Title
Distributed Optimization in Multi-Agent Systems: Game Theory Based Sensor Coverage and Continuous-Time Convex Optimization

Author
Rahili, Salar

Publication Date
2016

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
RIVERSIDE

Distributed Optimization in Multi-Agent Systems: Game Theory Based Sensor Coverage and continuous-time Convex Optimization

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

by

Salar Rahili

December 2016

Dissertation Committee:

Dr. Wei Ren, Chairperson
Dr. Jay Farrell
Dr. Fabio Pasqualetti

The Dissertation of Salar Rahili is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

I am grateful to my advisor, without whose help, I would not have been here.

To my parents for all the support.

ABSTRACT OF THE DISSERTATION

Distributed Optimization in Multi-Agent Systems: Game Theory Based Sensor Coverage and continuous-time Convex Optimization

by

Salar Rahili

Doctor of Philosophy, Graduate Program in Electrical Engineering
University of California, Riverside, December 2016
Dr. Wei Ren, Chairperson

A multi-agent system is defined as a collection of autonomous agents which are able to interact with each other or with their environments to accomplish a task. The distributed control of MSNs is the main focus of recent research in this area. Examples of cooperative tasks include mobile sensor networks, distributed optimization, automated parallel delivery of payloads, region following formation control and coordinated path planning. One of the most important and well known task in multi-agent system is optimizing a cost function (utility function) in a distributed manner. In this task, the action of each agent affects the team cost function, which is unknown to individual agents. Here our goal is to design a control algorithm for individual selfish agents to optimize the team cost function. In this dissertation, two optimization problems are investigated, where two different methods of game theory and convex optimization are employed to solve these problems.

In our first problem, the coverage problem in an unknown environment by a Mobile Sensor Network (MSN) is studied. An algorithm based on game theory is proposed, where a collection of distributed agents communicate with local neighbors and use their local information to make

decisions.

In our second problem, a time-varying distributed convex optimization problem is studied for continuous-time multi-agent systems. The objective is to minimize the sum of local time-varying cost functions, each of which is known to only an individual agent, through local interaction. Here the optimal point is time varying and creates an optimal trajectory.

In last part of this dissertation, the distributed average tracking problem is addressed for a group of heterogeneous physical agents consisting of single-integrator, double-integrator and Euler-Lagrange dynamics. Here, the goal is that each agent uses local information and local interaction to calculate the average of individual time-varying reference inputs, one per agent. Dynamic average tracking is the main challenge in many other distributed algorithms, such as distributed optimization, and distributed Kalman filtering.

Contents

List of Figures

Chapter 1

Introduction

The objective of this dissertation is the study of collective behaviors in multi-agent systems. Control of multi-agent systems has received a growing interest from researchers during the last decades. Two approaches are commonly used for controlling the multi-agent systems: a centralized approach and a distributed approach. In the centralized approach, a central station collects all of the information from all agents to control the whole group of agents. While, in the distributed approach, local information is used by each agent to achieve the collective group behavior. In general, sending all information to a command center that could process the information is not feasible for most of the multi-agent system's tasks. Also, even if this were possible, the communication cost would be significantly large. Furthermore, the complexity of the overall system makes the problem of constructing a centralized optimal policy intractable.

Recently, there has been much interest in distributed control and coordination of multi-agent systems, where the goal is to collectively optimize a global cost function. This is motivated mainly by the emergence of large scale networks and new networking applications such as mobile ad hoc networks and mobile sensor networks. In this problem, the action of each agent affects the global cost function (team cost function), which is unknown to individual agents. Here our goal is to design a control algorithm for individual selfish agents to optimize the team cost function. In this dissertation, two optimization problems are investigated, where two different methods of game theory and convex optimization are employed to find optimal solutions for multi-agent systems' behaviors. Here, we address the following subjects which are listed in the order of appearance

1. Distributed Coverage Control of Mobile Sensor Networks in Unknown Environment Using Game Theory,

2. Distributed Continuous-Time Convex Optimization for Time-Varying Cost Functions.

Thus, we continue with a short introduction and overview of sensor coverage problem and distributed convex optimization problems.

1.1 Sensor Coverage

Improvement in mobile network techniques and expense reduction in individual sensing devices have created a great foundation for Mobile Sensor Networks (MSNs). There have been significant research activities and practical applications in this area [83]. Mobile sensors can cooperate to perform different challenging tasks such as surveillance, search and recovery operations, smart house and remote environment monitoring [75]. In the coverage problem, a group of mobile sensor agents seek to collect and process data to collectively cover an area. Limited communication capabilities, local and dynamic information, faulty components, and an uncertain environment make the problem very complicated.

1.1.1 Overview of Related Works

In most of the previous studies on the coverage problem, a probabilistic function is defined which represents the probability or occurrence frequency of events in a task area. Agents start from their initial random locations in the task area and search to find their best configuration for sensing more valuable regions. In [89], the virtual force, a combination of attractive and repulsive forces, has been utilized to determine the updated sensors' locations at each iteration. After finding an effective configuration, a one-time movement with energy consideration is conducted. In their study, the coverage worth of the area is assumed to be a uniform function where the agents just change their locations slightly in order to minimize the intersection of their covered area. In [40], an optimization problem is defined to maximize the sensors' coverage while taking into account the communication cost. Then a gradient decent algorithm is used to converge to the local solution of the optimization problem. Another perspective is used in [20] where the area is partitioned to Voronoi regions and each agent is just responsible for sensing its own region. Then each agent converges to the centroid of its region while at the same time the Voronoi partitions are adjusted using the worth of the area and neighbors' locations. In all the above mentioned studies, the distribution of the sensing worth in the area is known *a priori* by all agents. Unfortunately, this is not a realistic assumption in most applications.

It is desirable to introduce a method that enables the optimal coverage of an unknown environment. There are just a few works which assume that the agents do not have this prior knowledge. These studies assume that instead, each agent determines the worth of an area after sensing it. In [67], a solution similar to [20] is offered, where the summation of n distributions is assumed as an estimation model. The estimation algorithm can adjust just the weighting coefficients of these distributions. The designer chooses n, where by increasing n more accurate estimation can be obtained but the computational complexity is increased accordingly. An adaptive estimation algorithm is introduced where the agents use their observations to update the weighting coefficients of their model. Because the only optimization variables are the weighting coefficients, the designer has to use a large number of distributions with different means and covariances to improve the accuracy of the model. This estimation scheme does not perform well in real applications where the task area is large because n has to be increased proportional to the size of the mission area and hence a high computational burden is caused for the MSN.

The complexity and distributed nature of the coverage problem, motivate the use of a game theory method. One approach for solving the coverage problem is to model the problem as a non-cooperative game, where the players (mobile sensors here) independently pursue their own objectives. The game theory method has many other advantages including robustness to failures and environmental disturbances, reducing communication requirements and improving scalability. The primary goal of game theory-based approaches is to design rules that guarantee the existence and efficiency of a pure Nash equilibrium while investigating some other properties including locality and informational dependencies [50]. The coverage problem can be modelled as a resource allocation problem. Proper utility functions and reinforcement learning methods are designed for the resource allocation of MSNs in [49, 50]. In these algorithms, each player must have access to the utility values of its alternative actions. However, in an unknown environment such an assumption is infeasible due to the lack of the knowledge of the area's worth. Such information constraints are taken into account in [32,88]. Different learning algorithms are defined where the agents experiment alternative actions without pre-evaluation of these actions' values. These learning algorithms can be interpreted as a trial-and-error approach for the agents. However, such trial-and-error approach results in unnecessary movements of the agents and hence increases the energy consumption. It also decreases the rate of convergence and causes chattering in applications.

1.2 Distributed Convex Optimization

The distributed optimization problem has attracted a significant attention recently. It arises in many applications of multi-agent systems, where agents cooperate in order to accomplish various tasks as a team in a distributed and optimal fashion. We are interested in a class of distributed convex optimization problems, where the goal is to minimize the sum of local cost functions, each of which is known to only an individual agent.

1.2.1 Overview of Related Works

The incremental subgradient algorithm is introduced as one of the earlier approaches addressing this problem [60, 65]. In this algorithm an estimate of the optimal point is passed through the network while each agent makes a small adjustment on it. Recently some significant results based on the combination of consensus and subgradient algorithms have been published [35,53,84]. For example, this combination is used in [35] for solving the coupled optimization problems with a fixed undirected graph. A projected subgradient algorithm is proposed in [53], where each agent is required to lie in its own convex set. It is shown that all agents can reach an optimal point in the intersection of all agents' convex sets even for a time-varying communication graph with doubly stochastic edge weight matrices.

However, all the aforementioned works are based on discrete-time algorithms. Recently, some new research is conducted on distributed optimization problems for multi-agent systems with continuous-time dynamics. Such a scheme has applications in motion coordination of multi-agent systems. For example, multiple physical vehicles modelled by continuous-time dynamics might need to rendezvous at a team optimal location. In [46], a generalized class of zero-gradient sum controllers is introduced for twice differentiable strongly convex functions under an undirected graph. In [70], a continuous-time version of [53] for directed and undirected graphs is studied, where it is assumed that each agent is aware of the convex optimal solution set of its own cost function and the intersection of all these sets is nonempty. Article [38] derives an explicit expression for the convergence rate and ultimate error bounds of a continuous-time distributed optimization algorithm. In [79], a general approach is given to address the problem of distributed convex optimization with equality and inequality constraints. A proportional-integral algorithm is introduced in [23, 31, 37], where [31] considers strongly connected weight balanced directed graphs and [37] extends these results using discrete-time communication updates. A distributed optimization for single-integrator agents is studied in [42] with the adaptivity and finite-time convergence properties.

In continuous-time optimization problems, the agents are usually assumed to have single-integrator dynamics. However, a broad class of vehicles requires double-integrator dynamic models. In addition, having time-invariant cost functions is a common assumption in the literature. However, in many applications the local cost functions are time varying, reflecting the fact that the optimal point could be changing over time and creates a trajectory. There are just a few works in the literature addressing the distributed optimization problem with time-varying cost functions, where in those works there exist bounded errors converging to the optimal trajectory. For example, the economic dispatch problem for a network of power generating units is studied in [16], where it is proved that the algorithm is robust to slowly time-varying loads. In particular, it is shown that for time-varying loads with bounded first and second derivatives the optimization error will remain bounded. In [71], a distributed time-varying stochastic optimization problem is considered, where it is assumed that the cost functions are strongly convex, with Lipschitz continuous gradients. It is proved that under the persistent excitation assumption, a bounded error in expectation will be achieved asymptotically. In [43], a distributed discrete-time algorithm based on the alternating direction method of multipliers (ADMM) is introduced to optimize a time-varying cost function.

Furthermore, in all articles on distributed optimization mentioned above, the agents will eventually approach a common optimal point while in some applications it is desirable to achieve swarm behavior. The goal of flocking or swarming with a leader is that a group of agents tracks a leader with only local interaction while maintaining connectivity and avoiding inter-agent collision [8, 21, 56, 76]. Swarm tracking algorithms are studied in [56] and [76], where it is assumed that the leader is a neighbor of all followers and has a constant and time-varying velocity, respectively. In [8], swarm tracking algorithms via a variable structure approach are introduced, where the leader is a neighbor of only a subset of the followers. In the aforementioned studies, the leader plans the trajectory for the team and the agents are not directly assigned to complete a task cooperatively. In [78], the agents are assigned a task to estimate a stationary field while exhibiting cohesive motions. Although optimizing a certain team criterion while performing the swarm behavior is a highly motivated task in many multi-agent applications, it has not been addressed in the literature.

1.3 Distributed Average Tracking

In many applications of multi-agent systems, agents are required to compute the summation of individual time-varying inputs in a distributed manner. For example, in sensor fusion [57], feature-based map merging [1], distributed Kalman filtering [2], and distributed optimization [62], computing the average of individual reference inputs is an inseparable part of the algorithms and hence this problem attracted a significant attention recently.

In this chapter, an average tracking problem for a team of heterogeneous agents is studied, where each agent uses local information to calculate the average of individual time-varying reference inputs, one per agent. Here, the average of individual reference inputs is time-varying and it is not available to any agent; hence distributed average tracking introduces additional complexities and theoretical challenges compared to the consensus and leader-followers problems.

1.3.1 Overview of Related Works

Researchers have introduced linear distributed algorithms as one of the earlier approaches addressing this problem [3, 27, 36, 73]. In [27], a proportional-integral algorithm is proposed to achieve distributed average tracking for slowly-varying reference inputs with a bounded tracking error. In [3], through the use of the internal model principle, an algorithm is introduced for a special group of time-varying reference inputs with a common denominator in their Laplace transforms. In [36], a distributed average tracking problem is solved, with steady-state errors, while the privacy of each agent's input is preserved.

However, in linear algorithms, the reference inputs are required to satisfy restrictive constraints and most of the results only can guarantee to have a bounded error. Therefore, some results based on nonlinear tracking algorithms have been published recently [12, 54]. A class of nonlinear algorithms is introduced in [54], where it is proved that for reference inputs with bounded deviations the tracking error is bounded. In [12], a nonsmooth algorithm is proposed for reference inputs with bounded derivatives.

However, all the aforementioned studies addressed the distributed average tracking problem from an estimation perspective, where the agents do not have a certain physical dynamics. There are various applications, where the distributed average tracking problem is employed as a control law for physical agents [9]. For example, multiple agents moving in a formation with local information and interaction might need to cooperatively figure out what optimal trajectory the virtual leader or center of the team should follow, where each individual agent specifies its motion using that knowledge. Distributed average tracking can be employed in this problem, where each agent can construct its own reference input using the gradient of its own local cost function [62]. A distributed average tracking algorithm is proposed in [15], for physical agents with double-integrator dynamics, where the reference inputs are allowed to have a bounded accelerations. A distributed algorithm without using velocity measurements for a group of physical second-order agents is introduced in [30], where the reference input are assumed to have bounded accelerations' deviations.

