stability analysis by posing a composite Lyapunov candidate function for the interconnected system formed from the boundary layer and

reduced order systems.

$$V_c = (1-d)(V_{b1} + V_{b2}) + dV_r, \quad 0 < d < 1 \tag{4.64}$$

The rate of change of the composite Lyapunov candidate function is then

$$\dot{V}_c = (1-d)(\dot{V}_{b1} + \dot{V}_{b2}) + d\dot{V}_r, \quad 0 < d < 1 \tag{4.65}$$

We use Lyapunov candidate functions (4.29), (4.58) and (4.60)

$$\begin{aligned}\dot{V}_c \leq & (1-d)(-\frac{\gamma_d \lambda_2^2}{2\lambda_1^3 M^2}\mathbf{W}_2^T\mathbf{W}_2 - \frac{3\psi\gamma_d\lambda_1}{2}\mathbf{Q}_2^T\mathbf{Q}_2 \\ & -\frac{3\gamma_p}{4}\mathbf{g}_2^T\mathbf{g}_2 - \frac{3\gamma_p M}{4}\mathbf{V}_2^T\mathbf{V}_2 - \gamma_d\lambda_1\mathbf{E}_2^T\mathbf{E}_2 \\ & -(\beta\lambda_1\gamma_d + \rho\beta + \frac{3\gamma_d\lambda_2^2}{2\lambda_1})\mathbf{D}_2^T\mathbf{D}_2 - \rho\alpha\mathbf{K}_2^T\mathbf{K}_2 - \gamma_p\mathbf{g}_2^T\mathbf{V}_2 \\ & + \epsilon^2\sigma\mathbf{\Pi}_2^T\mathbf{\Pi}_2) + d(-\frac{\epsilon\omega}{2}\mathbf{\Pi}_2^T\mathbf{\Pi}_2 + \frac{\epsilon\rho_d}{2\omega}\mathbf{Q}_2^T\mathbf{Q}_2)\end{aligned} \tag{4.66}$$

To approximate ϵ_2 we create the following inequality to ensure $\mathbf{\Pi}_2^T\mathbf{\Pi}_2$ is negative definite:

$$\epsilon < \left(\frac{1-d}{d}\right)\frac{\omega}{2\sigma} \leq \epsilon_2, \quad 0 < d < 1. \tag{4.67}$$

Furthermore, we approximate ϵ_3 through the following inequality

$$\epsilon < \frac{(1-d)}{d}\frac{3\psi\gamma_d\lambda_1\omega}{\rho_d} \leq \epsilon_3, \quad 0 < d < 1 \tag{4.68}$$

From 4.67 and 4.68, it can be shown that our approximations of ϵ_2 and ϵ_3 can be

arbitrarily large as d approaches 0. Finally, we obtain an estimate of ϵ^* as

$$\epsilon^* = \min_{\epsilon^* \in \mathcal{R}} \{\epsilon_1, \epsilon_2, \epsilon_3\}. \tag{4.69}$$

The composite Lyapunov candidate function is such that

$$\begin{aligned} \dot{V}_c \leq & (1-d)(-\frac{\gamma_d \lambda_2^2}{2\lambda_1^3 M^2}\mathbf{W}_2^T\mathbf{W}_2 - \frac{3\psi\gamma_d\lambda_1}{2}\mathbf{Q}_2^T\mathbf{Q}_2 \\ & -\frac{1\gamma_p}{32}\mathbf{g}_2^T\mathbf{g}_2 - \frac{1\gamma_p M^2}{16}\mathbf{V}_2^T\mathbf{V}_2 - \gamma_d\lambda_1\mathbf{E}_2^T\mathbf{E}_2 \\ & -(\beta\lambda_1\gamma_d + \rho\beta + \frac{3\gamma_d\lambda_2^2}{2\lambda_1})\mathbf{D}_2^T\mathbf{D}_2 - \rho\alpha\mathbf{K}_2^T\mathbf{K}_2) \\ & - d\frac{\epsilon^*\omega}{2}\mathbf{\Pi}_2^T\mathbf{\Pi}_2 \end{aligned} \tag{4.70}$$

Therefore, there exists a constant $0 < \epsilon < \epsilon^*$ such that the trajectories of the Lagrange multipliers, $\boldsymbol{\lambda}$, will achieve consensus at the unique optimum of the optimization problem. Through the composite Lyapunov candidate function, it is shown that the optimum is an exponentially stable equilibrium point of the system. This completes the stability analysis

Stability of the Discretized Dynamics

The results of this stability analysis show that in continuous time, the dynamics of the distributed resource allocation methodology are exponentially stable at the optimal solution of Lagrangian (3.7). To prove asymptotic convergence of the forward-Euler discretization, we follow the stability analysis outlined in [89] using the following theorem.

Theorem 2: Let the continuous-time system

$$\dot{x} = \phi(x) \tag{4.71}$$

have an equilibrium at the origin. Assume that there exists a suitable Lyapunov function $V(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ for (4.71) that satisfies the following inequalities for a_1, a_2, a_3, a_4.

$$a_1\mathcal{I} \leq \nabla^2 V(x) \leq a_2\mathcal{I} \tag{4.72}$$

$$\frac{\partial V(x)}{\partial x}\phi(x) \leq -a_3\|x\|^2 \tag{4.73}$$

$$\|\phi(x)\| \leq a_4\|x\| \tag{4.74}$$

for all $x \in \mathbb{R}^n$.Then the origin of system (4.71) is globally exponentially stable. Moreover, for the following forward-Euler discretization of the system

$$x(k+1) = x(k) + \varepsilon\phi(x(k)) \tag{4.75}$$

there exists a positive scalar $\bar{\varepsilon}$ such that for every $\varepsilon \in (0, \bar{\varepsilon})$ the discrete time system has an exponentially stable equilibrium at the origin.

4.3 Simulation Results

The solution strategy in Section 4.1 is proposed to be an initialization free strategy. The collection vectors and Lagrange multipliers initialized in each simulation

are randomly assigned values between 1 and 5. The resulting dynamics are evaluated using the simulation examples in Section A with known and unknown cost functions.

4.3.1 3-agent system

To achieve the dynamics shown in Figure 4.1, the following dynamic gains were used.

Table 4.1: Dynamic gains and step-sizes for the known 3-node example using the Distributed Newton Resource Allocation Strategy

Dynamic gain and step-size	ρ_p	γ_p	$\rho_{d,fast}$	$\gamma_{d,fast}$	$\gamma_{d,slow}$
Value	$\frac{1}{5}$	$\frac{3}{10}$	$\frac{15}{100}$	$\frac{15}{100}$	$\frac{1}{5}$

By setting the dynamic gains in this manner, we are able to achieve a time-scale separation between the primal and dual problems. Furthermore, to achieve a faster convergence rate, the dual problem was separately iterated 3 times for each solution of the primal problem. The dual problem's Euler-discretization gain was set to $k = 0.6$ to ensure convergence of the dual solution. The tuning parameters were chosen to minimize the error between the total resource and the allocated resource.

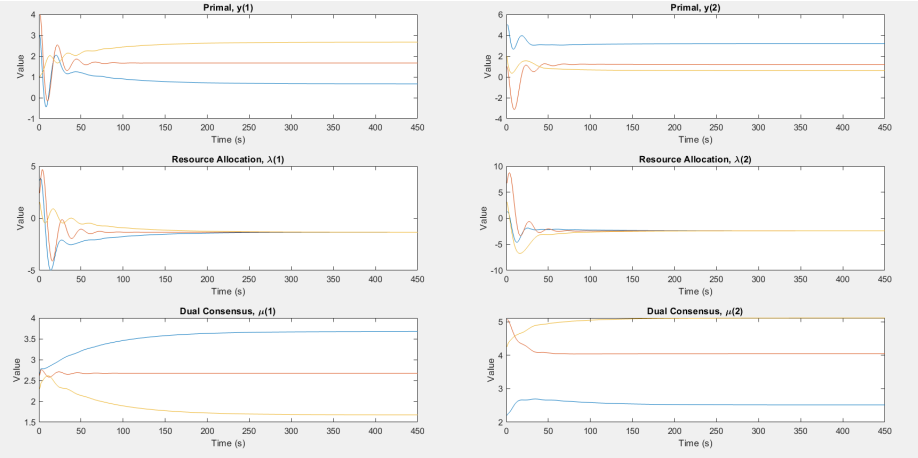

Figure 4.1: Simulation 1 with known cost functions using the Newton Distributed Resource Allocation strategy

The results show that our algorithm is able to determine the correct optimum at $f(x^*) = 8.93$.

4.3.2 3-agent system with unknown cost functions

Next, we consider the case where each node is given no information on their local cost function. To optimize the unknown cost function, we locally implement the Newton-seeking architecture outlined in Section 3.3. The time interval for the extremum-seeking architecture was set to $\Delta t = \frac{1}{1000}$. Following [92], we set the frequencies for each agent at

$$\omega = \begin{bmatrix} 100 & 300 \\ 125 & 375 \\ 145 & 435 \end{bmatrix}$$

with the low-pass and high-pass filter gains set to $\omega_l = \frac{1}{50}$ and $\omega_h = \frac{1}{800}$. The amplitude of each perturbation signal was set to $A_i = 0.5$. Finally, the forward-Euler discrete gain was set to $k = 0.995$.

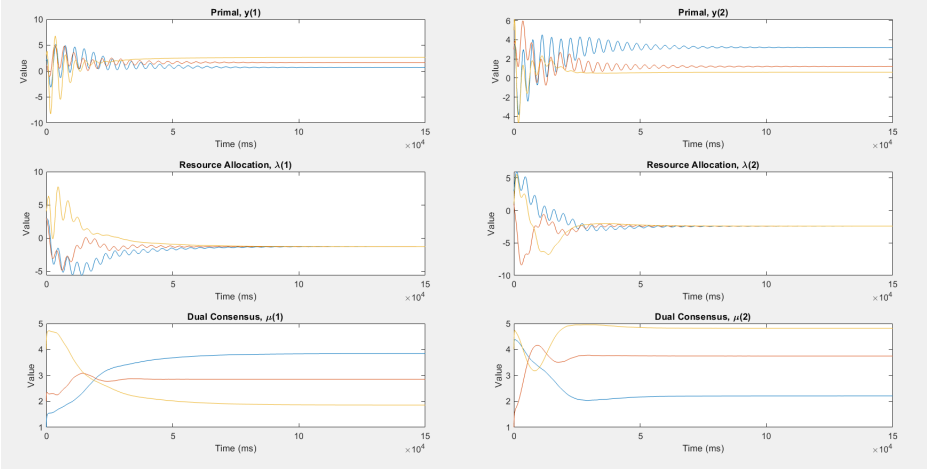

Figure 4.2: Simulation 1 with unknown cost functions using the Newton-seeking Distributed Resource Allocation strategy

The results shown in Figure 4.2 were calculated using the dynamic gains and step-sizes found in the table below.

Table 4.2: Dynamic gains and step-sizes for the unknown 3-node example using the Newton-Seeking Distributed Resource Allocation Strategy

Dynamic gain and step-size	ρ_p	γ_p	$\rho_{d,fast}$	$\gamma_{d,fast}$	$\gamma_{d,slow}$
Value	$\frac{1}{600}$	$\frac{1}{450}$	$\frac{1}{900}$	$\frac{1}{2500}$	$\frac{1}{2500}$

Again, the dual problem was iterated twice for each primal iteration, thereby increasing our convergence rate. As shown in Figure 4.2, the correct optimum was determined, indicating that the extremum-seeking architecture successfully converged to the Hessian and gradient of the system in a fully distributed fashion. The initial oscillations in the system are a result of the peak-seeking scheme converging to the correct Hessian and gradient. Furthermore, these oscillations are in part a result of allowing the dual problem to iterate more frequently than the primal problem.

4.3.3 25-agent system

To understand the scalability of the proposed strategy, the dynamics of Section 4.1 are tested using the 25-node case described in Section A. In this simulation, the total resource to be allocated is set to $R_1 = 45$ and $R_2 = 45$. The dynamic gains and step-sizes are found in table 4.3.

Table 4.3: Dynamic gains and step-sizes for the known 25-node example using the Distributed Newton Resource Allocation Strategy

Dynamic gain and step-size	ρ_p	γ_p	$\rho_{d,fast}$	$\gamma_{d,fast}$	$\gamma_{d,slow}$
Value	$\frac{1}{5}$	$\frac{15}{100}$	$\frac{7}{100}$	$\frac{1}{10}$	$\frac{7}{100}$

As shown in Figure 4.3, the optimal objective value was successfully determined at $f(x^*) = 57.31$. The dual problem was allowed to iterate twice per primal iteration. Furthermore, to ensure that the dual dynamics approached the correct equilibrium position, a gain of $k = 0.8$ is applied to the forward-euler discrete step of the dual dynamics.

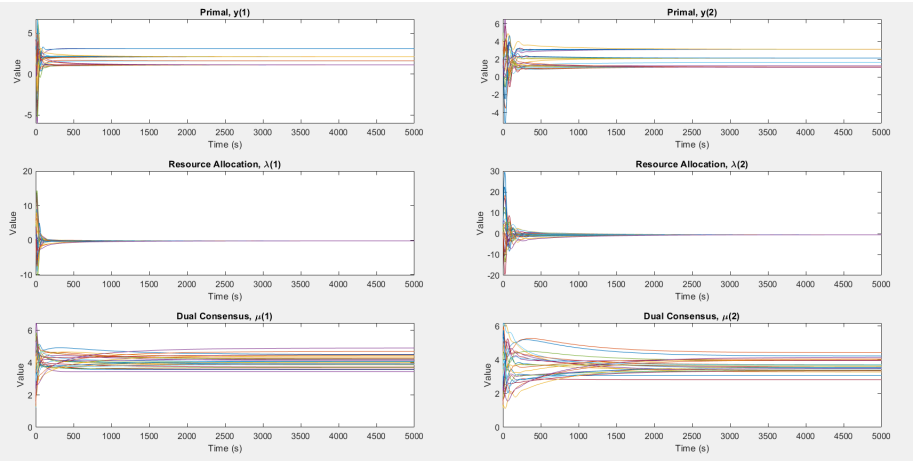

Figure 4.3: 25-node simulation with known functions using the Distributed Newton Resource Allocation strategy

4.3.4 25-agent system with unknown cost functions

With the addition of the peak-seeking scheme, we are again able to correctly identify the proper gradient and Hessian of the system after some transient oscillatory behaviour. Figure **??** illustrates the effectiveness of our strategy.

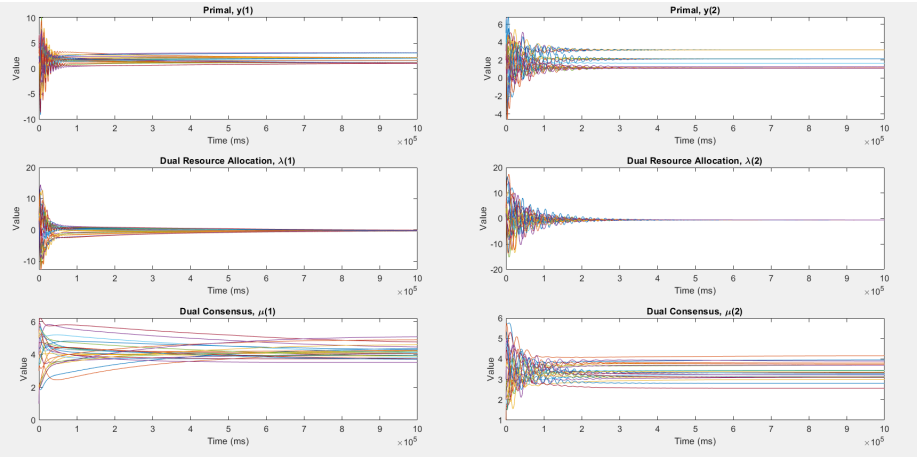

Figure 4.4: 25-node simulation with unknown functions using the Newton-seeking Distributed Resource Allocation strategy

The parameter initialization for the 25-node example with unknown cost functions follows the same strategy as the 3-node case, with the low and high-pass filters raised to $\omega_l = \frac{1}{700}$ and $\omega_h = \frac{1}{45}$, respectively. The dynamic gains and step-sizes were set to

Table 4.4: Dynamic gains and step-sizes for the unknown 25-node example using the Newton-Seeking Distributed Resource Allocation Strategy

Dynamic gain and step-size	ρ_p	γ_p	$\rho_{d,fast}$	$\gamma_{d,fast}$	$\gamma_{d,slow}$
Value	$\frac{1}{1000}$	$\frac{1}{1500}$	$\frac{1}{4500}$	$\frac{1}{4500}$	$\frac{1}{45000}$

To achieve a desirable convergence rate, we allowed the dual problem to iterate 8 times per primal iteration and set the forward-euler discrete gain to $k = 0.9995$.

To increase the stability of the solution dynamics when transitioning from the small-scale to large-scale cases, we found that the tuning parameters must be lowered. This is a direct result of the scale of the communication network; more autonomous agents require smaller step-sizes. This trend is consistent with findings in [96] and [28]. Furthermore, these parameters have to be further lowered with the addition of the Newton-seeking scheme to ensure stability during the the transient convergence process. This consequently forces the solution dynamics to be slowed down.

We found the most effective tuning strategy was to initially set the dynamic gains of the dual problem to $\rho_d = \gamma_d = \gamma_{d,slow} = 0$, and tune the primal dynamic gains such that the unconstrained problem (3.1) is solved. Next, we keep the dual μ dynamics at $\mathbf{0}$ and tune the resource allocation dynamics for the case where the λ values are not in consensus. These solution dynamics typically result in the primal problem being initially optimized, bringing the allocated resource away from $y_j(i) = \frac{R_j}{m}$ then, as the dual problem finds its optimum, the $y_j(i)$ terms are brought back to $\frac{R_j}{m}$. Finally, the dual μ dynamic gains are initialized at $\gamma_d = \gamma_{d,slow}$ and tuned to ensure long-term stability.

4.4 Conclusion

In this chapter, we presented our initial strategy to tackle the problem posed in Chapter 3. This strategy was developed for the case of twice differentiable convex cost functions that have a global minima. In creating dynamics for this problem, we employed a gradient-ascent optimization strategy to solve the resource constraint and created auxiliary variables v, d and q to define Newton steps for the primal, y, and dual consensus, μ, dynamics.

Through the stability analysis, we showed that for any fixed μ, our primal and resource allocation states are defined as $\lambda = -\frac{\partial f}{\partial y}$ at $y = \frac{R}{m} - \mathcal{M}\mu$. To prove the stability of the consensus dynamics, we utilized a coordinate transformation using the nullspace of the communication network and further showed that both the reduced and boundary layer systems are exponentially stable.

To create a more industry-applicable algorithm, we incorporated a distributed extremum-seeking architecture to estimate the Hessian and gradient of the local cost functions. The estimates of the distributed extremum-seeking architecture are proven to converge to the true functions and do not affect the stability of the system.

To test our algorithm, we performed two simulation examples, the first being a small-scale 3-node problem and the second being a more complex 25-node problem. In each simulation, our algorithm was tested twice, with and without known cost-functions and we were able to successfully determine the optimum collection vector in each case. In tuning our dynamic gains, we defined our tuning strategy

by starting with the primal case, then incorporating the dual resource constraint and finally adding the dual consensus. In approaching the problem in this way, we build complexity in the dynamics and ensure that the primal and resource vectors are always in equilibrium for any given μ.

Chapter 5

Generalized Newton-Seeking Distributed Resource Allocation

The strategy depicted in this chapter is a generalized version of the Newton-seeking Distributed Resource Allocation algorithm based on saddle-point dynamics that can be used to solve both global maxima/minima problems along with saddle-point problems with minor changes. This strategy decouples the resource allocation vector, $\boldsymbol{\lambda}$, from the primal dynamics by creating a Hessian of the Lagrangian with respect to y and λ. This pseudo-Hessian is constructed as follows

$$\hat{H} = \begin{bmatrix} -(\frac{\partial^2 f_i}{\partial y_i \partial y_j}) & \mathcal{I} \\ -\mathcal{I} & 0 \end{bmatrix} \tag{5.1}$$

where $\mathcal{I}$ is a $2m \times 2m$ identity matrix and 0 is a $2m \times 2m$ zeros matrix. By altering this Hessian, we propose a solution strategy to tackle saddle-point problems in

Section 5.2.

Similar to the previous chapter, this Hessian is effective for the case where the cost functions are known and the second derivative of the primal problem can be inverted directly. Inverting the estimated primal Hessian creates unstable dynamics in the unknown cost function case. To circumvent this issue, we employed the indirect Newton's method with the extremum-seeking scheme to estimate the primal Hessian then utilize it indirectly in the dynamics.