However, in real applications physical agents might have more complicated dynamics rather than single-integrator or double-integrator dynamics. There are only a few studies, that have addressed more complicated dynamics. For example, in [85], the problem is studied for physical agents with general linear dynamics, where reference inputs are bounded. A class of algorithms is proposed in [14], to achieve distributed average tracking for physical Euler-Lagrange systems, where it is proved that a bounded error is achieved for reference inputs with bounded derivatives.

In [29], a distributed average tracking algorithm is proposed for physical second-order agents, where there is a nonlinear term in both agents' and reference inputs' dynamics.

In most of the studies in the literature, agents are assumed to be identical. There are only few works assumed nonidentical parameters or nonidentical additive terms in agents' dynamics [14, 29]. However, in real applications, we might need to employ different agents (robots) with different abilities to accomplish a task. In these scenarios, agents obey completely different physical dynamics. To the best of our knowledge, the heterogeneous average tracking problem in the literature has been limited to the case that the reference inputs are time-invariant, where the problem is transformed into a distributed consensus [44, 52, 87]. In heterogeneous distributed consensus algorithms, there always exists a term forcing the velocity of each individual agent to zero. This tremendously reduces the complexity of the problem. However, in dynamic average tracking problem, our goal is to track a time-varying trajectory, where a precise control on velocities and accelerations of the agents are required. It is worthwhile to mention that having a heterogeneous multi-agent system consisting of agents with different dynamics, it is not possible to employ the algorithms proposed for homogeneous dynamics, corresponding to each agent's dynamic, and expect to have a well-behaved system. Therefore, a careful analysis considering the interaction among the agents with different dynamics is needed.

1.4 Contributions of Dissertation

In this dissertation, we focus on the following two problems for multi-agent systems:

1. Sensor coverage problem in an unknown environment using a game theory method,

2. Distributed convex optimization of time-varying cost functions.

Some materials from this dissertation have been published in three conferences [61, 63, 64]. In this subsection, we briefly talk about the contributions of this dissertation as follow.

- In the first part of the dissertation, we focus on sensor coverage problem in an unknown environment using a game theory method. We can categorize our results into three main parts:

  – In Section 2.2, the game theory control is used to design a utility function based on each agent's marginal sensing contribution and the energy consumption of the agent caused by communication, sensing and movement. By employing properly designed utility functions, each agent uses only local information and local communication to complete its tasks in a distributed manner. It is shown that the game properly defined using these utility functions is a state-based potential game, which is guaranteed to have at least one Nash equilibrium. In addition, the binary log-linear learning, which is one of the best fit for this application, is used for updating the agents' actions. Using the above design, the agents will converge stochastically to a Nash equilibrium. Considering all energy consumption by using a simple state-based utility function is valuable as no large computational burden is created. However, this game and learning design just works when the agents have a prior knowledge about the worth of the area.

- In Section 2.3, an estimation scheme is defined to overcome the lack of prior knowledge of the worth of the area in an unknown environment. A Gaussian Mixture Model (GMM) is introduced, where each agent uses its previous observations to estimate the unknown model parameters. An expectation-maximization algorithm is used as a tool to solve the ML method and calculate the unknown GMM parameters. By estimating the area's worth, the algorithm defined in Section 2.2 is able to converge to a Nash equilibrium stochastically. Particularly, using the agent's previous observations, their future actions will be chosen wisely so as to reduce the improper effect of the trial-and-error method in the literature.

- In Section 2.4, the entropy criterion is further utilized to select informative observations. The agents will consider the mutual information between their observed and unobserved regions in order to find more informative observations. It will be shown that the modified game with the addition of a proper term corresponding to the mutual information to the utility function will remain a state-based potential game. By taking the mutual information into account in the utility function design, the convergence rate will be noticeably increased.

• In the second part of the dissertation, we focus on the distributed convex optimization of time-varying cost functions for continuous-time dynamics.

The introduced framework, distributed continuous-time time-varying optimization, is of great significance in motion coordination. Here, multiple agents cooperatively achieve motion coordination while optimizing a team objective function with only local information and interaction. For example, multiple spacecraft might need to dock at a location distributively with only local information and interaction such that the total team performance is optimized. Multiple agents might need to maintain a geometric configuration or a swarm centered at a location closest to all regions observed by the agents distributively with only local information and interaction. Multiple agents moving in a formation or swarm with local information and interaction might need to cooperatively figure out what optimal trajectory the virtual leader or center of the team should follow and that knowledge would help the individual agents specify their motions. Furthermore, there is a significant need to use distributed optimization in various applications such as economic dispatch, internet congestion control, and home automation with smart electrical devices. While the studies in the aforementioned applications would be simplified by assuming that the changing rate of the cost functions or the constraints, is small and hence treated as invariant in each time interval, it might be more realistic and relevant to explicitly take into account the time-varying nature of the cost functions or constraints. As a result, distributed continuous-time optimization algorithms with time-varying cost functions or constraints might serve as continuous-time solvers to figure out the optimal trajectory in these applications.

Here, we are faced with several challenges such as:

- Having time-varying cost functions, which generally changes the problem from finding the fixed optimal point into tracking the optimal trajectory.
- Solving the problem in a distributed manner using only local information and local interaction.
- Solving the problem for continuous-time single-integrator and double-integrator dynamics, where in the latter case we have only direct control on agents' accelerations.

- In our algorithms, the signum function is employed to compensate for the effect of the inconsistent internal time-varying optimization signals among the agents so that the agents can reach consensus. As the signum function might cause chattering in some applications, it is replaced with continuous approximations in some algorithms but additional challenges in analysis would result from the replacement.
- Providing analysis on optimization error bounds in scenarios where the agents' states can not achieve consensus.
- The coexistence of the optimization objective and the inherent nonlinearity of the swarm tracking behavior.

- In the last part of this dissertation, a heterogeneous framework consisting of agents with three different dynamics, single-integrator, double-integrator and Euler-Lagrange dynamics, is considered. Two nonsmooth algorithms are proposed to achieve the distributed average tracking goal.
  - In our first proposed algorithm, each agent is required to have access to only its own position and the relative positions between itself and its neighbors. In some applications, the relative positions can be obtained by using only agents' local sensing capabilities, which might in turn eliminate the communication necessity between agents.
  - To relax some restrictive assumptions on admissible reference inputs, we propose an estimator-based algorithm, where a filter is introduced for each agent to generate an estimation of the average of the reference inputs. Then, each agent tracks its own generated signal to accomplish the average tracking task. In both algorithms, agents described by Euler-Lagrange dynamics, place a restrictive assumption on the admissible reference

inputs.

The advantage of the second algorithm will be more substantial for a mutli-agent system consisting of agents with only single-integrator and double-integrator dynamics. In such a framework, using estimator-based algorithm, the heterogeneous dynamic average tracking goal is achieved, where there is no restriction on reference inputs. As a trade-off, the estimator based algorithm necessitates communication between neighbors, where each agent must communicate its own filter's variables with its neighbors.

Finally, it is shown that how the proposed distributed average tracking algorithms can be employed to solve the distributed optimization problem for a heterogeneous framework. It is shown that the heterogeneous multi-agent system is able to solve the the distributed optimization problem if the velocity of the reference input for each agent is defined and equal to the gradient of its own cost function.

1.5 Preliminaries

In the reminder of this chapter, we introduce notations used in the dissertation and algebraic graph theory.

1.5.1 Notations

R set of real numbers

R^+ set of postive real numbers

R^p set of $p \times 1$ real vectors

$R^{m \times n}$ set of $m \times n$ real matirces

$\mathbf{1}_n$ $n \times 1$ column vector of all ones

$\mathbf{0}_n$ $n \times 1$ column vector of all zeros

I_n $n \times n$ identity matrix

$|S|$ cardinality of set S

$\lambda_{\min}(\cdot)$ minimum eigenvalue of a square real matrix with real eigenvalues

$\lambda_{\max}(\cdot)$ maximum eigenvalue of a square real matrix with real eigenvalues

$\text{diag}(z_1, \ldots, z_p)$ diagonal matrix with diagonal entries z_1 to z_p

$A > 0$ a positive matrix A

$A \geq 0$ a nonnegative matrix A

$A > B$ $A - B$ is positive definite

$A \geq B$ $A - B$ is nonnegative definite

$\| \cdot \|_p$ p-norm of a real vector

$\otimes$ Kronecker product

$\text{sgn}(\cdot)$ signum function defined component-wise

1.5.2 Graph Theory

Let a triplet $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ be an undirected graph, where $\mathcal{V} = \{1, ..., N\}$ is the node set and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set, and $\mathcal{A} = [a_{ij}]$ is the adjacency matrix. An edge between agents i and j, denoted by $e = (i, j) \in \mathcal{E}$, means that they can obtain information from each other. In an undirected graph the edges (i, j) and (j, i) are equivalent. We assume $(i, i) \notin \mathcal{E}$. The adjacency matrix $\mathcal{A}$ is defined as $a_{ij} = a_{ji} = 1$, if $(i, j) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise. The set of neighbors of agent i is denoted by $N_i = \{j \in \mathcal{V} : (j, i) \in \mathcal{E}\}$. A sequence of edges of the form $(i, j), (j, k), ...$, where $i, j, k \in \mathcal{V}$, is a path. The graph $\mathcal{G}$ is connected, if there is a path from every node to every other node. By arbitrarily assigning an orientation for the edges in $\mathcal{G}$, let $D \triangleq [d_{ik}] \in \mathbb{R}^{N \times |\mathcal{E}|}$ be the *incidence matrix* associated with $\mathcal{G}$, where $d_{ik} = -1$ if the edge e_k leaves node i, $d_{ik} = 1$ if it enters node i, and $d_{ik} = 0$ otherwise.

Let the Laplacian matrix $L = [l_{ij}] \in \mathbb{R}^{N \times N}$ associated with the graph $\mathcal{G}$ be defined as $l_{ii} = \sum_{j=1, j \neq i}^{N} a_{ij}$ and $l_{ij} = -a_{ij}$ for $i \neq j$. Note that $L \triangleq DD^T$. The Laplacian matrix L is symmetric positive semidefinite. The undirected graph $\mathcal{G}$ is connected if and only if L has a simple zero eigenvalue with the corresponding eigenvector $\mathbf{1}_N$ and all other eigenvalues are positive [17]. When the graph $\mathcal{G}$ is connected, we order the eigenvalues of L as $\lambda_1[L] = 0 < \lambda_2[L] \leq ... \leq \lambda_N[L]$. Particularly, $\lambda_2[L]$ is the second smallest eigenvalue of the Laplacian matrix L.

The above notations can also be adopted for time-varying graphs, where $\mathcal{G}(t), \mathcal{A}(t), D(t)$ and $L(t)$ are, respectively, the undirected graph, the adjacency matrix, the incidence matrix and the Laplacian matrix at time t. For time-varying graphs $\mathcal{G}(t)$, $\lambda_i[L(t)], \forall i \in \mathcal{I}$, is a function of t. As long as $\mathcal{G}(t)$ is connected, $\lambda_2[L(t)]$ is uniformly lower bounded above 0 because there is only a finite number of possible $L(t)$ associated with $\mathcal{G}(t)$.

Lemma 1 *[55] The second smallest eigenvalue $\lambda_2[L]$ of the Laplacian matrix L associated with the undirected connected graph $\mathcal{G}$ satisfies $\lambda_2[L] = \min_{x^T \boldsymbol{1}_N = 0, x \neq \boldsymbol{0}_N} \frac{x^T L x}{x^T x}$.*

Chapter 2

Distributed Sensor Coverage of Mobile Sensor Networks Using Game Theory

2.1 Sensor Coverage

An MSN is established by a distributed collection of agents, where each of them has communication, sensing, moving, and computation capabilities. In this dissertation, a class of MSN applications is considered, where a limited number of sensors are randomly deployed in a task area. The area's worth is different at different locations. The sensors' task is to change their locations in order to maximize the total worth of the covered area at their final configuration while minimizing the total energy consumption caused by communication, sensing and movement.

The area's worth can be interpreted as the probability of finding a target by sensors in that area. The agents do not have prior knowledge about the worth. However, the worth of a location will be determined by an agent after sensing it. Precisely, we aim at solving a coverage optimization problem in an unknown environment by formulating it as a non-cooperative game, where each sensor benefits from sensing the environment as a member of a group with local information and communication.

A convex two-dimensional mission space, discretized into a (squared) lattice is considered. Each square of the lattice has unit dimensions and is labeled with the coordinate of its center $q = (q_x, q_y)$. The collection of all squares of the lattice is denoted by Q. A numerical variable $f_q \geq 0$ is assigned to the worth or the probability of the occurrence of an event in each square with center $q \in Q$. The larger the worth f_q, the more important the sensing of the square with center q. Later, the value f_q will be recognized as a benefit of observing the point q. Here, f_q is assumed to be stationary. However, the result of this dissertation can be extended to a non-stationary field.