5.1 Solution Dynamics

To generalize the Newton-seeking Distributed Resource Allocation strategy, we alter the primal dynamics using the Lagrangian formed in equation (3.7). To decouple the the resource allocation vector, $\boldsymbol{\lambda}$, from the primal dynamics, an extended gradient is created by concatenating the critical points defined in equations (3.9) and (3.10)

$$\mathbf{g}_e = \begin{bmatrix} -\frac{\partial L}{\partial y_j} \\ \frac{\partial L}{\partial \lambda_j} \end{bmatrix} \tag{5.2}$$

Using this extended gradient, we create the following dynamics

$$\dot{\mathbf{v}}_e = -\gamma_e(-\hat{H}\mathbf{v}_e - \mathbf{g}_e) \tag{5.3}$$

where $\mathbf{v}_e$ is defined as the extended column vector

$$\mathbf{v}_e = \begin{bmatrix} v_1(1) & v_2(1) & \dots & v_1(m) & v_2(m) & \lambda_1(1) & \lambda_2(1) & \dots & \lambda_1(m) & \lambda_2(m) \end{bmatrix}. \tag{5.4}$$

Using the dynamics of equation (5.3), the Newton step has a unique equilibrium position at $\mathbf{v}_e = -\hat{H}^{-1}\mathbf{g}_e$. The dynamics of equation (5.3) can be split between the first and last $2m$ indices to better show the primal and resource allocation dynamics:

$$\dot{\mathbf{v}}_r(k) = -\gamma_p(\sum_{j=1}^{n} -H_{rj}\mathbf{v}_r(k) - \frac{\partial f_k}{\partial \mathbf{y}_r(k)}) \tag{5.5}$$

$$\dot{\mathbf{y}}_r(k) = \rho_p \mathbf{v}_r(k) \tag{5.6}$$

$$\dot{\boldsymbol{\lambda}}_r(k) = \gamma_d(-\mathbf{v}_r + \mathbf{y}_r - \frac{R_r}{m} + \sum_{j=1}^{m} \mathcal{L}_{kj}\boldsymbol{\mu}_r(j)) \tag{5.7}$$

As shown in equation (5.5), the Lagrange multiplier, λ, does not appear in the primal dynamics. Furthermore, the primal dynamic in (5.5) will have the unique asymptotic equilibrium at $\mathbf{v} = \mathbf{0}$. The resource allocation dynamics presented in equation (5.7) utilizes two equilibrium positions. From the primal dynamics, $\mathbf{v}$ will approach $\mathbf{0}$ asymptotically and the dynamics in equation (5.7) will approach $\mathbf{y} = \frac{R}{m} - \mathcal{M}\boldsymbol{\mu}$.

The dynamics for the consensus Lagrange multiplier, μ, directly follow the dynamics in section 4.1.2.

5.2 Saddle-point Case

To solve saddle-point problems, we propose a method to alter the pseudo-Hessian and extended gradient to simultaneously optimize convex and concave problems. The proposed pseudo-Hessian is defined as follows

$$\hat{H} = \begin{bmatrix} -(\frac{\partial^2 f}{\partial y^T \partial y}) & -\mathcal{I} \\ \mathcal{I} & 0 \end{bmatrix} \tag{5.8}$$

with the extended gradient defined as:

$$\mathbf{g}_e = \begin{bmatrix} -\frac{\partial L}{\partial y_j} \\ \frac{\partial L}{\partial \lambda_j} \end{bmatrix}. \tag{5.9}$$

Using these matrices and vectors, the dual resource allocation, $\boldsymbol{\lambda}$, vector will be decoupled from the primal problem. Taken together, we achieve dynamics that are able to minimize the convex cost functions while maximizing the concave cost functions and bring our primal system to the equilibrium point of the saddle point problem.

5.3 Stability Analysis

To assess the stability of the primal and resource allocation dynamics posed in-Section 5.1, we first must prove that the pseudo-Hessian is Hurwitz as follows.

Given the pseudo-Hessian,

$$H_e = \begin{bmatrix} -\nabla^2_{yy} L & \mathcal{I} \\ -\mathcal{I} & 0 \end{bmatrix} \tag{5.10}$$

the trace of H_e for the global minima problem is defined to be positive. Furthermore, due to the structure of H_e, the determinant will equal 1 for any separable cost function. Therefore, we can state that the real parts of the associated eigenvalues will be strictly negative, hence, the matrix is Hurwitz.

Using the extended gradient defined in equation (5.2), we further define the derivative of $\mathbf{g}_e$ as

$$\dot{\mathbf{g}_e} = \rho_e H_e \mathbf{v}_e. \tag{5.11}$$

The stability of the primal and dual resource dynamics are assessed through the Lyapunov function candidate

$$V = \frac{1}{2}\mathbf{v}_e^T \mathbf{v}_e + \frac{1}{2}\mathbf{g}_e^T \mathbf{g}_e + \frac{1}{2}\mathbf{v}_e^T \mathbf{g}_e + \beta(L - L^*) \tag{5.12}$$

where L^* is defined as the optimal Lagrangian, and $\mathbf{v}_e^T \mathbf{g}_e$ is positive definite. The derivative of the Lyapunov candidate function takes the form

$$\begin{aligned} \dot{V} = &- \gamma_e \mathbf{v}_e^T H_e \mathbf{v}_e - \gamma_e \mathbf{v}_e^T \mathbf{g}_e - \rho_e \mathbf{g}_e^T H_e \mathbf{v}_e \\ &- \frac{\gamma_e \mathbf{g}_e^T}{2}(H_e \mathbf{v}_e + \mathbf{g}_e) + \frac{\rho_e}{2}\mathbf{v}_e^T H_e \mathbf{v}_e + \beta \rho_e \mathbf{g}_e^T \mathbf{v}_e \end{aligned} \tag{5.13}$$

After collecting terms, the gradient of the Lyapunov candidate function becomes

$$\dot{V} = -(\gamma_e - \frac{\rho_d}{2})\mathbf{v}_e^T H_e \mathbf{v}_e - (\gamma_e - \beta\rho_e)\mathbf{g}_e^T \mathbf{v}_e - (\rho_e - \frac{\gamma_e}{2})\mathbf{g}_e^T H_e \mathbf{v}_e + \frac{\gamma_e}{2}\mathbf{g}_2^T \mathbf{g}_e \tag{5.14}$$

Through setting $\rho_e = \frac{\gamma_e}{2}$ and $\beta = 2$, equation (5.13) becomes

$$\dot{V} = \frac{-3\gamma_e}{4}\mathbf{v}_e^T H_e \mathbf{v}_e - \frac{\gamma_e}{2}\mathbf{g}_e^T \mathbf{g}_e \tag{5.15}$$

and is negative definite. As a result, we show that the equilibrium of the primal dynamics is exponentially stable. The dual consensus dynamics are unchanged from the previous strategy and will follow the same stability analysis for the singularly perturbed and reduced order systems.

5.4 Simulation Results

The simulation examples used to assess the efficiency of the Generalized Newton-Seeking Distributed Resource Allocation strategy are depicted in Section A. The collection vectors and Lagrange multipliers were initialized as random vectors with values between 1 and 5. We finish the simulation examples by demonstrating how the connectivity of the graph structure impacts the convergence of the algorithm with an additional reduced communication example. The convergence results will be compared to the Newton-seeking Distributed Resource Allocation strategy in Section 5.4.6.

5.4.1 3-agent system

We begin the simulation study with the 3-node case with known cost functions. The results illustrated in Figure 5.1 were simulated using the following dynamic gains and step-sizes

Table 5.1: Dynamic gains and step-sizes for the known 3-node example using the Distributed Newton Resource Allocation Strategy

Dynamic gain and step-size	ρ_p	γ_p	$\rho_{d,fast}$	$\gamma_{d,fast}$	$\rho_{d,slow}$	$\gamma_{d,slow}$
Value	$\frac{2}{3}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{3}{10}$	$\frac{1}{2}$

We further let the dual problem, defined as the consensus problem, iterate 4 times per primal iteration. As shown in Figure 5.1, the generalized algorithm converges to the correct optimum at $f(x^*) = 8.93$.

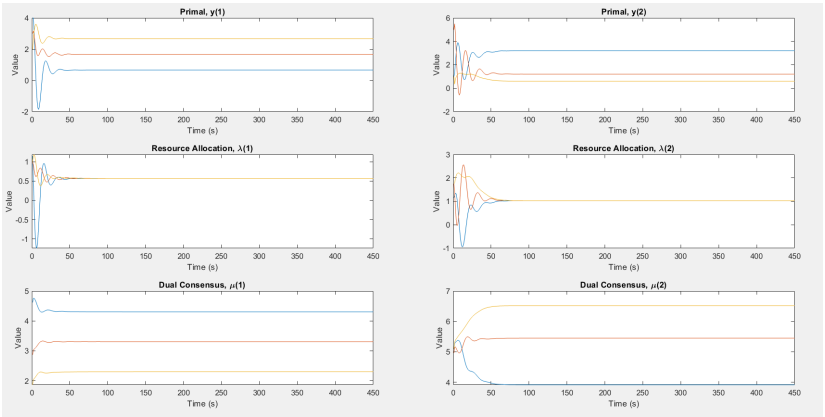

Figure 5.1: Simulation 1 with known cost functions using the Generalized Distributed Newton Resource Allocation strategy

5.4.2 3-agent system with unknown cost functions

To address the unknown local cost function case, we incorporated the peak-seeking scheme outlined in Section 3.3. Similar to the previous strategy, the low-pass and high-pass filter gains are set to $\omega_l = \frac{1}{45}$ and $\omega_h = \frac{1}{600}$, respectively. To ensure the estimate of the dual gradient and the consensus update parameter, $\mathbf{q}$, approach $\mathbf{0}$, we implement the following gain to the dual consensus dynamics' forward-Euler discrete step

$$k_e = \frac{\Psi}{\Psi + 1}$$

where Ψ is defined as the number of times that the primal problem has iterated.