There are N mobile sensor agents in the field where the location of agent, $i = 1, ..., N$, at time step t is denoted by $a_i^p(t) \triangleq (x_i(t), y_i(t)) \in Q$. The sensing region of agent i, at time step t is modelled as a disc with the center $a_i^p(t)$ and the radius $a_i^r(t)$. Each agent i chooses the radius $a_i^r(t)$ from a discrete set with the minimum and the maximum equal to r_{min}^i and r_{max}^i, respectively. The set of acceptable radii is not necessarily identical for different agents, but as it will be discussed in Subsection 2.2.1, it is assumed that all agents know the largest possible sensing range $r_{\text{max}} := \max r_{\text{max}}^i$. The agent's action at time step t is denoted by $a_i(t)$, where $a_i(t) := (a_i^p(t), a_i^r(t)) \in A_i(t)$ and $A_i(t)$ is the available action set for agent i, at time step t. Also let $S(a_i(t))$ be the area covered by agent i at time step t. The action profile of all agents is denoted by $a(t) = (a_1(t), ..., a_N(t)) \in A(t) := \prod_{i=1}^{N} A_i(t)$. The motion of the agents will be limited to their adjacent lattices if there is no obstacle there. We use $C_i(a_i(t-1))$ to denote the available actions for agent i at time step t, which is a function of agent i's action at time step $t-1$. Each

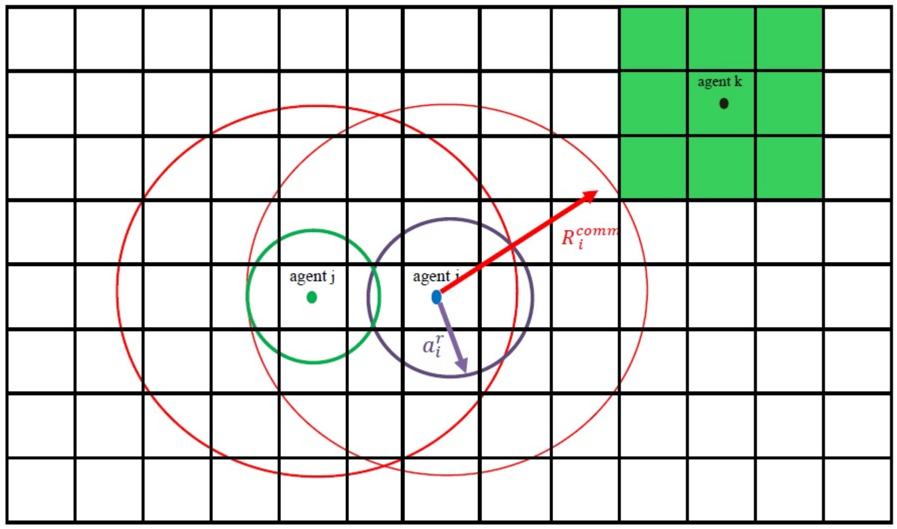

Figure 2.1: The mission space and sensor model. The green shaded squares denote the feasible actions for agent k.

agent is able to communicate with its neighbors to exchange information. The set of neighbors of agent i is given by $N_i^{\text{comm}}(a(t)) := \{j = 1, ..., N| \left\| a_i^p(t) - a_j^p(t) \right\|_2 \leq R_i^{\text{comm}}\}$, where R_i^{comm} is the communication range of agent i. The mission space and sensor model are illustrated in Fig. 2.1.

2.2 Game Design

In this section, we design a state-based potential game to relocate the agents from their random initial positions to a valuable configuration by taking into account both the worth of the covered area and the energy consumption. The binary log-linear learning is employed to update the agents' actions. It is shown that the agents converge stochastically to a Nash equilibrium.

2.2.1 Utility Design

Mobile sensors consume energy for their communication, sensing and movement. Due to energy limitations, taking the energy consumption into account in the utility design will improve the performance of the MSNs. We first consider the energy consumption of the agents due to communication with neighbors. To select the communication ranges, we require R_i^{comm} is larger than or equal to $2r_{\text{max}}$, because if R_i^{comm} is assumed less than $2r_{\text{max}}$, there might be a region sensed by two agents while they can not communicate with each other. That is the reason it is assumed that all the agents have the same r_{max}. To decrease the energy usage, the communication range and the amount of data transfer should be reduced. Hence, we select $R_i^{\text{comm}} = 2r_{\text{max}}, i = 1, ..., N$. In the following, R^{comm} denotes the communication range for each agent. In addition, each agent just sends its own action $a_i(t)$ to its neighbors, which does not need significant communication between agents. As mentioned, the communication range, R^{comm}, and the amount of data transfer cannot be further reduced in our problem set-up. Thus there is no need to incorporate the communication energy into our optimization cost function.

We then consider the energy consumption caused by sensing. There is a trade-off between the power usage and the size of the covered region. The energy consumption of agent i due to sensing is defined as

$$E_i^{\text{sense}}(a_i(t)) = K_i(a_i^r(t))^2,$$

where $K_i > 0$ is a coefficient. Increasing the radius will increase the energy consumption. Hence $a_i^r(t)$ is used as an optimization variable, where each agent attempts to optimize its own power usage by finding a proper radius. In [58], a coverage algorithm for vision-based sensors is proposed to turn off the sensors with overlapping fields of view. Here by letting $r_{\text{min}} = 0$, our proposed method is able to save energy in such a situation. Agent i can choose $a_i^r = 0$ when the agent cannot improve the sensing performance of the MSN.

We finally consider the energy consumption caused by movement. The energy consumption of agent i due to movement is defined as

$$E_i^{\text{move}}(a_i(t), z_i(t)) = K_i'(|a_i^p(t) - z_i(t)|), \tag{2.1}$$

where $K_i' > 0$ is a coefficient and $z_i(t) \triangleq a_i^p(t-1)$ is the previous location of the agent i. As it is seen in (2.1), the movement energy consumption, and the utility to be designed next as a result, is a function of the *state* (the agent's previous location in this case). Here the term state has a different meaning in the context of game theory in comparison with that in control theory. In our problem the term state denotes the previous location of the agent. This is the reason that a state-based potential game will be designed for this problem.

We now proceed to formulate our coverage optimization problem as a state-based game. A utility function U_i is designed for each agent that aims to capture the trade-off between the worth of the covered area and the energy consumption by agent i. The utility function for agent i is designed as

$$\begin{aligned} U_i(a(t), z(t)) &= F(a_i(t), a_{-i}(t)) - F(a_i^0(t), a_{-i}(t)) \\ &\quad - E_i^{\text{sense}}(a_i(t)) - E_i^{\text{move}}(a_i(t), z_i(t)) \\ F(a(t)) &= \sum_{q \in \bigcup\limits_{i=1}^{N} S(a_i(t)) \bigcap Q} f_q, \end{aligned} \tag{2.2}$$

where $F(a(t))$ denotes the worth of the covered area by the agents, $a_i^0(t)$ is the null action agent i (i.e. there is no agent i or equivalently agent i is turned off), and $a_{-i}(t)$ is the actions of all agents other than agent i. Here $F(a_i(t), a_{-i}(t)) - F(a_i^0(t), a_{-i}(t))$ is agent i's marginal contribution to sense the area. Sometimes the marginal contribution design is refereed as Wonderful Life Utility (WLU) [82]. Note that the utility function U_i is local over the sensed region by agent i. Here U_i is dependent only on the actions of $\{i\} \cup N_i^{\text{sense}}(a(t))$, where $N_i^{\text{sense}}(a(t))$ is the set of all agents whose sensing regions have an intersection with that of agent i. As mentioned before, by setting $R^{\text{comm}} = 2r_{\text{max}}$, if $j \in N_i^{\text{sense}}(a(t))$, it follows that $j \in N_i^{\text{comm}}(a(t))$. That is, when the agents have a sensing intersection, they can communicate with each other. The definition of the marginal contribution term shows that the effect of the agents without a sensing intersection is cancelled. Because of this cancellation, there is no need for an agent to know the actions of the agents that do not have a sensing intersection with it. This makes the defined utility function local. After defining our game ingredients, now a state-based game will be introduced. The following lemma will show that our defined game is a state-based potential game.

Lemma 2 *The coverage state-based game* $\Upsilon := \langle N, A, U \rangle$ *, where* $U = \{U_j, j = 1, ..., N\}$ *with* U_j *given by* (2.2)*, is an exact state-based potential game with the potential function*

$$\Phi_1(a(t), z(t)) = F(a(t)) - \sum_{j=1}^{N} \big(E_j^{\textit{sense}}(a_j(t)) + E_j^{\textit{move}}(a_j(t), z_j(t))\big), \tag{2.3}$$

where $z(t) = (z_1(t), ..., z_N(t))$ *is the current state of the game.*

Proof:

As shown in [47], a state-based potential game has to satisfy two conditions:

1) For any agent $i = 1, ..., N$ and action $a_i'(t) \in A_i$

$$\Phi_1\big(a_i'(t), a_{-i}(t), z(t)\big) - \Phi_1\big(a(t), z(t)\big) = U_i\big(a_i'(t), a_{-i}(t), z(t)\big) - U_i\big(a(t), z(t)\big), \tag{2.4}$$

2) For any state $z'(t)$ in the support of $(a(t), z(t))$, the inequality $\Phi_1(a(t), z'(t)) \geq \Phi_1(a(t), z(t))$ holds.

We first verify condition 1. According to (2.2) and (2.3), we have

$$
\begin{aligned}
&\Phi_1(a_i'(t), a_{-i}(t), z(t)) - \Phi_1(a(t), z(t)) \\
&\quad = F(a_i'(t), a_{-i}(t)) + E_i^{\text{sense}}(a_i'(t)) + E_i^{\text{move}}(a_i'(t), z_i(t)) \\
&\quad\ \ - F(a(t)) - E_i^{\text{sense}}(a_i(t)) - E_i^{\text{move}}(a_i(t), z_i(t)) \\
&\quad = U_i(a_i'(t), a_{-i}(t), z(t)) - U_i(a(t), z(t)),
\end{aligned}
\tag{2.5}
$$

which shows that (2.4) holds.

We next verify condition 2. The fact that $z'(t)$ is in the support of $(a(t), z(t))$ implies that $z'(t) = a^p(t)$. Thus we have

$$
\Phi_1(a(t), z'(t)) - \Phi_1(a(t), z(t)) = -E_i^{\text{move}}(a_i(t), z_i'(t)) + E_i^{\text{move}}(a_i(t), z_i(t)) \geq 0,
$$

where we used the fact that $E_i^{\text{move}}(a_i(t), z_i'(t)) = 0$. As a result, it follows that condition 2 is satisfied. ■

2.2.2 Reinforcement Learning

Converging to a solution such as a Nash equilibrium requires distributed adaptation rules. The motivation for using these adaptation rules is that each agent can maximize its own utility function using these rules. Game theoretic reinforcement learning provides a starting point for the construction of iterative algorithms to reach a Nash equilibrium [28], [66].

In the early literature of reinforcement learning, forecasted best-response dynamics have been used, where players choose a best response to the forecasted strategies of other players. Log-linear learning, is introduced in [5], where only a single player updates its action at each iteration. In the log-linear learning, agents are allowed to explore. The agents can select nonoptimal actions but with relatively low probabilities. This plays an important role for agents to escape the suboptimal actions, which increases the probability of finding a better Nash equilibrium.

The log-linear learning, introduced in [5], assumes that the agents have a constant action set. However, as mentioned in Section 2.1 the feasible actions available to an agent are restricted by the agent's state in the game and hence the action set is varying. Binary log-linear learning is a modified version of the log-linear learning for potential games in which varying available action sets can be used by the agents. In [48] the binary log-linear method has been analyzed. Theorem 5.1 in [48], shows that a potential game will converge to stochastically stable actions. These actions are the set of potential maximizers if all agents adhere to the binary log-linear learning where the following assumptions should be satisfied on the agents' available sets.

1) Feasibility: For any agent $i = 1, ..., N$ and any action pair $a_i(0), a_i(m) \in A_i$, there exists a sequence of actions from $a_i(0)$ to $a_i(m)$ satisfying $a_i(t) \in C_i(a_i(t-1))$ for all $t \in 1, 2, ..., m$.

2) Reversibility: For any agent $i = 1, ..., N$ and any action pair $a_i', a_i'' \in A_i, a_i' \in C_i(a_i'') \longleftrightarrow a_i'' \in C_i(a_i')$.

It is easy to check that in our problem set-up the above assumptions are satisfied. In this study, the binary log-linear learning method is employed to update each agent's actions. At each time t, one agent i is randomly selected and allowed to alter its action while all other agents repeat their actions, i.e., $a_{-i}(t) = a_{-i}(t-1)$. The selected agent i chooses a trial action $a_i'(t)$ uniformly randomly from the available action set $C_i(a_i(t-1))$. The player calculates the utility function for this trial action. Then agent i randomizes its action according to

$$\begin{aligned}
P_i^{(a_i(t-1),a_{-i}(t-1))}(t) = & \quad \frac{exp(\frac{1}{\tau}U_i(a(t-1),z(t)))}{exp(\frac{1}{\tau}U_i(a(t-1),z(t)))+exp(\frac{1}{\tau}U(a_i'(t),a_{-i}(t-1),z(t)))}, \\
P_i^{(a_i'(t),a_{-i}(t-1))}(t) = & \quad \frac{exp(\frac{1}{\tau}U_i(a_i'(t),a_{-i}(t-1),z(t)))}{exp(\frac{1}{\tau}U_i(a(t-1),z(t)))+exp(\frac{1}{\tau}U_i(a_i'(t),a_{-i}(t-1),z(t)))}, \\
P_i^{(a_i(t),a_{-i}(t-1))}(t) = & \quad 0, \ \ \forall a_i(t) \neq a_i'(t), a_i(t-1).
\end{aligned} \tag{2.6}$$

where $a_i'(t)$ is the alternative action for agent i and $P_i^{(\star,a_{-i}(t-1))}(t)$ denotes the probability of choosing action $\star$ at time t. In other words, the agent uses a distribution to randomly select between its previous action, $a_i(t-1)$, and an alternative action $a_i'(t)$. The coefficient $\tau > 0$ determines how likely agent i is to choose a suboptimal action. For $\tau \to \infty$, the learning algorithm will choose the action $a_i(t-1)$ or $a_i'(t)$ with an equal probability while for $\tau \to 0$, it will choose the action which has the greatest utility function among the set $a_i(t-1)$ and $a_i'(t)$.

2.3 Gaussian Mixture Model Estimation

In this section, we introduce an estimation scheme to overcome the limitations discussed in Section 1.3. To apply the binary log-linear method (2.6), the agents have to pre-calculate the utility function of the alternative action $a_i'(t)$. However, the lack of the prior knowledge about the worth of the area makes this impractical. Here, the issue will be resolved by using the information that the agents collect after sensing a region. In our new set-up, the agents will keep the worth of the sensed regions in their memories. This information will be used to estimate the worth of the uncovered regions. The GMM is chosen as a distribution model of the worth of the area. The GMM is a parametric probability density function represented as a weighted sum of Gaussian component densities. The GMM is commonly used as a probability distribution model because it is one of the most frequently observed and is a starting point for modeling many natural processes. The GMM is defined as

$$\sum_{k=1}^{M} w_k g(x|\mu_k, \Sigma_k), \tag{2.7}$$

and

$$g(x|\mu_k, \Sigma_k) = \frac{1}{2\pi|\Sigma_k|^{\frac{1}{2}}} \; exp\,[-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)], \tag{2.8}$$

where x is a 2-dimensional location data vector, M is the number of Gaussian functions in the GMM, w_k is the kth weighting coefficient, $g(x|\mu_k, \Sigma_k)$ is a 2-dimensional Gaussian function with mean vector μ_k and covariance matrix $\Sigma_k \in R^{2\times 2}$ and $|\Sigma_k|$ denotes the determinant of the covariance matrix. The weights have to satisfy the constraint that $\sum_{k=1}^{M} w_k = 1$.

In this study, the GMM is parametrized by $3M$ components given by $\lambda = \{w_j, \mu_j, \Sigma_j | j = 1, 2, .., M\}$, where w_j, μ_j and Σ_j are the weighting coefficient, mean and covariance corresponding to the jth Gaussian of the GMM, respectively. There are several techniques available for estimating λ [51]. One of the most accurate and well-established method is the Maximum Likelihood (ML) estimation. Agent i has its own estimate of λ denote by $\hat{\lambda}^i = \{\hat{w}_j^i, \hat{\mu}_j^i, \hat{\Sigma}_j^i | j = 1, 2, .., M\}$, where $\hat{w}_j^i, \hat{\mu}_j^i$ and $\hat{\Sigma}_j^i$ are the estimates of w_j, μ_j and Σ_j by agent i, respectively. Agent i has the observation $O_t^i \triangleq S(a_i(t))$, at time step t. The sequence of these observations, from the step 1 to t, will be used as a training set $O_{1:t}^i = \{O_1^i, O_2^i, ..., O_t^i\}$. Given the training vector $O_{1:t}^i$, we wish to find the estimate $\hat{\lambda}^i$ that maximizes the probability of the occurrence of the observation sequence set $O_{1:t}^i$ denoted by $p(O_{1:t}^i|\hat{\lambda}^i)$, The ML estimation can be written as

$$\max_{\hat{\lambda}^i} p(O_{1:t}^i|\hat{\lambda}^i) = \max_{\hat{\lambda}^i} \prod_{t'=1}^{t'=t} p(O_{t'}^i|\hat{\lambda}^i). \tag{2.9}$$

Although (2.9) is a nonlinear function of the parameter $\hat{\lambda}^i$ and direct maximization is not possible, the expectation-maximization (EM) has been introduced in [22] to obtain an iterative algorithm.

Using the EM approach, the GMM parameters can be estimated as

$$
\begin{aligned}
\hat{w}_j^i(t) = & \quad \frac{1}{T}\sum_{t'=1}^{t'=t}\sum_{k=1}^{k=|O_{t'}^i|} Pr^i(j|{}^k o_{t'}^i, \hat{\lambda}^i), \\
\hat{\mu}_j^i(t) = & \quad \frac{\sum_{t'=1}^{t'=t}\sum_{k=1}^{k=|O_{t'}^i|} Pr^i(j|{}^k o_{t'}^i, \hat{\lambda}^i)\,{}^k o_{t'}^i}{\sum_{t'=1}^{t'=t}\sum_{k=1}^{k=|O_{t'}^i|} Pr^i(j|{}^k o_{t'}^i, \hat{\lambda}^i)}, \\
\hat{\Sigma}_j^{i\,2}(t) = & \quad \frac{\sum_{t'=1}^{t'=t}\sum_{k=1}^{k=|O_{t'}^i|} Pr^i(j|{}^k o_{t'}^i, \hat{\lambda}^i)({}^k o_{t'}^i - \hat{\mu}_j^i)({}^k o_{t'}^i - \hat{\mu}_j^i)^T}{\sum_{t'=1}^{t'=t}\sum_{k=1}^{k=|O_{t'}^i|} Pr^i(j|{}^k o_{t'}^i, \hat{\lambda}^i)}, \\
Pr^i(j|{}^k o_t^i, \hat{\lambda}^i) = & \quad \frac{\hat{w}_j^i g({}^k o_t^i|\hat{\mu}_j^i, \hat{\Sigma}_j^i)}{\sum_{k=1}^{M} \hat{w}_k^i g({}^k o_t^i|\hat{\mu}_k^i, \hat{\Sigma}_k^i)},
\end{aligned}
\tag{2.10}
$$

for all observations ${}^k o_{t'}^i \in O_{t'}^i, k = 1, ..., |O_{t'}^i|$ at time step t', where $T = \sum_{t'=1}^{t'=t}\sum_{k=1}^{k=|O_{t'}^i|} 1$ and $|O_{t'}^i|$ is the cardinality of the set $O_{t'}^i$, and $Pr^i(j|{}^k o_{t'}^i, \hat{\lambda}^i)$ denotes the posteriori probability of the observation ${}^k o_{t'}^i$ for the jth Gaussian of agent i.

The problem encountered using the ML estimation algorithm (2.10) is that the EM method optimizes the likelihood of the parameters given the position of the observation without taking the worth of that observation into account. Thus a minor change has to be made to (2.10). The same problem has been pointed out in the image processing literature [39].

We use a modified version of the solution introduced in [39] to repeat the EM algorithm m times in worthy locations with m chosen as

$$
m = \begin{cases} 1 + \gamma \text{ round}(\frac{f_q}{f_{\text{mode}}}) & f_q \geq f_{\text{mode}} \\ 1 & \text{otherwise} \end{cases}
\tag{2.11}
$$

where f_{mode} and γ are a threshold and an integer correction factor, respectively. Here for the locations with the worth lower than the threshold f_{mode}, we have $m = 1$, which means algorithm (2.10) just runs once. It is apparent that if the correction factor, γ, is increased, for a region with a higher worth algorithm (2.10) will be repeated more often.

2.4 Mutual Information

In this section, we add a mutual information term to each agent's utility function in order to find an action with a more informative observation. The agent's observations as an input to the estimation algorithm (2.10) play an important role in the performance of the system. Sending rich observation data can improve the estimation performance. Here our goal is to reduce the uncertainty of the unobserved area by finding more informative locations for future observations. In order to select informative observations, the entropy criterion has been used in information theory and applied mathematics contexts [68, 69]. In information theory, the conditional entropy quantifies the amount of information needed to describe the outcome of a random variable Y given the value of another variable X, which is written as $H(Y|X)$. In [33], the mutual information criterion has been proposed for observation selections, where it is a measure of the mutual independence of two variables. It is shown that maximizing the mutual information between an observed area and an unobserved area is more effective than the conditional entropy to reduce the uncertainty.

The mutual information objective function can be written as

$$I(X_{O^i_{1:t}} : X_{Q\backslash O^i_{1:t}}) = H(X_{Q\backslash O^i_{1:t}}) - H(X_{Q\backslash O^i_{1:t}}|X_{O^i_{1:t}}), \tag{2.12}$$

where I, $X_{O^i_{1:t}}$ and $X_{Q\backslash O^i_{1:t}}$ are the mutual information, the observed and unobserved variables corresponding to agent i, respectively. To compute the mutual information, two entropy functions have to be calculated in (2.12). Fortunately, for Gaussian processes there exists a close form to compute the conditional entropy which is given by

$$H(X_{Q\backslash O^i_{1:t}}|X_{O^i_{1:t}}) = \frac{1}{2}\log((2\pi e)^l|\Sigma_{X_{Q\backslash O^i_{1:t}}|X_{O^i_{1:t}}}|), \tag{2.13}$$

where e is the Euler number, and $\Sigma_{X_{Q\backslash O^i_{1:t}}|X_{O^i_{1:t}}}$ is a $l \times l$ covariance matrix of the conditional distribution of $X_{Q\backslash O^i_{1:t}}$ given $X_{O^i_{1:t}}$. The covariance $\Sigma_{X_{Q\backslash O^i_{1:t}}|X_{O^i_{1:t}}}$ calculated as

$$\Sigma_{X_{Q\backslash O^i_{1:t}}|X_{O^i_{1:t}}} = \Sigma_{X_{Q\backslash O^i_{1:t}}X_{Q\backslash O^i_{1:t}}} - \Sigma_{X_{Q\backslash O^i_{1:t}}X_{O^i_{1:t}}}\Sigma^{-1}_{X_{O^i_{1:t}}X_{O^i_{1:t}}}\Sigma_{X_{O^i_{1:t}}X_{Q\backslash O^i_{1:t}}},$$

where $\Sigma_{X_{Q\backslash O^i_{1:t}}X_{O^i_{1:t}}}$ is a covariance matrix each of whose entries is a function of the kernel function $K(a,b)$ for $a \in X_{Q\backslash O^i_{1:t}}, b \in X_{O^i_{1:t}}$, and $\Sigma_{X_{Q\backslash O^i_{1:t}}X_{O^i_{1:t}}} = \Sigma^T_{X_{Q\backslash O^i_{1:t}}X_{O^i_{1:t}}}$. The kernel function determines the correlation of the observations and can be defined using different functions. However, for spatial phenomena the correlation between observations depends on the distance of their locations. One of the most frequently used kernel functions is $K(a,b) = exp(-\frac{\|a-b\|_2^2}{h^2})$, where $\|a-b\|_2$ is the distance between the locations a and b and h is a positive constant. As can be seen, the variance does not depend on the observed values, so the mutual information can be calculated ahead of an agent's new action. Before adding the mutual information term to the utility function defined in (2.2), it should be discussed whether the modified game will remain a state-based potential game or not.

Lemma 3 *The coverage game Υ with $U_i(a(t),z(t)) = H(X_{Q\backslash O^i_{1:t}}) - H(X_{Q\backslash O^i_{1:t}}|X_{O^i_{1:t}})$ is an exact state-based potential game with the potential function*

$$\Phi_2(a(t),z(t)) = \sum_{j=1}^{N} U_j(a(t),z(t)). \tag{2.14}$$

Proof:

Note that the utility function of agent j, U_j, is only a function of agent j's observations $O^j_{1:t}$; hence $U_j\big(a_i'(t),a_{-i}(t),z(t)\big) = U_j\big(a(t),z(t)\big), \forall j \neq i$. Now, it is easy to see that (2.4) holds.

As can be seen, the state $z(t)$ does not appear in the utility function here, so there is no need to check the second condition as discussed in Lemma 2. ■

Now, we define a new utility function as

$$U_i(a(t), z(t)) = F(a_i(t), a_{-i}(t)) - F(a_i^0(t), a_{-i}(t)) - E_i^{\text{sense}}(a_i(t)) - E_i^{\text{move}}(a_i(t), z_i(t)) + \eta(t)(H(X_{Q \setminus O_{1:t}^i}) - H(X_{Q \setminus O_{1:t}^i} | X_{O_{1:t}^i})), \tag{2.15}$$

where $\eta(t)$ is a time varying coefficient that adjusts the importance of the mutual information term in comparison with the sensing and the energy consumption parts. The selection of $\eta(t)$ follows the following principle. Starting the search, the agents do not have a good estimate of the area and gathering proper data serves as an important role for improved sensing. However, by having a more accurate estimate, the weight of the optimal sensing considering the energy consumption has to be increased over time because we need to put more effort on finding the best configuration of the MSN.

Remark 4 *It is easy to show that the game Υ is an exact state-based potential game with the utility function (2.15) because $\Phi = \Phi_1 + \eta(t)\Phi_2$, where Φ_1 and Φ_2 are given by, respectively,* (2.3) *and* (2.14)*, can be used as a potential function.*

Remark 5 *In our solution set-up, the agents use the estimated parameter $\hat{\lambda}^i$ to calculate the estimation of f_q at each iteration where the estimate of f_q is used to evaluate the utility function (2.2). As mentioned in Section 2.1, f_q is assumed stationary but not the estimation of f_q. Hence the game parameters are not stationary any more. However, a standard assumption in game theory is a stationary ingredients. In [41], the case of slow variation of the game parameters is investigated.*

Suppose $\hat{\lambda} = \bigcup_{i=1}^{N} \hat{\lambda}^i$ *and*

$$|\lambda(t+1) - \lambda(t)| < \delta_2$$

$$|\hat{\lambda}(t) - \lambda(t)| < \delta_3,$$

where δ_2 *and* δ_3 *are the bound on the changing rate of* λ *and the bound on the estimation error of* λ*, respectively. Now, we can use Corollary 4 in [41], where it shows that using the binary log-linear learning for any* $\delta_1 > 0$*, there exist* $\delta_2 > 0$ *and* $\delta_3 > 0$ *such that the probability of converging to the potential function maximizer (here Nash equilibrium) will be greater than* $1 - \delta_1$*.*

Under the assumption of a stationary environment in Section 2.1, we will have $|\lambda(t+1) - \lambda(t)| = 0$*. In addition, by having more observations, the estimation error decreases, which implies the estimates* $\hat{\lambda}(t)$ *converge to* $\lambda(t)$ *and there exists a decreasing bound* δ_3 *on the estimation error. Thus* δ_1 *can be selected arbitrarily close to zero and the agents will converge to the Nash equilibrium stochastically.*

Remark 6 *It is possible to use a term corresponding to the mutual information when* $M = 1$ *is fixed. As it is shown in (2.13), the agents are able to calculate the conditional entropy before sensing an area for a Gaussian distribution. However, to calculate the conditional entropy of the future action (alternative action* $a_i'(t)$*) for* $M > 1$*, the future weighting coefficient* $\hat{w}_j^i(t),\ j = 1, ..., M$*, is required but cannot be calculated. Thus the designer has two choices: 1) The utility function (2.2) can be used for* $M \geq 1$*. 2) By fixing* $M = 1$*, the mutual information can be added to the utility function where (2.15) is used as the utility function.*

2.5 Simulation and Discussion

In this section, an illustrative simulation will be presented to show the advantages of our approach. Our results guarantee the stochastic stability of the Nash equilibrium. Although the theoretical results do not provide any estimate of the convergence rate, the simulation results in this section show an acceptable convergence rate in practice. Consider a 30×30 square in which each grid is 1×1 where a group of five mobile sensors ($N = 5$) are deployed in this area and each sensor can choose its a_i^r from the set $\{0, 1, 2, 3\}$, which means $r_{\text{min}} = 0, r_{\text{max}} = 3$, let $R^{\text{comm}} = 6$. We let $M = 1$ and use the Gaussian distribution as an estimation model. To clarify the ability of the proposed approach, a different distribution (not Gaussian distribution) has been used to create f_q. Here, the sum of two Copula distributions from the Frank family with the scalar parameters ρ = -0.2, 15 is used to generate f_q for all $q \in Q$. Fig. 2.2 shows the distribution worth of the area. It bears mentioning that the agents do not know the worth of the area before sensing them.