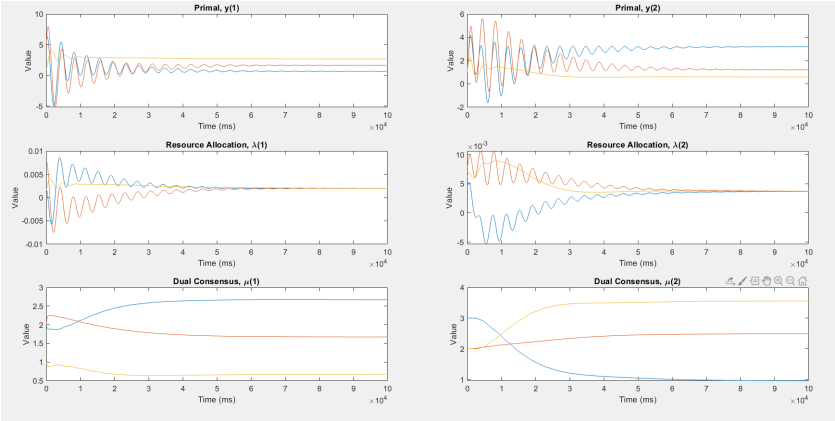

Figure 5.2: Simulation 1 with unknown cost functions using the Generalized Newton-seeking Distributed Resource Allocation strategy

The Generalized Newton-Seeking Distributed Resource Allocation algorithm correctly identifies the optimum with some initial oscillations, as shown in Figure 5.2. The initial oscillation can be attributed to three factors; the sinusoidal dither

signal, initialization of the Lagrange multipliers far from their optimums, as well as the algorithm allowing the dual consensus problem to iterate 5 times per primal iteration. The dynamic gains and step-sizes are found in the following table.

Table 5.2: Dynamic gains and step-sizes for the unknown 3-node example using the Newton-seeking Distributed Resource Allocation Strategy

Dynamic gain and step-size	ρ_p	γ_p	$\rho_{d,fast}$	$\gamma_{d,fast}$	$\rho_{d,slow}$	$\gamma_{d,slow}$
Value	$\frac{1}{600}$	$\frac{1}{300}$	$\frac{1}{650}$	$\frac{1}{1400}$	$\frac{1}{650}$	$\frac{1}{1400}$

5.4.3 25-agent system

Aside from tuning considerations, the convergence rate of multi-agent systems are limited by the connectivity of the communication network. The connectivity of the system can be examined through the eigenvalues of the network's Laplacian. The system created in Figure **??** has 6 eigenvalues at
$\Omega = [7.65\ 6.92\ 6.48\ 5.88\ 5.42\ 5.17]$, which will be compared to those in the low communication case in Section 5.4.6.

In the 25-node example, the Generalized Newton-seeking Distributed Resource Allocation strategy was able to determine the correct optimum at $f(x^*) = 57.31$. As shown in Figure 5.3, we observe the same transient oscillatory behaviour.

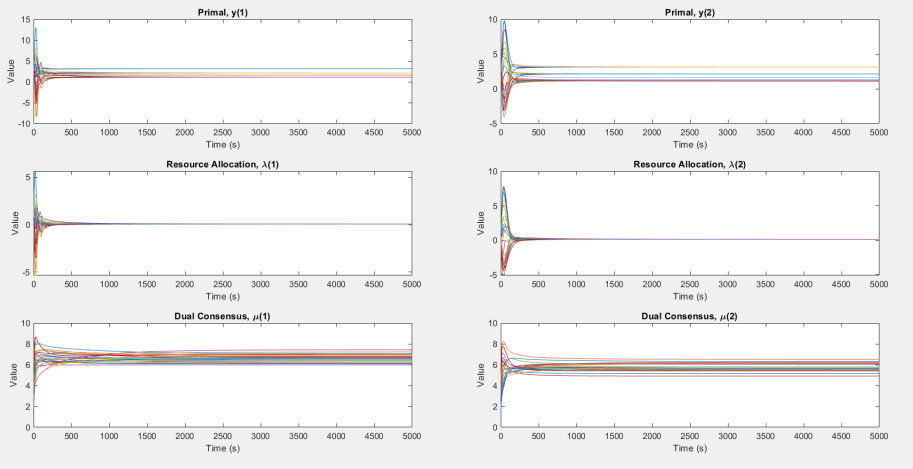

Figure 5.3: Simulation 2 with known cost functions using the Generalized Newton-seeking Distributed Resource Allocation strategy

To achieve the dynamics depicted in Figure 5.3, we used the following dynamic gains and step-sizes. We further let the dual consensus problem iterate 2 times per primal iteration.

Table 5.3: Dynamic gains and step-sizes for the known 25-node example using the Distributed Newton Resource Allocation Strategy

Dynamic gain and step-size	ρ_p	γ_p	$\rho_{d,fast}$	$\gamma_{d,fast}$	$\rho_{d,slow}$	$\gamma_{d,slow}$
Value	$\frac{15}{100}$	$\frac{15}{100}$	$\frac{5}{100}$	$\frac{5}{100}$	$\frac{1}{5}$	$\frac{15}{100}$

5.4.4 25-agent system with unknown cost functions

We now consider the dynamics with the incorporation of the peak-seeking scheme. The low and high pass-filters were increased to aid with convergence and set to

$\omega_l = \frac{1}{12.5}$ and $\omega_h = \frac{1}{600}$

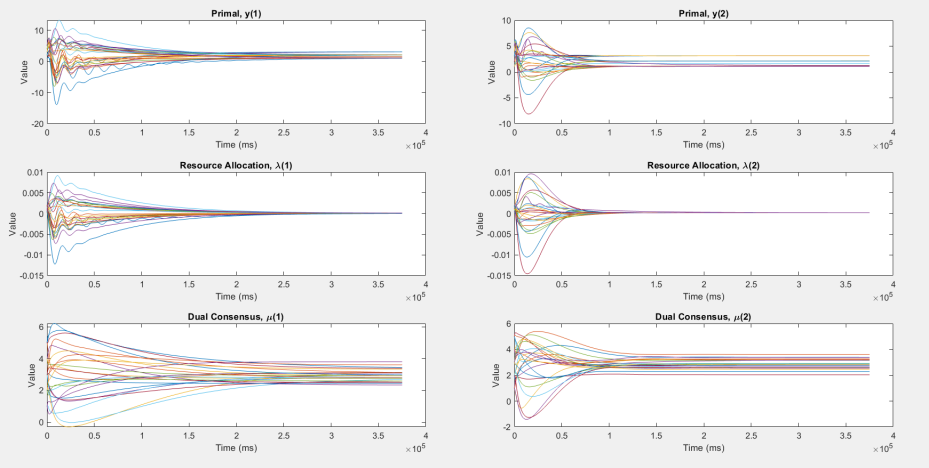

Figure 5.4: Simulation 2 with unknown cost functions using the Generalized Newton-seeking Distributed Resource Allocation strategy

To create the dynamics of Figure 5.4, we set the primal and dual dynamic gains as per table 5.4.

Table 5.4: Dynamic gains and step-sizes for the unknown 25-node example using the Newton-seeking Distributed Resource Allocation Strategy

Dynamic gain and step-size	ρ_p	γ_p	$\rho_{d,fast}$	$\gamma_{d,fast}$	$\rho_{d,slow}$	$\gamma_{d,slow}$
Value	$\frac{1}{2000}$	$\frac{1}{2000}$	$\frac{1}{1000}$	$\frac{1}{15000}$	$\frac{1}{2500}$	$\frac{1}{2500}$

For this simulation, we allowed the dual consensus problem to iterate 9 times per primal iteration. Using this strategy, we were able to achieve consensus at the optimal cost function value of $f(x^*) = 57.31$.

5.4.5 25-agent system with reduced communication

To show how the communication network will impact the convergence rates of the dynamics, a third communication network was created. In this configuration, the 6 eigenvalues of the Laplacian are $\Omega = [5.5485, 4.9053, 4.7757, 4.4981, 4.3628, 3.6980]$, with the smallest eigenvalue at 71.5% of the previous example. The configuration is found in Figure 5.5.

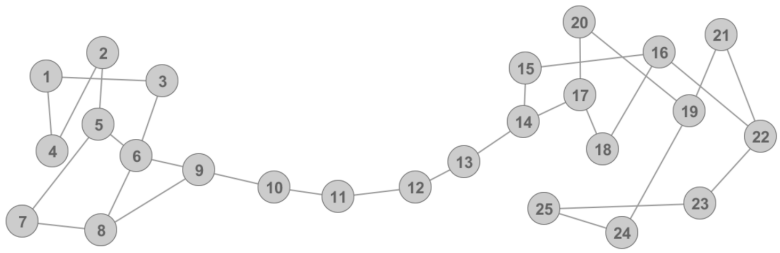

Figure 5.5: 25-node configuration with reduced communication

The Laplacian created from this configuration is represented in Appendix A. To better understand the results with reduced communication, we initialized the dynamic gains, step-sizes and additional dual iterations at the same values in the previous network configuration. The dynamic gains and step-sizes are found in the table below.

Table 5.5: Dynamic gains and step-sizes for the known 25-node with reduced communication example using the Distributed Newton Resource Allocation Strategy

Dynamic gain and step-size	ρ_p	γ_p	$\rho_{d,fast}$	$\gamma_{d,fast}$	$\rho_{d,slow}$	$\gamma_{d,slow}$
Value	$\frac{15}{100}$	$\frac{15}{100}$	$\frac{5}{100}$	$\frac{5}{100}$	$\frac{1}{5}$	$\frac{15}{100}$

The simulated dynamics are illustrated in the following figure.

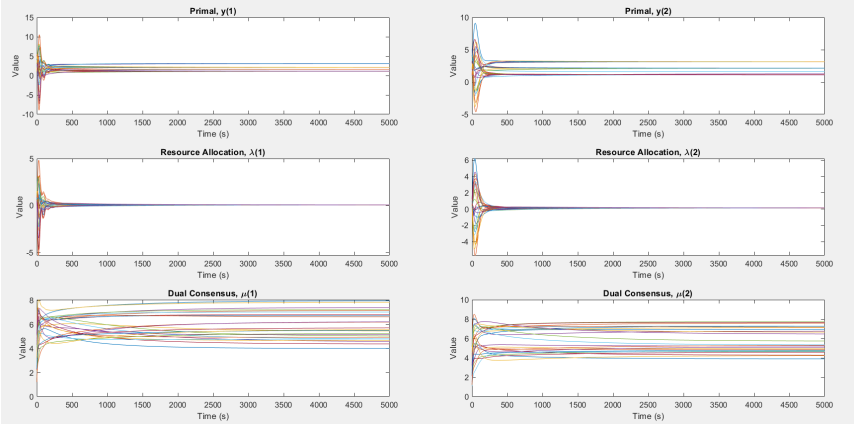

Figure 5.6: 25-node simulation with reduced communication and known cost functions using the Generalized Newton-seeking Distributed Resource Allocation

Comparing the convergence rates of the increased and reduced communication systems, the algorithm was able to correctly converge to the optimum at $f(x^*) = 8.93$. However, as shown by the dual consensus and resource allocation dynamics, the reduced communication system took longer to converge to the correct Lagrange multipliers.