The mobile sensors' initial locations are chosen completely randomly. The agents employ the defined approach using (2.6), (2.10) and (2.11) with the utility function (2.15). The simulation parameters are chosen as $K_i = 3 \times 10^{-5}, K_i' = 3 \times 10^{-4}, i = 1, 2, .., N, \tau = 5 \times 10^{-3}, f_{\text{mode}} = 10^{-4}$, $\gamma = 1$ and $\eta(t) = \frac{0.001t+1}{100+t}$. Fig. 2.3 shows the final locations of the mobile sensors where they are aligned with the regions corresponding to the highest worth. The 5th agent shown as a green circle is far from the worthy region. Thus the agent has decided to turn off its own sensor to save power. Changing the actions of the sensing neighbors would create some space in more worthy area for agent 5. In such a situation the sensor will change its action and start sensing the area again.

In the next part of our simulation, we investigate the effect of using the mutual information

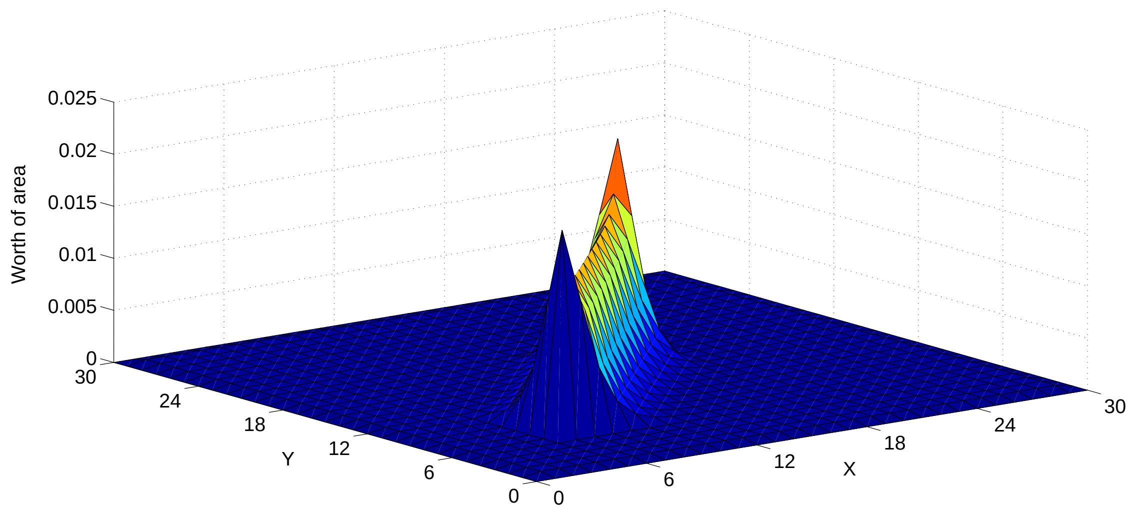

Figure 2.2: The worth of an area generated by Copula distributions

in the utility function and compare our method with an existing method in the literature. First, two simulations for $\eta(t) = \frac{0.001t+1}{100+t}$ and $\eta(t) = 0$ are compared with each other where the agents start from the same locations in both cases. As it can be seen in Fig. 2.4, when the mutual information is used, almost 70 percent of the area's worth is covered by the MSN after only 1700 time steps. However, without using the mutual information term, the MSN needs much more time to converge to its Nash equilibrium. Fig. 2.4 clarifies that using informative data to feed the estimation algorithm (2.10) will help the system to cover the area faster. Second, we compare our method with the Distributed Inhomogeneous Synchronous Coverage Learning (DISCL) method introduced in [88], where the method was just tested on an area that has an equal worth for all the grids. In such a set-up, the sensor agents' configuration will be a Nash equilibrium as long as their sensing regions do not have any intersection. Hence the agents would just spread out in the mission space, which is not a complex task for an MSN. To compare our method with the DISCL method on a fair basis, we

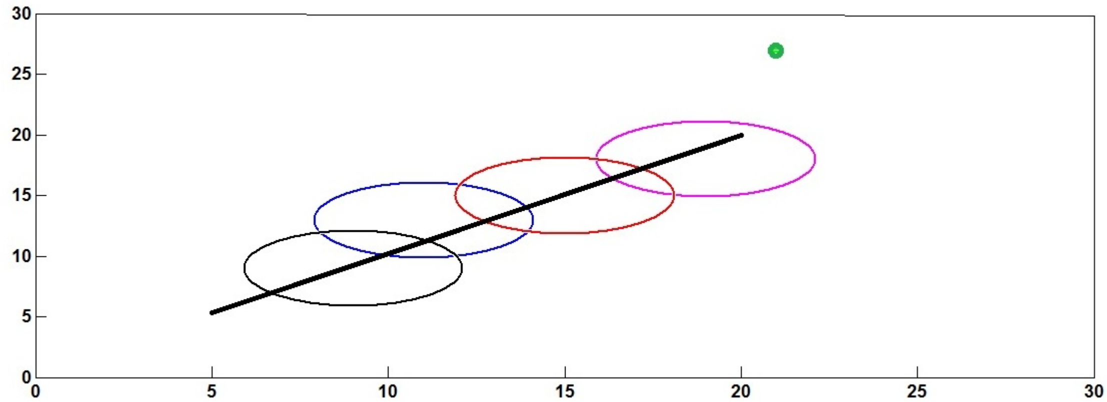

Figure 2.3: The final locations of the mobile sensors. The black line shows more worthy area and the color circles show the sensing area for each sensor. The green circle shows that agent 5 does not sense the area $a_5^r(T) = 0$.

implement the DISCL method in our simulation scenario. Different exploration rates are simulated where the most suitable performance is obtained using $(\frac{1}{t+2^{10}})^{\frac{1}{2}}$ as the exploration rate. The worth of the covered area using the DISCL method is illustrated in Fig. 2.4. The simulation results show that estimating the worth of the area using the knowledge of previous observations will improve the MSN performance. Sudden changes in the worth of the covered area can be seen when the DISCL method has been used. The reason is that the agents choose their future temporary actions uniformly from the set of available actions without considering the utility value of these actions. As a result, some of these actions might be less valuable.

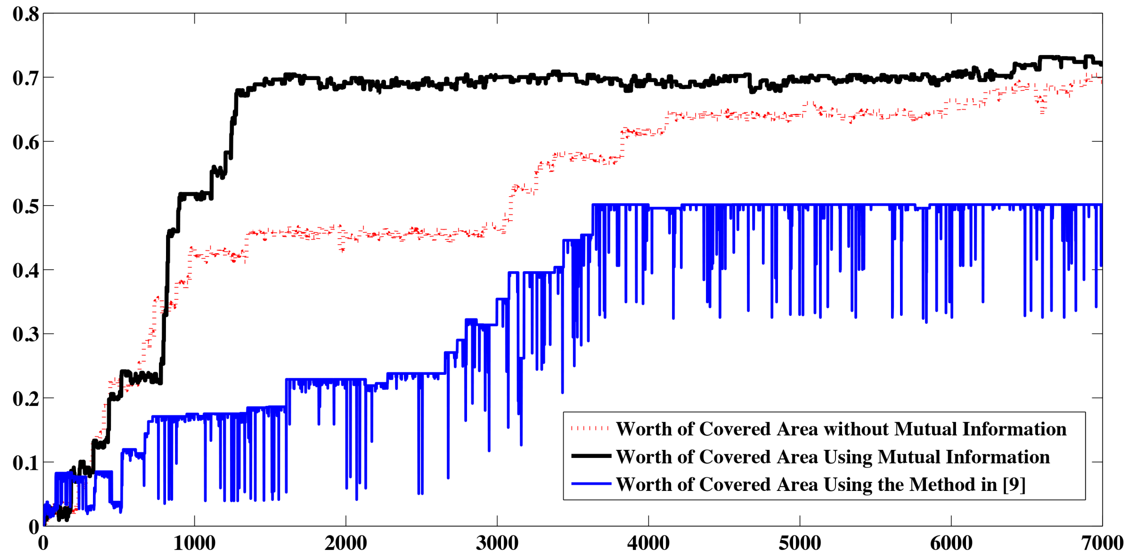

Figure 2.4: The worth of the covered area for two cases $\eta(t) = 0$, $\eta(t) = \frac{0.001t+1}{100+t}$ and simulation results using DISCL method with exploration rate $(\frac{1}{t+2^{10}})^{\frac{1}{2}}$.

Remark 7 *As mentioned in Remark 6, using the utility function (2.2) allows us to utilize $M \geq 1$. However, our simulations illustrate that for $M > 1$ one of the Gaussians' weights $\hat{w}_j^i(t)$ becomes clearly dominant after only a limited number of observations. Its consequence is that the estimated distribution will be one Gaussian instead of a mixture of Gaussians. Such a result happens because the agents maximize their utility functions while they observe the area at the same time. Hence the agent's observations will not cover the whole area and be focused more on one of the Gaussians in the GMM. This result shows that fixing $M = 1$ will not noticeably degrade the performance of our method.*

2.6 Experimental Validation

In this section, we provide experimental results to validate the algorithms introduced in (2.6), (2.10) and (2.11) with the utility function (2.15).

2.6.1 Robot Platform and Program Implementation

The physical robot experiments are based on three Pioneer 3-ATs from Adept Mobilerobots that are shown in Fig. 2.5. A top level diagram of the multi-agent control system architecture is shown in Fig. 2.6. Each robot is equipped with an on-board PC on which a server program is running. The on-board PC is connected to robot internally. Three client programs run on one laptop, which makes the communication between clients easier. Although all actions information is accessible, we emulate the distributed control scheme by using limited inner-robot information exchange. As it is shown by dashed arrows in Fig. 2.6, the robots are not allowed to use the actions of the robots that are not in their communication neighbor set. This means client i does not have access to the action of client j, if the distance between robots i and j is larger than R^{comm}. This enables us to test distributed control algorithms that involve only local information. The laptop and all three robots join the same WIFI network, where the communication between clients and servers is based on TCP.

A client program runs on the laptop, which keeps requesting a robot's action and calculating the new action using the algorithm introduced in (2.6), (2.10) and (2.11). The on-board PC is responsible for high-level control. It runs a server that keeps monitoring on a specified TCP port. According to the command received, it will choose to send the robot's position to a client or to receive a new action from the client. The Pioneer 3-AT also has a microcontroller which is designed to perform low-level control. It maintains support for all sensing and actuation features of the robots. A high-precision encoder/decoder will be utilized where there is no need for correction on the calculation of the distance traveled nor angle turned. Each server periodically (every 100 ms) receives

Figure 2.5: Multi-robot experimental platform at University of Califronia Riverside.

the current pose, (position and orientation), from the microcontroller. Then by using a feedback control, the server generates a new command and sends it to the microcontroller in order to move the robot into its desired position. Here a deadband of 2 degrees and 2.236 cm are incorporated to prevent oscillation.

2.6.2 Experimentation results

In our experiments, a 5×5 meters area is used, which is discretized into a 10×10 squared lattice. Fig. 2.7 shows the distribution of the worth, where a worthy area is assumed on the diagonal. We also deliberately introduce a worthy area on the corner. This will illustrate the advantage of using the mutual information term to escape the suboptimal equilibria. To show how the proposed algorithms perform in real applications, an obstacle is also introduced in the coverage area, where the obstacle is shown by red squares in Fig. 2.7. A group of three robots ($N = 3$) are

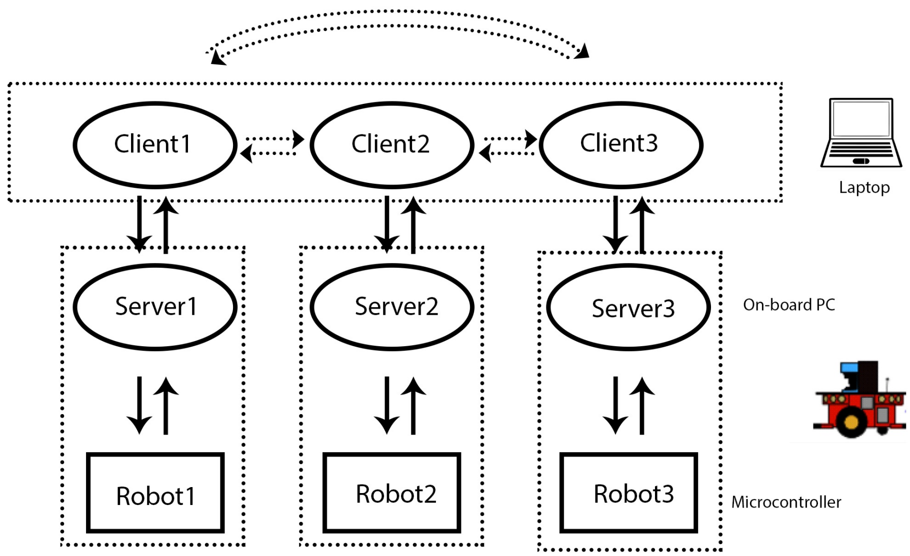

Figure 2.6: The implemented software architecture.

deployed in this area. The robots are initialized at $\begin{pmatrix} 1 \\ 3 \end{pmatrix}$, $\begin{pmatrix} 1 \\ 9 \end{pmatrix}$, and $\begin{pmatrix} 10 \\ 3 \end{pmatrix}$, respectively, where their initial positions are shown by green squares in Fig. 2.7. Each robot can choose its a_i^r from the set $\{0, 1, 2\}$, which means $r_{\text{min}} = 0, r_{\text{max}} = 2$, and $R^{\text{comm}} = 2r_{\text{max}} = 4$. We let $M = 1$ and use the Gaussian distribution as an estimation model. The agents employ the proposed approach using (2.6), (2.10) and (2.11) with the utility function (2.15). The experiment parameters are chosen as $K_i = 2 \times 10^{-4}, K_i' = 2.5 \times 10^{-2}, i = 1, 2, .., N, \tau = 5 \times 10^{-3}, f_{\text{mode}} = 10^{-3}$, and $\gamma = 1$.

In our first experiment the utility function (2.15) has been used with $\eta = 0$, where the robots maximize the worth of the covered area while considering the energy consumption. Fig. 2.8 shows the initial positions, the trajectories and the final positions of the robots. Green colored squares represent the worth of each square location. The percentage of opacity indicates the worth of the squares, where the lighter it is, the less worth it has. Also the obstacles are shown by red squares.

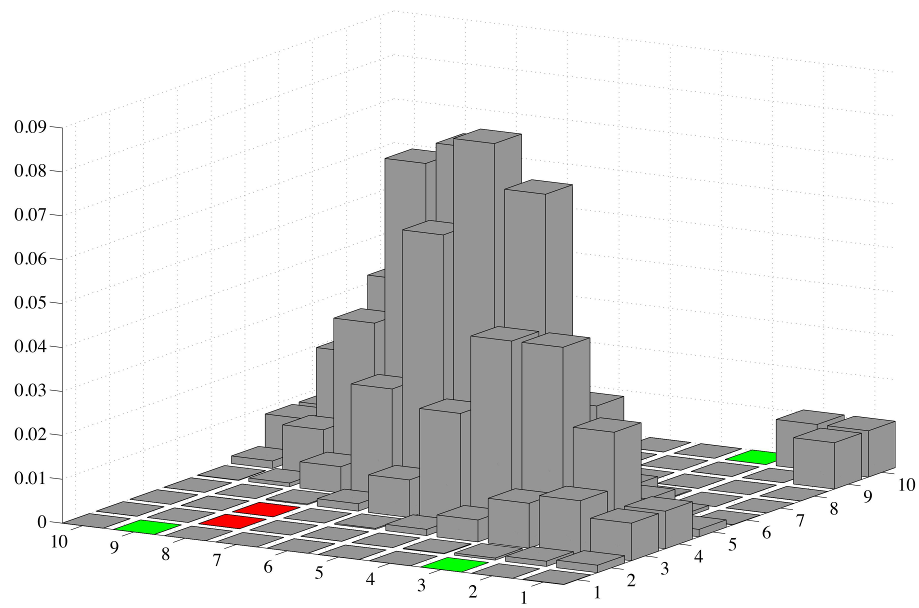

Figure 2.7: The worth of the area used for experimental validation. The initial positions of the robots and the obstacle are shown by green and red squares, respectively.

As we can see, robots 1 and 2 move toward better locations, which can cover more worthy areas. After 400 steps, their final sensing regions are indicated by yellow circles. Robot 3 is initialized at the corner, where it can sense some worthy areas. As it is shown in Fig. 2.8, robot 3 decides to stay there because the neighborhood areas have lower worth than its current position and the robot has no knowledge of more worthy areas far from its initial position.

In our next experiment, the utility function (2.15) has been used with $\eta = \frac{0.01t+5}{t+10}$. Here the mutual information term in the utility function helps the robots to choose more informative actions. Fig. 2.9 shows the initial positions, the trajectories and the final positions of the robots. In this case, the robots move to gather more information. Unlike Fig. 2.8, robot 3, does not stay at the corner in Fig. 2.9. Mutual information force makes the robots to move and gather more information,

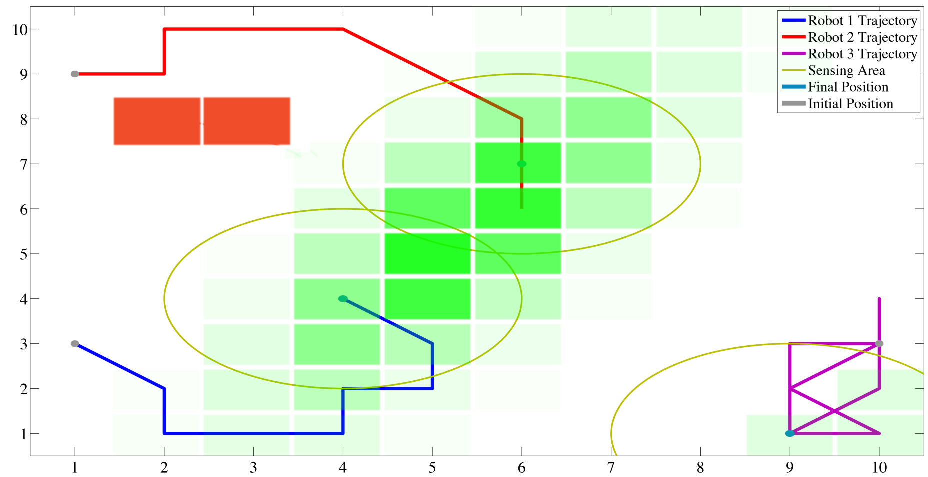

Figure 2.8: The trajectories and the final positions of the robots using the utility function (2.15) and for $\eta = 0$ (without mutual information).

where robot 3 find a better action to sense a more worthy area.

In Fig. 2.10, the worth of the covered area by three robots are compared for our two experiments. Fig. 2.10 shows that without the mutual information term the robots sense 84% of the worth of the area at about step 250. Using mutual information the robots move frequently, gather more information, and converge to the Nash equilibrium faster. Here the robots can sense 84% of the worth of the area before the 100th step. This shows the faster convergence rate using the mutual information in our utility function.

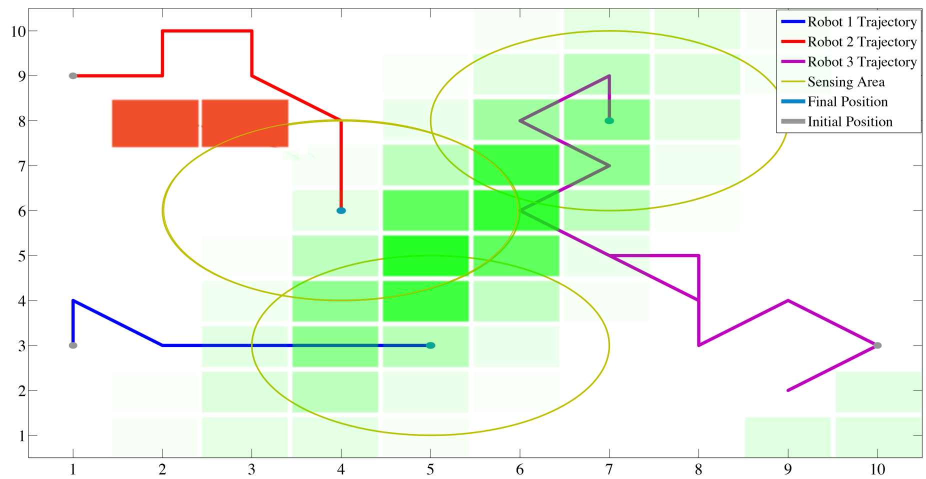

Figure 2.9: The trajectories and the final positions of the robots using the utility function (2.15) and for $\eta = \frac{0.01t+5}{t+10}$ (with mutual information).

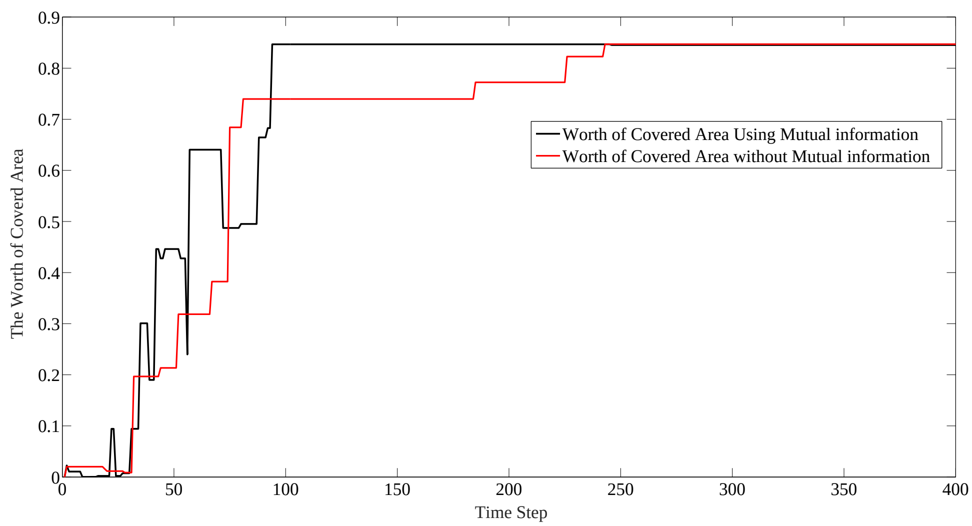

Figure 2.10: The worth of the covered area for two cases $\eta(t) = 0$ and $\eta = \frac{0.01t+5}{t+10}$.

Chapter 3

Distributed Convex Optimization with Time-Varying Cost Functions

3.1 Time-Varying Convex Optimization For Single-Integrator Dynamics

Consider a multi-agent system consisting of N physical agents with an interaction topology described by the undirected graph $\mathcal{G}$. It is common to adopt single-integrator or double-integrator models. Here, suppose that the agents satisfy the continuous-time single-integrator dynamics

$$\dot{x}_i(t) = u_i(t), \tag{3.1}$$

where $x_i(t) \in \mathbb{R}^m$ is the position of agent i, and $u_i(t) \in \mathbb{R}^m$ is the control input of agent i. Note that $x_i(t)$ and $u_i(t)$ are functions of time. Later for ease of notation we will write them as x_i and u_i. A time-varying local cost function $f_i : \mathbb{R}^m \times \mathbb{R}^+ \to \mathbb{R}$ is assigned to agent $i \in \mathcal{I}$, which is known to only agent i. The team cost function $f : \mathbb{R}^m \times \mathbb{R}^+ \to \mathbb{R}$ is denoted by $f(x,t) \triangleq \sum_{i=1}^{N} f_i(x,t)$ and is assumed to be convex. Note that here only $\sum_{i=1}^{N} f_i(x,t)$ is required to be convex but not necessarily each $f_i(x,t)$. Our objective is to design u_i for (3.1) using only local information and local interaction with neighbors such that all agents track the optimal state $x^*(t)$, where $x^*(t)$ is the minimizer of the time-varying convex optimization problem

$$\min_{x \in \mathbb{R}^m} f(x,t). \tag{3.2}$$

Assumption 8 *There exists a continuous $x^*(t)$ that minimizes the cost function $f(x,t)$.*

Because the inverse of the Hessian will be used in our algorithm, we need one of the following assumptions to guarantee its existence.

Assumption 9 *The function $f(x,t)$ is twice continuously differentiable with respect to x, with invertible Hessian $H(x,t) = \sum_{j=1}^{N} H_j(x,t),\ \forall x, t$.*

Assumption 10 *Each function $f_i(x,t)$ is twice continuously differentiable with respect to x, with invertible Hessian $H_i(x,t),\ \forall x, t$.*

Lemma 11 *Let $f(x) : \mathbb{R}^m \to \mathbb{R}$ be a continuously differentiable convex function. The function $f(x)$ is minimized at x^* if and only if $\nabla f(x^*) = 0$ [4]. Furthermore, for any strictly convex function $h(x) : \mathbb{R}^m \to \mathbb{R}$, the optimal solution x^*, assuming that it exists , is unique [7].*

Lemma 12 *[6] The symmetric matrix $\begin{pmatrix} Q & S \\ S^T & R \end{pmatrix}$ is positive definite if and only if one of the following conditions hold: (i) $Q > 0, R - S^T Q^{-1} S > 0$; or (ii) $R > 0, Q - S R^{-1} S^T > 0$.*

3.1.1 Centralized Time-Varying Convex Optimization

As a first step in this subsection, we focus on the time-varying convex optimization problem of

$$\min_{x} f_0(x,t), \tag{3.3}$$

where $f_0 : \mathbb{R}^m \times \mathbb{R}^+ \to \mathbb{R}$ is convex in x, for single-integrator dynamics

$$\dot{x} = u, \tag{3.4}$$

where $x, u \in \mathbb{R}^m$ are the system's state and control input, respectively. Next, an algorithm adapted from [77] will be proposed to solve the problem defined by (3.3) for the system (3.4). The control input is proposed for (3.4) as

$$u = -H_0^{-1}(x,t)(\tau \nabla f_0(x,t) + \frac{\partial \nabla f_0(x,t)}{\partial t}), \tag{3.5}$$

where $\tau > 0$ is a positive coefficient; $\nabla f_0(x,t)$ and $H_0(x,t)$ are respectively, the first and the second derivative of the cost function $f_0(x,t)$ with respect to x, called the gradient and Hessian.