5.4.6 Comparison of the Newton-seeking Distributed Resource Allocation strategy and the Generalized strategy

Two factors to consider when comparing the solution strategies proposed in this thesis are; convergence rate of the solutions and stability of the dynamics. To

quantify the convergence rate and stability, we take the value of the dynamic gains and step-sizes into consideration.

Table 5.6: Dynamic gains and step-sizes for each simulation

Dynamic gain and step-size	3-node known cost function	25-node known cost function	3-node unknown cost function	25-node unknown cost function
Newton-seeking Distributed Resource Allocation strategy				
γ_p	$\frac{3}{10}$	$\frac{15}{100}$	$\frac{1}{450}$	$\frac{1}{1500}$
$\gamma_{d,slow}$	$\frac{1}{5}$	$\frac{7}{10}$	$\frac{1}{2500}$	$\frac{1}{7000}$
$\gamma_{d,fast}$	$\frac{15}{100}$	$\frac{1}{10}$	$\frac{1}{400}$	$\frac{1}{40000}$
ρ_p	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{600}$	$\frac{1}{1000}$
$\rho_{d,fast}$	$\frac{15}{100}$	$\frac{7}{100}$	$\frac{1}{900}$	$\frac{1}{12000}$
Generalized Newton-seeking Distributed Resource Allocation strategy				
γ_p	$\frac{1}{4}$	$\frac{15}{100}$	$\frac{1}{300}$	$\frac{1}{2000}$
$\gamma_{d,slow}$	$\frac{1}{2}$	$\frac{15}{100}$	$\frac{1}{1400}$	$\frac{1}{2500}$
$\gamma_{d,fast}$	$\frac{1}{4}$	$\frac{5}{100}$	$\frac{1}{1400}$	$\frac{1}{15000}$
ρ_p	$\frac{2}{3}$	$\frac{15}{100}$	$\frac{1}{600}$	$\frac{1}{2000}$
$\rho_{d,slow}$	$\frac{3}{10}$	$\frac{1}{5}$	$\frac{1}{650}$	$\frac{1}{2500}$
$\rho_{d,fast}$	$\frac{1}{4}$	$\frac{5}{100}$	$\frac{1}{1400}$	$\frac{1}{1000}$

Table 5.6 shows that in every simulation, the Generalized version of the Newton-seeking Distributed Resource Allocation strategy had gains and step-sizes of comparable or larger values. By scaling the the Lagrange multipliers through the Generalized strategy, we are able to create a more aggressive solution strategy. Ultimately this indicates that the generalized distributed resource allocation strategy can be initialized with a larger range of primal and dual gains, allowing for a more robust strategy.

However, in the Newton-seeking Distributed Resource Allocation strategy, the primal and dual problems are solved in sequential steps. This allows the dual consensus and resource allocation problems to be iterated numerous times per primal iteration and, as a result, the algorithm achieves similar convergence results. In contrast, in the generalized algorithm, the resource allocation problem is solved with the primal problem and only the consensus problem can be iterated.

5.5 Conclusion

In this chapter we provided an alternative set of dynamics to solve the resource allocation problem posed in Chapter 3. These dynamics featured Newton's method to optimize primal problem along with the dual consensus problem. They further allowed the resource allocation dynamics to be decoupled from the primal problem through the creation of a pseudo-Hessian across both the primal collection, and resource allocation vectors.

Using this methodology, the generalized strategy can be further altered to tackle saddle-point problems. The rationale behind the alterations stemmed from assigning a direction to each gradient of the Lagrangian with respect to the primal vector $\mathbf{y}$, allowing both the concave and convex problems to be optimized. We outlined the procedure to achieve these dynamics in this chapter for future work.

To prove exponential stability of the proposed fast dynamics, we initially analyzed the pseudo-Hessian and determined that it is Hurwitz for the global minima problem. The stability of the equilibrium at the critical points is then investigated

through a Lyapunov analysis. Through this analysis, the equilibrium point for the fast dynamics are determined to be exponentially stable. The dual consensus dynamics are unchanged from the previous strategy, and therefore, the same stability analysis applies.

After the dynamics were created and exponential stability was confirmed, we performed a simulation study that utilized two examples of contrasting size. The first example used 3 agents and was created to analyze the dynamics, define correct tuning parameters and test the performance of the peak-seeking scheme. The second case was created to challenge the efficiency of our algorithm on a larger, more complex problem. The 25 agent case took longer to solve as the communication network was more sparse than the 3 agent case. Furthermore, with an increased number of nodes in the system, the dynamic gains and step-sizes must be reduced to ensure stability across the network. As shown in our results, our algorithm is able to determine the correct optimum in each case.

Unlike in the previous chapter, to tune this algorithm, we needed to tune the primal dynamics along with the resource allocation dynamics together as the λ terms were prevalent in the creation of the primal dynamics. However, it was still imperative to tune the primal dynamic gains such that the primal problem was always in equilibrium compared to the dual. To increase the convergence rate of our dynamics, we allowed the dual problem to be iterated numerous times per primal iteration.

Chapter 6

Conclusions

In this thesis we built the understanding of multi-agent systems, distributed optimization problems and extremum-seeking control; highlighting prominent work in the field and how each are being applied to further the understanding of control theory. We further applied each concept to solve resource allocation problems in a distributed setting. Resource allocation problems inherently involve optimizing a separable, often convex, cost function given the constraint that the sum of the allocated resource must equal to the total resource available. To solve this problem, we applied a dual decomposition to decouple the constraints, creating a dual Lagrange multiplier for resource allocation. After this decomposition, we applied an additional constraint that the resource allocation Lagrange multipliers must be in consensus across the communication network. This additional constraint was created such that we would achieve a trade-off between optimizing the primal problem and the dual problem. To consolidate this constraint into the

optimization problem, another dual decomposition was taken.

To create a more applicable algorithm, we implemented the model-free optimization technique known as extremum-seeking control. This techniques utilizes a perturbation signal within their closed feedback control loops to estimate the gradient of the cost function without any explicit knowledge. By implementing this strategy, we considered the case where each agent is not aware of the mathematical formulation of their local cost function. Furthermore, using advances in this field, we applied a peak-seeking scheme to estimated the local cost function's Hessian and employed Newton step dynamics. These dynamics are known to be more direct as compared to their gradient descent based counterparts. Our technique to estimate the gradient and Hessian are applied in a distributed setting and are known to converge to the actual cost functions given enough iterations.

In Chapter 4 we designed the first solution dynamics that utilized a Newton step to optimize the primal and dual consensus problems. As these dynamics were designed to be fully distributed, the primal dynamics can easily utilize the cost function's Hessian. In the dual consensus, the Hessian is harder to use as we established collection vectors across y_1 and y_2. To circumvent this, we created local auxiliary variables that utilized the structure of the communication network to converge to the gradient of the dual problem, allowing a Newton step to be implemented while preserving the security of the system. The dual resource allocation variables utilized gradient descent dynamics based on their critical point.

After creating the solution dynamics, we proved their validity through a stability analysis. The point of this stability analysis was to show that our solution

dynamics will approach the optimal solution asymptotically. This was accomplished by applying a single perturbation analysis and by proving the asymptotic stability for the consensus problem. The stability analysis was performed in continuous time and it was shown that using a forward-euler discretization could be stability preserving for appropriate step-sizes.

To test the effectiveness of this strategy, we devised a simulation study based on 4 cases; a 3-agent example with and without knowledge of the cost functions, and a 25-agent example with and without knowledge of the cost functions. In the 3-agent known-function case, our algorithm converges to the correct solution with minimal oscillatory behaviour. In the unknown cost function case, we were still able to find the correct optimal values, however, more time was required to let our peak-seeking scheme converge to the proper Hessian and gradient. In this case, the algorithm exhibited oscillatory behaviour when started far from the optimal values. The 25-agent case took more time to converge due in part to the increased size and complexity of the problem. With increased size and complexity, we are required to reduce the dynamic gains to increase our stability, thereby slowing down the convergence rate.

In Chapter 5, we approached the problem in a slightly different manner. In this approach, we generalized the strategy proposed in the previous chapter by creating a larger pseudo-Hessian across both the primal and dual $\boldsymbol{\lambda}$ vectors along with the extended $\mathbf{v}$ vector. This approach allowed for saddle-point problems to be solved with minimal changes in the cost function's Hessian and gradient. As a result of the pseudo-Hessian, the resource allocation vector was fully decoupled

from the primal dynamics, allowing solution trajectories to be less oscillatory.

The stability analysis for the Generalized Newton-seeking distributed Resource Allocation strategy was performed using the same techniques as in the initial case, by first writing the dynamics in terms of their error variables followed by a singular perturbation analysis. The dynamics of the new primal and resource allocation dynamics were shown to be exponentially stable through a Lyapunov analysis, after proving the pseudo-Hessian was Hurwitz. The consensus dynamics were unchanged for this strategy and the stability analysis followed the same methodology as the previous.

When transitioning from the smaller to larger simulation cases, we were forced to decrease the dynamic gains to increase the stability of the system. To show how the sparsity of the communication network affects the solution dynamics, we created a second 25-agent case with a more sparse communication network. The decrease in communication was proven through the Laplacian's associated eigenvalues. Through this comparison, we demonstrated slower convergence rate in the dynamics as shown through the convergence of the dual vectors.

We were able to successfully create fully distributed solution dynamics for the general resource allocation problem without the need for initialization conditions. In our approach, one is able to optimize multiple state vectors simultaneously with dynamics that converge at the same rate. We further applied these dynamics to the case where each node has an unknown cost-function whose structure must be estimated using a peak-seeking scheme. By applying recent advances in ESC, we estimated the Hessian directly, allowing the use of Newton step dynamics,

which on average converge faster than gradient based methods. Finally, by using dynamic gains, our algorithm has a user-assignable convergence rate.

6.1 Future work

The proposed distributed dynamics of this thesis can solve resource allocation problems for the cases where the static cost functions are either known and unknown.

In the development presented in this thesis, we made the assumption that the communication topology was static for the entirety of the solution dynamics. Such an assumption may be valid for specific industries, however as the Internet of Things and Smart Cities become more prevalent, it is necessary to develop solution strategies that can operate in changing and uncertain environments. This can be with respect to changing topologies, switching topologies and time-varying cost functions.

Further advances include creating strategies to minimize the need for frequent communication and computation. The implementation of event-triggered communication and control could prove useful to control these problems. In this strategy, once at steady state further computation will only occur once a change, such as a disturbance or new minima, is recognized.

As multi-agent systems are able to solve a broad class of problems, further applying this RTO strategy to the case where each node governs a linear-quadratic regulator could extend the proposed approach to the solution of distributed control

problems.