Theorem 13 *Suppose that f_0 satisfies Assumptions 8 and 9. Using* (3.5) *for* (3.4)*, $x(t)$ converges to the optimal trajectory $x_0^*(t)$, the minimizer of* (3.3)*, i.e.* $\lim_{t\to\infty}[x(t) - x_0^*(t)] = 0$.

Proof: Define the positive-definite Lyapunov function candidate $W = \frac{1}{2}\nabla f_0(x,t)^T \nabla f_0(x,t)$. The derivative of W along the system (3.4) with control input (3.5) is $\dot{W} = \nabla f_0(x,t)^T H_0(x,t)\dot{x} + \nabla f_0(x,t)^T \frac{\partial \nabla f_0(x,t)}{\partial t} = -\tau \nabla f_0(x,t)^T \nabla f_0(x,t)$. Therefore, $\dot{W} < 0$ for $\nabla f_0 \neq 0$. This guarantees that ∇f_0 will asymptotically converge to zero when $t \to \infty$. Then by using Lemma 11 and under Assumption 8, it is easy to see that $x(t)$ converges to $x_0^*(t)$, and f_0 will be minimized. ■

Remark 14 *There exist other choices for control input u instead of the one proposed in (3.5). For example, $u = -\tau\nabla f_0(x,t) - H_0^{-1}(x,t)\frac{\partial\nabla f_0(x,t)}{\partial t}$ might be used. In this alternative control input, it can be seen that for a time-invariant cost function, $\frac{\partial\nabla f_0(x,t)}{\partial t} = 0$. Hence we will have the well-known gradient descent algorithm. For a time-invariant cost function, the proposed algorithm* (3.5) *will become a Newton algorithm, which is generally much faster than the gradient descent algorithm.*

The results from Theorem 13 can be extended to minimize the convex function $f(x,t) = \sum_{i=1}^{N} f_i(x,t)$. If Assumption 9 holds, with

$$u = \Big(\sum_{j=1}^{N} H_j(x,t)\Big)^{-1}\Big(\tau\sum_{j=1}^{N}\nabla f_j(x,t) + \sum_{j=1}^{N}\frac{\partial\nabla f_j(x,t)}{\partial t}\Big) \tag{3.6}$$

for the single-integrator dynamic system (3.4), the function $f(x,t)$ is minimized. Unfortunately, (3.6) is a centralized solution for agents with single-integrator dynamics relying on the knowledge of all $f_i, i = 1, ..., N$. In Subsections 3.1.2 and 3.1.3, (3.6) will be exploited to propose two algorithms for solving the time-varying convex optimization problem for single-integrator dynamics in a distributed manner.

3.1.2 Distributed Time-Varying Convex Optimization Using Neighbors' Positions

In this subsection, we focus on solving the distributed time-varying convex optimization problem (3.2) for agents with single-integrator dynamics (3.1). Each agent has access to only its own position and the relative positions between itself and its neighbors. In some applications, the relative positions can be obtained by using only agents' local sensing capabilities, which might in turn eliminate the communication necessity between agents. The problem defined in (3.2) is equivalent to

$$\min_{x_i} \sum_{i=1}^{N} f_i(x_i, t) \text{ subject to } x_i = x_j, \ \ \forall i, j \in \mathcal{I}. \tag{3.7}$$

Intuitively, the problem is deformed as a consensus problem and a minimization problem on the team cost function $\sum_{i=1}^{N} f_i(x_i, t)$. Here the goal is that the states $x_i(t), \forall i \in \mathcal{I}$, converge to the optimal trajectory $x^*(t)$, i.e.,

$$\lim_{t\to\infty} [x_i(t) - x^*(t)] = 0. \tag{3.8}$$

The control input is proposed for (3.1) as

$$\begin{aligned}
u_i &= -\sum_{j\in N_i} \beta_{ij}\text{sgn}\big(x_i - x_j\big) + \phi_i, \\
\dot{\beta}_{ij} &= \|x_i - x_j\|_1, \quad j \in N_i, \\
\phi_i &\triangleq -H_i^{-1}(x_i, t)\big(\nabla f_i(x_i, t) + \frac{\partial \nabla f_i(x_i, t)}{\partial t}\big),
\end{aligned} \tag{3.9}$$

where ϕ_i is an internal signal, β_{ij} is a varying gain with $\beta_{ij}(0) = \beta_{ji}(0) \geq 0$, and sgn$(\cdot)$ is the signum function defined componentwise. Note that ϕ_i depends on only agent i's position. Here (3.9) is a discontinuous controller. It is worth mentioning that unlike continuous or smooth systems, the equilibrium concept of setting the right hand equal to zero to find the equilibrium point might not be valid for discontinuous systems. Let $X = [x_1^T, x_2^T, ..., x_N^T]^T$, and $\Phi = [\phi_1^T, \phi_2^T, ..., \phi_N^T]^T$ denote, respectively, the aggregated states and the aggregated internal signals of the N agents. We also define $\Pi \triangleq I_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T$. Define agent i's consensus error as $e_{X_i} = x_i - \frac{1}{N}\sum_{\ell=1}^{N} x_\ell$. Define the consensus error vector $e_X = (\Pi \otimes I_m)X$. Note that Π has one simple zero eigenvalue with $\mathbf{1}_N$ as its right eigenvector and has 1 as its other eigenvalue with the multiplicity $N-1$. Then it is easy to see that $e_X = 0$ if and only if $x_i = x_j \ \forall i, j \in \mathcal{I}$.

Remark 15 *With the signum function in the proposed algorithms in this study, the right-hand sides of the closed-loop systems are discontinuous. Thus, the solution should be investigated in terms of differential inclusions by using nonsmooth analysis [19, 25]. However, since the signum function is measurable and locally essentially bounded, the Filippov solutions of the closed-loop dynamics always exist. Also the Lyapunov function candidates adopted in the proofs hereafter are continuously differentiable. Therefore, the set-valued Lie derivative of them is a singleton at the discontinuous points and the proofs still hold. To avoid symbol redundancy, we do not use the differential inclusions in the proofs. Furthermore, Filippov solutions are absolutely continuous curves [19], which means that the agents' states are continuous functions.*

The remainder of this subsection is devoted to the verification of the algorithm (3.9). In Proposition 1, we will show that the agents reach consensus using (3.9). Then this result will be used in Theorem 18 to prove that the agents minimize the team cost function as $t \to \infty$.

Definition 16 *Defining $a'_{ij} = a_{ij}\beta_{ij}$, a new Laplacian matrix $L' = [l'_{ij}] \in \mathbb{R}^{N\times N}$ is introduced, where $l'_{ii} = \sum_{j=1,j\neq i}^{N} a'^2_{ij}$ and $l'_{ij} = -a'^2_{ij}$. Since $a'_{ij} = a'_{ji}$, the matrix L' is symmetric. Similar to the definition of D, $D' \triangleq [d'_{ij}] \in \mathbb{R}^{n\times m}$ is the incidence matrix associated with L', where $d'_{ij} = -a'_{ij}$ if the edge e_j leaves node i, $d_{ij} = a'_{ij}$ if it enters node i, and $d_{ij} = 0$ otherwise. Thus, L' can be given by $L' = D'D'^T$.*

Assumption 17 *With ϕ_i defined in* (3.9)*, there exists a positive constant $\bar{\phi}$ such that $\|\phi_i - \phi_j\| \leq \bar{\phi}$, $\forall i, j \in \mathcal{I}$, and $\forall t$.*

Proposition 1 *Suppose that the graph $\mathcal{G}$ is connected and Assumption 17 holds. System* (3.1) *with the algorithm (3.9) reaches consensus, i.e, $x_i = x_j, \forall i, j \in \mathcal{I}$, as $t \to \infty$.*

Proof: Using Definition 16, the closed-loop system (3.1) with the control input (3.9) can be recast into a compact form as

$$\dot{X} = -(D' \otimes I_m)\operatorname{sgn}\big((D^T \otimes I_m)X\big) + \Phi, \tag{3.10}$$

where D and D' are defined in Section 1.2 and Definition 16, respectively. We can rewrite (3.10) as

$$\dot{e}_X = -\,(D' \otimes I_m)\operatorname{sgn}\big((D^T \otimes I_m)e_X\big) + (\Pi \otimes I_m)\Phi, \tag{3.11}$$

where we have used the fact that $\Pi D' = D'$. Define the Lyapunov function candidate

$$W = \frac{1}{2}e_X^T e_X + \frac{1}{2}\sum_{i=1}^{N}\sum_{j\in N_i}(\beta_{ij} - \bar{\beta})^2.$$

The time derivative of W along (3.11) can be obtained as

$$\begin{aligned}
\dot{W} &= -e_X^T(D' \otimes I_m)\operatorname{sgn}\big((D^T \otimes I_m)e_X\big) + e_X^T(\Pi \otimes I_m)\Phi + \frac{1}{2}\sum_{i=1}^{N}\sum_{j\in N_i}(\beta_{ij} - \bar{\beta})\dot{\beta}_{ij} \\
&= -\sum_{i=1}^{N}\sum_{j\in N_i}\frac{\beta_{ij}}{2}\left\|e_{X_i} - e_{X_j}\right\|_1 \\
&+ \frac{1}{2N}\sum_{i=1}^{N}\sum_{j=1}^{N}(e_{X_i} - e_{X_j})(\phi_i - \phi_j) + \frac{1}{2}\sum_{i=1}^{N}\sum_{j\in N_i}(\beta_{ij} - \bar{\beta})(\|x_i - x_j\|_1) \\
&= -\sum_{i=1}^{N}\sum_{j\in N_i}\frac{\beta_{ij}}{2}\left\|e_{X_i} - e_{X_j}\right\|_1 + \frac{1}{2N}\sum_{i=1}^{N}\sum_{j=1}^{N}(e_{X_i} - e_{X_j})(\phi_i - \phi_j) \\
&- \frac{\bar{\beta}}{2}\sum_{i=1}^{N}\sum_{j\in N_i}\left\|e_{X_i} - e_{X_j}\right\|_1 + \sum_{i=1}^{N}\sum_{j\in N_i}\frac{\beta_{ij}}{2}\left\|e_{X_i} - e_{X_j}\right\|_1 \\
&\le \frac{1}{2N}\sum_{i=1}^{N}\sum_{j=1}^{N}\left\|e_{X_i} - e_{X_j}\right\|_1\|\phi_i - \phi_j\| - \frac{\bar{\beta}}{2}\sum_{i=1}^{N}\sum_{j\in N_i}\left\|e_{X_i} - e_{X_j}\right\|_1 \\
&\le \frac{\bar{\phi}}{2N}\sum_{i=1}^{N}\sum_{j=1}^{N}\left\|e_{X_i} - e_{X_j}\right\|_1 - \frac{\bar{\beta}}{2}\sum_{i=1}^{N}\sum_{j\in N_i}\left\|e_{X_i} - e_{X_j}\right\|_1,
\end{aligned} \tag{3.12}$$

where the last inequality holds under Assumption 17. Because $\mathcal{G}$ is connected, we have

$$\begin{aligned}
\dot{W} &\le \frac{\bar{\phi}}{2}\max_i\{\sum_{j=1,j\neq i}^{N}\left\|e_{X_i}-e_{X_j}\right\|_1\}-\frac{\bar{\beta}}{2}\sum_{i=1}^{N}\sum_{j\in N_i}\left\|e_{X_i}-e_{X_j}\right\|_1\\
&\le \frac{(N-1)\bar{\phi}}{4}\sum_{i=1}^{N}\sum_{j\in N_i}\left\|e_{X_i}-e_{X_j}\right\|_1-\frac{\bar{\beta}}{2}\sum_{i=1}^{N}\sum_{j\in N_i}\left\|e_{X_i}-e_{X_j}\right\|_1.
\end{aligned}$$

Selecting $\bar{\beta}$ such that $\bar{\beta} > \frac{(N-1)\bar{\phi}}{2}$, we have

$$\begin{aligned}
\dot{W} &\le (\frac{(N-1)\bar{\phi}}{4}-\frac{\bar{\beta}}{2})\sum_{i=1}^{N}\sum_{j\in N_i}\left\|e_{X_i}-e_{X_j}\right\|_1 \qquad (3.13)\\
&= (\frac{(N-1)\bar{\phi}}{2}-\bar{\beta})e_X^T(D\otimes I_m)\mathrm{sgn}\big((D^T\otimes I_m)e_X\big) \le (\frac{(N-1)\bar{\phi}}{2}-\bar{\beta})\left\|(D^T\otimes I_m)e_X\right\|_1\\
&\le (\frac{(N-1)\bar{\phi}}{2}-\bar{\beta})\sqrt{e_X^T(DD^T\otimes I_m)e_X} \le (\frac{(N-1)\bar{\phi}}{2}-\bar{\beta})\sqrt{\lambda_2[L]}\left\|e_X\right\|_2 < 0,
\end{aligned}$$

where in the last inequality the fact that $L = DD^T$ and Lemma 1 have been used. Therefore, having $W \ge 0$ and $\dot{W} \le 0$, we can conclude that $e_X \in \mathcal{L}_\infty$. By integrating both sides of (3.13), we can see that $e_X \in \mathcal{L}_2$. Now, applying Barbalat's Lemma [72], we obtain that e_X will asymptomatically converge to zero when $t \to \infty$ and hence the agents' positions reach consensus, i.e, $x_i = x_j, \forall i, j \in \mathcal{I}$, as $t \to \infty$. ■

Theorem 18 *Suppose that the graph $\mathcal{G}$ is connected, and Assumptions 8, 10 and 17 hold. If $H_i(x_i,t) = H_j(x_j,t), \forall t,\ \forall i,j \in \mathcal{I}$, by employing the algorithm (3.9) for the system (3.1), the optimization goal* (3.8) *is achieved.*