Bibliography

[1] E. H. Durfee and J. S. Rosenschein, "Distributed problem solving and multi-agent systems: Comparisons and examples," in *Proceedings of the Thirteenth International Distributed Artificial Intelligence Workshop*, 1994, pp. 94–104.

[2] A. Nedić, A. Olshevsky, and M. G. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 953–976, 2018.

[3] G. Soatti, M. Nicoli, A. Matera, S. Schiaroli, and U. Spagnolini, "Weighted consensus algorithms for distributed localization in cooperative wireless networks," in *2014 11th International Symposium on Wireless Communications Systems (ISWCS)*, 2014, pp. 116–120.

[4] L. Shuang, "Consensus problem of second-order multi-agent time-varying networks," in *2010 International Conference on Information, Networking and Automation (ICINA)*, vol. 1. IEEE, 2010, pp. V1–43.

[5] J. Wei and H. Fang, "Multi-agent consensus with time-varying delays and switching topologies," *Journal of Systems Engineering and Electronics*, vol. 25, no. 3, pp. 489–495, 2014.

[6] Z. Zuo, W. Yang, L. Tie, and D. Meng, "Fixed-time consensus for multi-agent systems under directed and switching interaction topology," in *2014 American Control Conference*. IEEE, 2014, pp. 5133–5138.

[7] N. A. Lynch, *Distributed algorithms*. Elsevier, 1996.

[8] S. S. Kia, "Distributed optimal resource allocation over networked systems and use of an e-exact penalty function," *IFAC-PapersOnLine*, vol. 49, no. 4, pp. 13–18, 2016.

[9] C. Li, S. Chen, J. Li, and F. Wang, "Distributed multi-step subgradient optimization for multi-agent system," *Systems & Control Letters*, vol. 128, pp. 26–33, 2019.

[10] C. Li, X. Yu, T. Huang, and X. He, "Distributed optimal consensus over resource allocation network and its application to dynamical economic dispatch," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 6, pp. 2407–2418, 2017.

[11] G. Soatti, M. Nicoli, A. Matera, S. Schiaroli, and U. Spagnolini, "Weighted consensus algorithms for distributed localization in cooperative wireless networks," in *2014 11th International Symposium on Wireless Communications Systems (ISWCS)*. IEEE, 2014, pp. 116–120.

[12] S. Wang, D. Sun, L. Du, and J. Ye, "Noncooperative distributed social welfare optimization with ev charging response," in *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2018, pp. 2097–2102.

[13] P. Frihauf, M. Krstic, and T. Basar, "Nash equilibrium seeking in noncooperative games," *IEEE Transactions on Automatic Control*, vol. 57, no. 5, pp. 1192–1207, 2012.

[14] P. Jan't Hoen, K. Tuyls, L. Panait, S. Luke, and J. A. La Poutre, "An overview of cooperative and competitive multiagent learning," in *International Workshop on Learning and Adaption in Multi-Agent Systems*. Springer, 2005, pp. 1–46.

[15] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[16] H. Lakshmanan and D. P. de Farias, "Decentralized approximate dynamic programming for dynamic networks of agents," in *2006 American Control Conference*. IEEE, 2006, pp. 6–pp.

[17] J. A. Bondy, U. S. R. Murty *et al.*, *Graph theory with applications*. Macmillan London, 1976, vol. 290.

[18] H. Yu and J.-G. Jian, "Multi-agent consensus with a time-varying reference state in directed network with switching topology and time-delay," in

2009 International Conference on Wavelet Analysis and Pattern Recognition. IEEE, 2009, pp. 476–481.

[19] Y. Sun, G. Scutari, and D. Palomar, "Distributed nonconvex multiagent optimization over time-varying networks," in *2016 50th Asilomar Conference on Signals, Systems and Computers*. IEEE, 2016, pp. 788–794.

[20] A. Cherukuri and J. Cortes, "Initialization-free distributed coordination for economic dispatch under varying loads and generator commitment," *Automatica*, vol. 74, pp. 183–193, 2016.

[21] D. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1439–1451, 2006.

[22] S. Boyd, L. Xiao, A. Mutapcic, and J. Mattingley, "Notes on decomposition methods," *Notes for EE364B, Stanford University*, vol. 635, pp. 1–36, 2007.

[23] D. Palomar, "Convex primal decomposition for multicarrier linear mimo transceivers," *IEEE Transactions on Signal Processing*, vol. 53, no. 12, pp. 4661–4674, 2005.

[24] D. Palomar, M. Bengtsson, and B. Ottersten, "Minimum ber linear transceivers for mimo channels via primal decomposition," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2866–2882, 2005.

[25] H. Pennanen, A. Tolli, and M. Latva-aho, "Decentralized coordinated downlink beamforming via primal decomposition," *IEEE Signal Processing Letters*, vol. 18, no. 11, pp. 647–650, 2011.

[26] Q. Ling and A. Ribeiro, "Decentralized dynamic optimization through the alternating direction method of multipliers," *IEEE Transactions on Signal Processing*, vol. 62, no. 5, pp. 1185–1197, 2013.

[27] B. Wang, Q. Fei, and Q. Wu, "Distributed time-varying resource allocation optimization based on finite-time consensus approach," *IEEE Control Systems Letters*, vol. 5, no. 2, pp. 599–604, 2020.

[28] K. Li, Q. Liu, S. Yang, J. Cao, and G. Lu, "Cooperative optimization of dual multiagent system for optimal resource allocation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 11, pp. 4676–4687, 2020.

[29] F. Qiu, J. Wang, C. Chen, and J. Tong, "Optimal black start resource allocation," *IEEE Transactions on Power Systems*, vol. 31, no. 3, pp. 2493–2494, 2015.

[30] H. Wang, K. Zeng, J. Liu, B. Du, and Y. Tang, "Power flow analysis of the economic dispatch considering the flexible ev charging," in *2020 IEEE 4th Conference on Energy Internet and Energy System Integration (EI2)*. IEEE, pp. 2885–2888.

[31] X. Jiang, K. Li, H. Jiang, N. Zhu, and X. Tong, "A bandwidth-link resources cooperative allocation strategy of data communication in intelligent transportation systems," *China Communications*, vol. 16, no. 4, pp. 234–249, 2019.

[32] W. Liu, Z.-H. Guan, and R. Liao, "Optimal resource allocation on heterogeneous complex transport networks," in *2010 Chinese Control and Decision Conference*. IEEE, 2010, pp. 2985–2988.

[33] K. Mochizuki and S.-i. Kuribayashi, "Evaluation of optimal resource allocation method for cloud computing environments with limited electric power capacity," in *2011 14th International Conference on Network-Based Information Systems*. IEEE, 2011, pp. 1–5.

[34] R. R. Reyes and T. Bauschert, "Adaptive and state-dependent online resource allocation in dynamic optical networks," *Journal of Optical Communications and Networking*, vol. 9, no. 3, pp. B64–B77, 2017.

[35] W. Zhang, S. Zhang, M. Cai, and J. Huang, "A new manufacturing resource allocation method for supply chain optimization using extended genetic algorithm," *The International Journal of Advanced Manufacturing Technology*, vol. 53, no. 9, pp. 1247–1260, 2011.

[36] K.-J. Wang and M.-J. Chen, "Cooperative capacity planning and resource allocation by mutual outsourcing using ant algorithm in a decentralized sup-

ply chain," *Expert Systems with Applications*, vol. 36, no. 2, pp. 2831–2842, 2009.

[37] K. Li, Q. Liu, S. Yang, J. Cao, and G. Lu, "Cooperative optimization of dual multiagent system for optimal resource allocation," *IEEE Transactions on systems, man, and cybernetics: systems*, vol. 50, no. 11, pp. 4676–4687, 2018.

[38] H. Li, Q. Lü, X. Liao, and T. Huang, "Accelerated convergence algorithm for distributed constrained optimization under time-varying general directed graphs," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 7, pp. 2612–2622, 2018.

[39] S. Liang, X. Zeng, and Y. Hong, "Distributed nonsmooth optimization with coupled inequality constraints via modified lagrangian function," *IEEE Transactions on Automatic Control*, vol. 63, no. 6, pp. 1753–1759, 2017.

[40] E. Fu, H. Gao, M. Fasehullah, and L. Tan, "A gradient tracking method for resource allocation base on distributed convex optimization," in *2019 3rd International Symposium on Autonomous Systems (ISAS)*, 2019, pp. 41–46.

[41] C. Li, X. Yu, T. Huang, and X. He, "Distributed optimal consensus over resource allocation network and its application to dynamical economic dispatch," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2407–2418, 2018.

[42] G. Chen and Z. Guo, "Initialization-free distributed fixed-time convergent algorithms for optimal resource allocation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020.

[43] P. Yi, Y. Hong, and F. Liu, "Initialization-free distributed algorithms for optimal resource allocation with feasibility constraints and application to economic dispatch of power systems," *Automatica*, vol. 74, pp. 259–269, 2016.

[44] Y. Xu, T. Han, K. Cai, Z. Lin, G. Yan, and M. Fu, "A distributed algorithm for resource allocation over dynamic digraphs," *IEEE Transactions on Signal Processing*, vol. 65, no. 10, pp. 2600–2612, 2017.

[45] Y. Zhu, W. Ren, W. Yu, and G. Wen, "Distributed resource allocation over directed graphs via continuous-time algorithms," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019.

[46] Y. Zhu, W. Yu, G. Wen, and D. Chen, "Continuous-time algorithm for distributed resource allocation over a weight-unbalanced digraph," in *2019 Chinese Control And Decision Conference (CCDC)*, 2019, pp. 56–59.

[47] L. Xiao and S. Boyd, "Optimal scaling of a gradient method for distributed resource allocation," *Journal of optimization theory and applications*, vol. 129, no. 3, pp. 469–488, 2006.

[48] Z. Deng, S. Liang, and Y. Hong, "Distributed continuous-time algorithms for resource allocation problems over weight-balanced digraphs," *IEEE Transactions on Cybernetics*, vol. 48, no. 11, pp. 3116–3125, 2018.

[49] S. E. Alam, R. Shorten, F. Wirth, and J. Y. Yu, "Communication-efficient distributed multi-resource allocation," in *2018 IEEE International Smart Cities Conference (ISC2)*, 2018, pp. 1–8.

[50] J. R. Marden and T. Roughgarden, "Generalized efficiency bounds in distributed resource allocation," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 571–584, 2014.

[51] J. Poveda and N. Quijano, "Distributed extremum seeking for real-time resource allocation," in *2013 American Control Conference*, 2013, pp. 2772–2777.

[52] M. Aghajani, L. Parolini, and B. Sinopoli, "Dynamic power allocation in server farms: A real time optimization approach," in *49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 3790–3795.

[53] M. Leblanc, "On the electrification of railways by means of alternating currents of high frequency," *Revue G ’e n é rale de l ’é l’ectricit ’e*, vol. 12, no. 8, pp. 275–277, 1922.

[54] M. Krstic and H.-H. Wang, "Design and stability analysis of extremum seeking feedback for general nonlinear systems," in *Proceedings of the 36th IEEE Conference on Decision and Control*, vol. 2, 1997, pp. 1743–1748 vol.2.

[55] M. Guay, D. Dochain, and M. Perrier, "Adaptive extremum seeking control of continuous stirred tank bioreactors with unknown growth kinetics," *Automatica*, vol. 40, no. 5, pp. 881–888, 2004.

[56] P. Cougnon, D. Dochain, M. Guay, and M. Perrier, "On-line optimization of fedbatch bioreactors by adaptive extremum seeking control," *Journal of Process Control*, vol. 21, no. 10, pp. 1526–1532, 2011.

[57] M. S. Stankovic and D. M. Stipanovic, "Stochastic extremum seeking with applications to mobile sensor networks," in *2009 American Control Conference*. IEEE, 2009, pp. 5622–5627.

[58] H.-B. Dürr, M. S. Stanković, and K. H. Johansson, "Distributed positioning of autonomous mobile sensors with application to coverage control," in *Proceedings of the 2011 American control conference*. IEEE, 2011, pp. 4822–4827.

[59] Q. Xu and L. Cai, "Active braking control of electric vehicles to achieve maximal friction based on fast extremum-seeking and reachability," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14 869–14 883, 2020.

[60] S. F. Toloue and M. Moallem, "Torque control of a brushless dc motor using multivariable sliding mode extremum seeking pi tuning," in *IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2017, pp. 2917–2922.

[61] R. Leyva and H. Zazo, "Saturation effects in digital extremum seeking control for photovoltaic applications," in *2013 IEEE Energy Conversion Congress and Exposition*. IEEE, 2013, pp. 4775–4779.

[62] C. Festila, R. Both, E.-H. Dulf, M. Secara, and S. Chetan, "Original extremum control method implemented in mpp photovoltaic systems," in *2015 19th International Conference on System Theory, Control and Computing (ICSTCC)*. IEEE, 2015, pp. 378–383.

[63] X. Li, Y. Li, J. E. Seem, and P. Lei, "Maximum power point tracking for photovoltaic systems using adaptive extremum seeking control," in *Dynamic Systems and Control Conference*, vol. 54754, 2011, pp. 803–810.

[64] T. R. Oliveira, M. Krstić, and D. Tsubakino, "Extremum seeking for static maps with delays," *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1911–1926, 2016.

[65] K. B. Ariyur and M. Krstic, "Analysis and design of multivariable extremum seeking," in *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)*, vol. 4. IEEE, 2002, pp. 2903–2908.

[66] M. Krstić, "Performance improvement and limitations in extremum seeking control," *Systems & Control Letters*, vol. 39, no. 5, pp. 313–326, 2000.

[67] F. D. Sahneh, G. Hu, and L. Xie, "Extremum seeking control for systems with time-varying extremum," in *Proceedings of the 31st Chinese Control Conference*. IEEE, 2012, pp. 225–231.

[68] E. Moshksar and M. Guay, "Invariant manifold approach for time-varying extremum-seeking control problem," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 9129–9134, 2014.

[69] A. Ghaffari, M. Krstić, and D. Nešić, "Multivariable newton-based extremum seeking," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, 2011, pp. 4436–4441.

[70] A. Ghaffari, M. Krstić, and S. Seshagiri, "Power optimization for photovoltaic microconverters using multivariable newton-based extremum seeking," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 6, pp. 2141–2149, 2014.

[71] M. Ye and G. Hu, "A distributed extremum seeking scheme for networked optimization," in *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 4928–4933.

[72] J. Ebegbulem and M. Guay, "Resource allocation for a class of multi-agent systems with unknown dynamics using extremum seeking control," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 2496–2501.

[73] C. Li, Z. Qu, and M. A. Ingram, "Distributed extremum seeking and cooperative control for mobile communication," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, 2011, pp. 4510–4515.

[74] J.-Y. Choi, M. Krstic, K. B. Ariyur, and J. S. Lee, "Extremum seeking control for discrete-time systems," *IEEE Transactions on automatic control*, vol. 47, no. 2, pp. 318–323, 2002.

[75] I. Vandermeulen, M. Guay, and P. J. McLellan, "Discrete-time distributed extremum-seeking control over networks with unstable dynamics," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 3, pp. 1182–1192, 2018.

[76] Q. Ling and Z. Tian, "Decentralized sparse signal recovery for compressive sleeping wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 58, no. 7, pp. 3816–3827, 2010.

[77] M. M. I. Khan and R. Hossain, "Efficient and fast convergent consensus algorithms for faulty nodes tracking in distributed wireless sensor networks," in *2016 3rd International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*. IEEE, 2016, pp. 1–6.

[78] G. Xiong and S. Kishore, "Linear high-order distributed average consensus algorithm in wireless sensor networks," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, pp. 1–6, 2010.

[79] W. Ren, "Consensus based formation control strategies for multi-vehicle systems," in *2006 American Control Conference*. IEEE, 2006, pp. 6–pp.

[80] J. Lin, X. Wang, and Q. Yang, "A consensus-based algorithm for electric vehicle charging system," in *2020 IEEE 4th Conference on Energy Internet and Energy System Integration (EI2)*. IEEE, pp. 4403–4406.

[81] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5262–5276, 2010.

[82] S. Boyd, N. Parikh, and E. Chu, *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[83] S. Sundhar Ram, A. Nedić, and V. V. Veeravalli, "A new class of distributed optimization algorithms: Application to regression of distributed data," *Optimization Methods and Software*, vol. 27, no. 1, pp. 71–88, 2012.

[84] X. Zhang, D. Liu, F. Yu, and D. Zhao, "A distributed alternating direction method of multipliers algorithm for consensus optimization," in *2019 Chinese Automation Congress (CAC)*. IEEE, 2019, pp. 4104–4107.

[85] R. Carli and M. Dotoli, "Distributed alternating direction method of multipliers for linearly constrained optimization over a network," *IEEE Control Systems Letters*, vol. 4, no. 1, pp. 247–252, 2019.

[86] I. Matei and J. S. Baras, "Performance evaluation of the consensus-based distributed subgradient method under random communication topologies," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 754–771, 2011.

[87] Q. Liu, S. Yang, and Y. Hong, "Constrained consensus algorithms with fixed step size for distributed convex optimization over multiagent networks,"

IEEE Transactions on Automatic Control, vol. 62, no. 8, pp. 4259–4265, 2017.

[88] F. Alshammari and A. El-Refaie, "Time-varying optimization-based consensus control for microgrid's secondary control," in *2021 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. IEEE, 2021, pp. 1–5.

[89] F. Zanella, D. Varagnolo, A. Cenedese, G. Pillonetto, and L. Schenato, "Newton-raphson consensus for distributed convex optimization," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*. IEEE, 2011, pp. 5917–5922.

[90] D. P. Bertsekas, *Convex optimization theory*. Athena Scientific Belmont, 2009.

[91] R. D. Brackston, A. Wynn, and J. F. Morrison, "Extremum seeking to control the amplitude and frequency of a pulsed jet for bluff body drag reduction," *Experiments in Fluids*, vol. 57, no. 10, pp. 1–14, 2016.

[92] M. Guay, "Distributed newton seeking," *Computers & Chemical Engineering*, vol. 146, p. 107206, 2021.

[93] R. Tutunov, H. Bou-Ammar, and A. Jadbabaie, "Distributed newton method for large-scale consensus optimization," *IEEE Transactions on Automatic Control*, vol. 64, no. 10, pp. 3983–3994, 2019.

[94] ——, "Distributed sddm solvers: Theory & applications," *arXiv preprint arXiv:1508.04096*, 2015.

[95] H. K. Khalil, *Nonlinear systems; 3rd ed.* Upper Saddle River, NJ: Prentice-Hall, 2002, the book can be consulted by contacting: PH-AID: Wallet, Lionel. [Online]. Available: https://cds.cern.ch/record/1173048

[96] J. Lei, H.-F. Chen, and H.-T. Fang, "Primal–dual algorithm for distributed constrained optimization," *Systems & Control Letters*, vol. 96, pp. 110–117, 2016.

Appendix A

Simulation study cases

A.0.1 3-agent system

The first simulation we explored was created using 3-agents in the following undirected configuration.

Figure A.1: 3-agent communication network

This communication network yields the following Laplacian

$$\mathcal{L} = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

The local cost functions are given by:

$$f_1(x_1) = x_1(1)^2 + (x_1(2) - 2)^2 \tag{A.1}$$

$$f_2(x_2) = (x_2(1) - 1)^2 + x_2(2)^2 \tag{A.2}$$

$$f_3(x_3) = 4 + (x_3(1) - 2)^2 + 2x_3(2)^2 \tag{A.3}$$

The total allocated resource constraints are set to

$$x_1(1) + x_2(1) + x_3(1) = 5 \tag{A.4}$$

$$x_1(2) + x_2(2) + x_3(2) = 5. \tag{A.5}$$

Under these conditions, the minimum of the Lagrangian (3.7) is $L(y^*, \lambda^*, \mu^*) = 8.93$ with λ^* at any constant that satisfies the critical point (3.9) and y^*, μ^* shown in Table 3.1.

Table A.1: Optimal values for the 3-agent system

y_1^*	y_2^*	μ_1^*	μ_2^*
$\frac{2}{3}$	$\frac{16}{5}$	$\frac{89}{20}$	$\frac{36}{25}$
$\frac{5}{3}$	$\frac{6}{5}$	$\frac{69}{20}$	$\frac{149}{25}$
$\frac{8}{3}$	$\frac{3}{5}$	$\frac{49}{20}$	$\frac{101}{25}$

A.0.2 25-agent system

The next simulation reflects a less academic scenario involving the graph shown in Figure **??**.

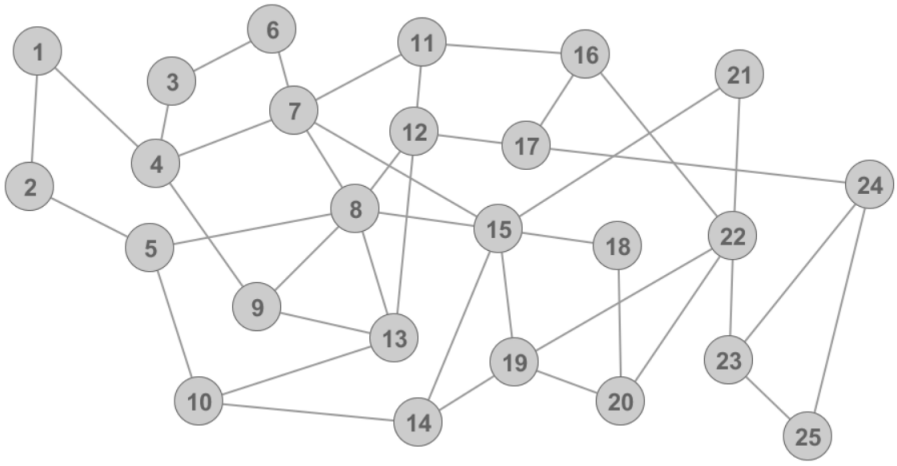

Figure A.2: 25-agent system

The Laplacian can be described through the following block matrix:

$$\mathcal{L} = \begin{bmatrix} \mathcal{L}_{1,1} & \mathcal{L}_{1,2} & 0 & 0 & 0 \\ \mathcal{L}_{2,1} & \mathcal{L}_{2,2} & \mathcal{L}_{2,3} & 0 & 0 \\ 0 & \mathcal{L}_{3,2} & \mathcal{L}_{3,3} & \mathcal{L}_{3,4} & 0 \\ 0 & 0 & \mathcal{L}_{4,3} & \mathcal{L}_{4,4} & \mathcal{L}_{4,5} \\ 0 & 0 & 0 & \mathcal{L}_{5,4} & \mathcal{L}_{5,5} \end{bmatrix}$$

with components:

$$\mathcal{L}_{1,1} = \begin{bmatrix} 2 & -1 & 0 & -1 & 0 \\ -1 & 2 & 0 & 0 & -1 \\ 0 & 0 & 2 & -1 & 0 \\ -1 & 0 & -1 & 4 & 0 \\ 0 & -1 & 0 & 0 & 3 \end{bmatrix} \mathcal{L}_{1,2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 \end{bmatrix}$$

$$\mathcal{L}_{2,1} = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix} \mathcal{L}_{2,2} = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 5 & -1 & 0 & 0 \\ 0 & -1 & 6 & -1 & 0 \\ 0 & 0 & -1 & 3 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix} \mathcal{L}_{2,3} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 \\ 0 & -1 & -1 & 0 & -1 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 \end{bmatrix}$$

$$\mathcal{L}_{3,2} = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & 0 & -1 & -1 \\ 0 & -1 & -1 & 0 & 0 \end{bmatrix} \mathcal{L}_{3,3} = \begin{bmatrix} 3 & -1 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & 0 \\ 0 & -1 & 4 & 0 & 0 \\ 0 & 0 & 0 & 3 & -1 \\ 0 & 0 & 0 & -1 & 5 \end{bmatrix} \mathcal{L}_{3,4} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & -1 & 0 \end{bmatrix}$$

$$\mathcal{L}_{4,3} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \mathcal{L}_{4,4} = \begin{bmatrix} 3 & -1 & 0 & 0 & 0 \\ -1 & 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & -1 \\ 0 & 0 & 0 & 4 & -1 \\ 0 & 0 & -1 & -1 & 3 \end{bmatrix} \mathcal{L}_{4,5} = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathcal{L}_{5,4} = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \mathcal{L}_{5,5} = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 5 & -1 & 0 & 0 \\ 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & -1 & -1 & 2 \end{bmatrix}.$$

The resource constraint for this multivariate problem is set to $R_1 = R_2 = 45$ and the cost functions for each node are highlighted in the following table:

Table A.2: Local cost functions for each node

$f_1(y_1, y_2) = (y_1(1) - 1)^2 + 2(y_2(1) - 1)^2 + 2$
$f_2(y_1, y_2) = (y_1(2) - 2)^2 + 2(y_2(2) - 1)^2 + 4$
$f_3(y_1, y_2) = (y_1(3) - 1)^2 + 2(y_2(3) - 2)^2 + 2$
$f_4(y_1, y_2) = (y_1(4) - 2)^2 + 2(y_2(4) - 3)^2$
$f_5(y_1, y_2) = (y_1(5) - 1)^2 + 2(y_2(5) - 1)^2 + 1$
$f_6(y_1, y_2) = (y_1(6) - 2)^2 + 2(y_2(6) - 1.5)^2 + 4$
$f_7(y_1, y_2) = (y_1(7) - 2)^2 + 2(y_2(1) - 1)^2 + 1$
$f_8(y_1, y_2) = (y_1(8) - 2)^2 + 2(y_2(8) - 3)^2 + 2$
$f_9(y_1, y_2) = (y_1(9) - 1)^2 + 2(y_2(9) - 1)^2 + 2$
$f_{10}(y_1, y_2) = (y_1(10) - 2)^2 + 2(y_2(10) - 2)^2 + 3$
$f_{11}(y_1, y_2) = (y_1(11) - 3)^2 + (y_2(11) - 1)^2 + 1$
$f_{12}(y_1, y_2) = (y_1(12) - 1)^2 + 2(y_2(12) - 2)^2 + 2$
$f_{13}(y_1, y_2) = (y_1(13) - 1.5)^2 + 2(y_2(13) - 1)^2$
$f_{14}(y_1, y_2) = (y_1(14) - 2)^2 + 2(y_2(14) - 3)^2 + 2$
$f_{15}(y_1, y_2) = (y_1(15) - 3)^2 + 2(y_2(15) - 3)^2 + 2$
$f_{16}(y_1, y_2) = (y_1(16) - 1.5)^2 + 2(y_2(16) - 1)^2 + 2$
$f_{17}(y_1, y_2) = (y_1(17) - 2)^2 + 2(y_2(17) - 1)^2 + 2$
$f_{18}(y_1, y_2) = (y_1(18) - 1)^2 + 2(y_2(18) - 1)^2 + 2$
$f_{19}(y_1, y_2) = (y_1(19) - 2)^2 + 2(y_2(19) - 1)^2 + 5$
$f_{20}(y_1, y_2) = (y_1(20) - 2)^2 + 2(y_2(20) - 2)^2 + 2$
$f_{21}(y_1, y_2) = (y_1(21) - 1)^2 + 2(y_2(21) - 2)^2 + 2$
$f_{22}(y_1, y_2) = (y_1(22) - 2)^2 + 2(y_2(22) - 2)^2 + 2$
$f_{23}(y_1, y_2) = (y_1(23) - 1)^2 + 2(y_2(23) - 1)^2 + 2$
$f_{24}(y_1, y_2) = (y_1(24) - 2)^2 + 2(y_2(24) - 3)^2 + 2$
$f_{25}(y_1, y_2) = (y_1(25) - 1)^2 + 3(y_2(25) - 1)^2 + 2$

The optimal Lagrangian given the resource constraints is $L(y^*, \lambda^*, \mu^*) = 57.31$. The optimal solution is given by λ^* at any constant that satisfies critical point (3.9) and y^* and μ^* at

Table A.3: Optimal values for the 25-agent system

y_1^*	y_2^*	μ_1^*	μ_2^*
1.12	1.136	4.66	5.14
2.12	1.136	4.36	5.34
1.12	2.136	4.49	4.19
2.12	3.136	4.28	4.28
1.12	1.136	4.37	4.87
2.12	1.636	4.02	4.43
2.12	1.136	3.88	4.50
2.12	3.136	4.08	4.31
1.12	1.136	4.32	4.63
2.12	2.136	4.01	4.30
3.12	1.273	3.60	4.87
1.12	2.136	4.18	4.59
1.62	1.136	4.22	4.63
2.12	3.136	3.75	3.74
3.12	3.136	3.72	3.98
1.62	1.136	4.05	4.97
2.12	1.136	4.17	4.91
1.12	1.136	4.35	4.39
2.12	1.136	3.85	4.28
2.12	2.136	4.03	4.27
1.12	2.136	4.62	4.26
2.12	2.136	4.22	4.47
1.12	1.136	4.84	4.90
2.12	3.136	4.58	4.51
1.12	1.09	5.05	5.06

A.1 25-agent system structure with reduced communication

The Laplacian of the 25-agent system with reduced communication depicted in Figure 5.5 is represented as follows:

$$\mathcal{L} = \begin{bmatrix} \mathcal{L}_{1,1} & \mathcal{L}_{1,2} & 0 & 0 & 0 \\ \mathcal{L}_{2,1} & \mathcal{L}_{2,2} & \mathcal{L}_{2,3} & 0 & 0 \\ 0 & \mathcal{L}_{3,2} & \mathcal{L}_{3,3} & \mathcal{L}_{3,4} & 0 \\ 0 & 0 & \mathcal{L}_{4,3} & \mathcal{L}_{4,4} & \mathcal{L}_{4,5} \\ 0 & 0 & 0 & \mathcal{L}_{5,4} & \mathcal{L}_{5,5} \end{bmatrix}$$

with components:

$$\mathcal{L}_{1,1} = \begin{bmatrix} 2 & 0 & -1 & -1 & 0 \\ 0 & 2 & 0 & -1 & -1 \\ -1 & 0 & 2 & 0 & 0 \\ -1 & -1 & 0 & 2 & 0 \\ 0 & -1 & 0 & 0 & 3 \end{bmatrix} \mathcal{L}_{1,2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathcal{L}_{2,1} = \begin{bmatrix} 0 & 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \mathcal{L}_{2,2} = \begin{bmatrix} 4 & 0 & -1 & -1 & 0 \\ 0 & 2 & -1 & 0 & 0 \\ -1 & -1 & 0 & -1 & 0 \\ -1 & 0 & 3 & 0 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix} \mathcal{L}_{2,3} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathcal{L}_{3,2} = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \mathcal{L}_{3,3} = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix} \mathcal{L}_{3,4} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathcal{L}_{4,3} = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \mathcal{L}_{4,4} = \begin{bmatrix} 3 & 0 & -1 & 0 & 0 \\ 0 & 3 & -1 & 0 & -1 \\ -1 & -1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 3 & -1 \\ 0 & -1 & 0 & -1 & 2 \end{bmatrix} \mathcal{L}_{4,5} = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathcal{L}_{5,4} = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \mathcal{L}_{5,5} = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 \\ 0 & -1 & 2 & 0 & -1 \\ 0 & 0 & 0 & 2 & -1 \\ 0 & 0 & -1 & -1 & 2 \end{bmatrix}.$$