Proof: Define the Lyapunov function candidate

$$W = \frac{1}{2}(\sum_{j=1}^{N}\nabla f_j(x_j))^T(\sum_{j=1}^{N}\nabla f_j(x_j)), \qquad (3.14)$$

where W is positive definite with respect to $\sum_{j=1}^{N}\nabla f_j(x_j)$. The time derivative of W can be obtained as $\dot{W} = (\sum_{j=1}^{N}\nabla f_j(x_j))^T(\sum_{j=1}^{N}H_j\dot{x}_j + \sum_{j=1}^{N}\frac{\partial}{\partial t}\nabla f_j(x_j))$. Under the assumption of identical Hessians, we will have

$$\dot{W} = (\sum_{j=1}^{N} \nabla f_j(x_j))^T (H_i)(\sum_{j=1}^{N} \dot{x}_j + H_i^{-1} \sum_{j=1}^{N} \frac{\partial}{\partial t} \nabla f_j(x_j)). \tag{3.15}$$

On the other hand, by using controller (3.9) for the system (3.1) and summing up both sides for $j = 1, 2, ..., N$, we know that $\sum_{j=1}^{N} \dot{x}_j = \sum_{j=1}^{N} \phi_j$. Then we can rewrite (3.15) as $\dot{W} = -(\sum_{j=1}^{N} \nabla f_j(x_j))^T (\sum_{j=1}^{N} \nabla f_j(x_j)$. Therefore, $\dot{W} < 0$ for $\sum_{j=1}^{N} \nabla f_j(x_j) \neq 0$. This guarantees that $\sum_{j=1}^{N} \nabla f_j(x_j)$ will asymptomatically converge to zero. Now, using Proposition 1, Lemma 11 and under Assumption 8, it is easy to see that as $t \to \infty$ the team cost function $\sum_{i=1}^{N} f_i(x_i, t)$ will be minimized, where $x_i = x_j, \ \forall i, j \in \mathcal{I}$. ■

Remark 19 *In algorithm* (3.9) *each agent i is required to know $\frac{\partial \nabla f_i(x_i,t)}{\partial t}$, which might be restrictive. However, there are applications where each agent knows the closed form of its own local cost function (e.g., motion control with an optimization objective) or at least the agent knows how the cost function is varying with respect to time (e.g., home automation). For example, in motion control with an optimization objective, it is possible that each agent knows the closed form of its local cost function or in home automation smart electrical devices need to agree on the total amount of energy consumption that maximizes an overall utility-function. This utility-function is given by the sum of the utility functions of the devices. However, a varying price rate for electricity during a day makes the optimization problem time varying. Although, the price rate of the electricity is varying during the day, it is known to the agents beforehand. Hence, calculating $\frac{\partial \nabla f_i(x_i,t)}{\partial t}$ might not be an issue in this application. Furthermore, there are algorithms to estimate the derivative of a function by knowing only the value of the function at each time t. How to apply the idea to distributed continuous-time time-varying optimization is a possible direction for our future studies.*

Remark 20 *Assumption 17 intuitively places a bound on the Hessians and the changing rates of the gradients of the cost functions with respect to* t*. In Appendix .1, we will show that Assumption 17 holds if the cost functions with identical Hessians satisfy certain conditions such that the boundedness of* $\|x_i(t) - x_j(t)\|$ *for all* t *guarantees the boundedness of* $\|\nabla f_j(x_j, t) - \nabla f_i(x_i, t)\|$ *and* $\left\|\frac{\partial \nabla f_j(x_j,t)}{\partial t} - \frac{\partial \nabla f_i(x_i,t)}{\partial t}\right\|$, $\forall i, j \in \mathcal{I}$, *for all* t*. For example, consider the cost functions that are commonly used for energy minimization, e.g.,* $f_i(x_i, t) = (ax_i + g_i(t))^2$, *where* a *is a positive constant and* $g_i(t)$ *is a time-varying function particularly for agent* i*. For these cost functions, the boundedness of* $\|x_i(t) - x_j(t)\|$ *for all* t *guarantees the boundedness of* $\|\nabla f_j(x_j, t) - \nabla f_i(x_i, t)\|$ *and* $\left\|\frac{\partial \nabla f_j(x_j,t)}{\partial t} - \frac{\partial \nabla f_i(x_i,t)}{\partial t}\right\|$, $\forall i, j \in \mathcal{I}$, *for all* t, *if* $\|g_i(t) - g_j(t)\|$ *and* $\|\dot{g}_i(t) - \dot{g}_j(t)\|$ *are bounded. Hence to satisfy Assumption 17 for* $f_i(x_i, t) = (ax_i + g_i(t))^2$, *it is sufficient to have a bound on* $\|g_i(t) - g_j(t)\|$ *and* $\|\dot{g}_i(t) - \dot{g}_j(t)\|$.

In Subsection 3.1.3 an estimator-based algorithm is introduced, where the assumption on identical Hessians is relaxed.

3.1.3 Estimator-Based Distributed Time-Varying Convex Optimization

In this subsection, an estimator-based algorithm is designed such that each agent calculates (3.6) in a distributed manner. To achieve this goal, distributed average tracking is used as a tool. Each agent generates an estimate of (3.6). Then a controller is designed such that each agent tracks its own generated signal while guaranteeing that the agents reach consensus.

The proposed algorithm for the system (3.1) has two separate parts, the estimator and controller, where the estimator part is given by

$$\dot{\xi}_i = \alpha \sum_{j \in N_i(t)} \operatorname{sgn}(w_j - w_i), \qquad w_i = \xi_i + \nabla f_i(x_i, t) \tag{3.16}$$

$$\dot{\psi}_i = \beta \sum_{j \in N_i(t)} \operatorname{sgn}(\theta_j - \theta_i), \quad \theta_i = \psi_i + H_i(x_i, t) \tag{3.17}$$

$$\dot{\phi}_i = \gamma \sum_{j \in N_i(t)} \operatorname{sgn}(\varsigma_j - \varsigma_i), \quad \varsigma_i = \phi_i + \frac{\partial}{\partial t}(\nabla f_i(x_i, t)) \tag{3.18}$$

$$S_i = -\theta_i^{-1}(\tau w_i + \varsigma_i), \tag{3.19}$$

where $N_i(t)$ is the set of agent i's neighbors at time t. The controller part is given by

$$u_i = -\sum_{j \in N_i(t)} \operatorname{sig}(x_i - x_j)^\eta + S_i, \tag{3.20}$$

where α, β, γ, and τ are positive coefficients to be selected and $0 < \eta < 1$. Also ξ_i, ψ_i, and ϕ_i are the internal states of the distributed average tracking estimators, where their initial values are such that[1]

$$\sum_{j=1}^{N} \xi_j(0) = \sum_{j=1}^{N} \psi_j(0) = \sum_{j=1}^{N} \phi_j(0) = 0. \tag{3.21}$$

The estimator part (3.16)-(3.19), generates the internal signal for each agent and the controller part (3.20), guarantees the consensus. Here the separation principle can be applied if the estimator part will converge in a finite time.

Assumption 21 *The estimators' coefficients α, β, and γ satisfy the following inequalities: $\alpha > \sup_t \left\| \frac{\partial}{\partial t}(\nabla f_i(x_i, t)) \right\|_\infty$, $\beta > \sup_t \left\| \frac{\partial}{\partial t} H_i(x_i, t) \right\|_\infty$, and $\gamma > \sup_t \left\| \frac{\partial^2}{\partial t^2}(\nabla f_i(x_i, t)) \right\|_\infty$, $\forall i \in \mathcal{I}$.*

Assumption 21 can be satisfied if the partial derivatives of the Hessians, the first- and second-order partial derivatives of the gradient are bounded.

[1] As a special case the initial values can be chosen as $\xi_j(0) = \psi_j(0) = \phi_j(0) = 0 \; \forall j \in \mathcal{I}$.

Theorem 22 *Suppose that the graph $\mathcal{G}(t)$ is connected for all t. If Assumptions 8, 9, and 21 and the initial condition* (3.21) *hold, for the system (3.1) with the algorithm* (3.16)-(3.20)*, the optimization goal* (3.8) *is achieved.*

Proof: **Estimator**: It follows from Theorem 2 in [13] that if $\alpha > \sup_t \left\|\frac{\partial}{\partial t}(\nabla f_i(x_i,t))\right\|_\infty, \beta > \sup_t \left\|\frac{\partial}{\partial t}H_i(x_i,t)\right\|_\infty$, and $\gamma > \sup_t \left\|\frac{\partial^2}{\partial t^2}(\nabla f_i(x_i,t))\right\|_\infty$, then there exists a $T > 0$ such that for all $t \geq T$,

$$\begin{aligned}\left\|w_i - \frac{1}{N}\sum_{j=1}^{N}\nabla f_j(x_j,t)\right\|_2 =&0\\ \left\|\theta_i - \frac{1}{N}\sum_{j=1}^{N}H_j(x_j,t)\right\|_2 =&0\\ \left\|\varsigma_i - \frac{1}{N}\sum_{j=1}^{N}\frac{\partial}{\partial t}\nabla f_j(x_j,t)\right\|_2 =&0\end{aligned}$$

Now from (3.19), for all $t \geq T$, the estimated signal S_i is equal to

$$S_i = -(\sum_{j=1}^{N}H_j(x_j,t))^{-1}\big(\tau\sum_{j=1}^{N}\nabla f_j(x_j,t) + \sum_{j=1}^{N}\frac{\partial\nabla f_j(x_j,t)}{\partial t}\big).$$

Till now we have shown that all agents generate the internal signal S_i, where $S_i = S_j, \forall i,j \in \mathcal{I}$, in a finite time.

Controller: Note that $\forall\, t \geq T,\ S_i = S_j,\ \forall i,j \in \mathcal{I}$, denoted as $\bar{S}$. For $t \geq T$ using (3.20) for (3.1), we have

$$\dot{x}_i = -\sum_{j\in N_i(t)}\text{sig}(x_i - x_j)^\eta + \bar{S}. \tag{3.22}$$

For $t \geq T$, rewriting (3.22) using new variables $\tilde{x}_i = x_i - \int_T^t \bar{S}dt$, we have

$$\dot{\tilde{x}}_i = -\sum_{j\in N_i(t)}\text{sig}(\tilde{x}_i - \tilde{x}_j)^\eta. \tag{3.23}$$

It is proved in [80] that using (3.23) for the system (3.1), there exists a time T' such that $\tilde{x}_i = \tilde{x}_j, \ \forall i, j \in \mathcal{I}$. As a result we have $x_i = x_j, \ \forall i, j \in \mathcal{I}$, and $\dot{x}_i = \bar{S}, \ \forall \ t \geq T + T'$. Now, it is easy to see that according to (3.6) the optimization goal (3.8) is achieved. ■

Remark 23 *Satisfying the conditions mentioned in Assumption 21 might be restrictive but they hold for an important class of cost functions. For example, if the agents' cost functions are in the form of* $f_i(x_i, t) = (a_i x_i + g_i(t))^{2n}$*, where the Hessians are not equal, the above conditions are equivalent to the conditions that* $\|g_i(t)\|, \|\dot{g}_i(t)\|$*, and* $\|\ddot{g}_i(t)\|$ *are bounded. This is applicable to a vast class of time-varying functions,* $g_i(t)$*, such as* $sin(t), e^{-t}cos(t), \frac{1}{1+t}$ *and* $tanh(t)$*.*

Remark 24 *The algorithm introduced in (3.16)-(3.20) just requires that Assumptions 9 and 21 hold. Note that in Assumption 9, it is not required that each agent's cost function* $f_i(x, t)$ *has invertible Hessian but instead their sum, which is weaker than Assumption 10. In contrast, for the algorithm* (3.9)*, not only Assumption 10 and the conditions mentioned in Remark 20 have to be satisfied for each individual function* $f_i(x_i, t)$*, it requires the agents' Hessians to be equal. However, in algorithm* (3.9) *the agents just need their own positions and the relative positions between themselves and their neighbors. In some applications, these pieces of information can be obtained by sensing; hence the communication necessity might be eliminated. In contrast, in algorithm (3.16)-(3.20) each agent must communicate three variables* w_i, ς_i *and* ψ_i *with its neighbors, which necessitates the communication requirement.*

3.2 Time-Varying Convex Optimization For Double-Integrator Dynamics

In this section, we study the convex optimization problem of time-varying cost functions for double-integrator dynamics. In some applications, it might be more realistic to model the equations of the motion of the agents with double-integrator dynamics, i.e., mass-force model, to take into account the effect of inertia. Unlike single-integrator dynamics, in case of double-integrator dynamics, the agents' positions and velocities at each time must be determined properly such that the team cost function is minimized. However, we have only direct control on each agent's acceleration and hence there exist new challenges. As a first step, in Subsection 3.2.1, a centralized algorithm will be introduced.

3.2.1 Centralized Time-Varying Convex Optimization

In this subsection, we focus on the time-varying convex optimization problem of (3.3) for double-integrator dynamics

$$\dot{x}(t) = v(t) \qquad \dot{v}(t) = u(t), \tag{3.24}$$

where $x, v, u \in \mathbb{R}^m$ are the position, the velocity, and the control input, respectively. Our goal is to design the control input u to minimize the cost function f_0. In Theorem 25, an algorithm will be proposed to solve the problem defined by (3.3) and (3.24). The control input is proposed for (3.24) as

$$\begin{aligned} u = &- H_0^{-1}(x,t)\big(\frac{\partial}{\partial t}\frac{d\nabla f_0(x,t)}{dt} + \frac{d\nabla f_0(x,t)}{dt}\big) - H_0(x,t)\nabla f_0(x,t) \\ &+ \Big[H_0^{-1}(x,t)(\frac{d}{dt}H_0(x,t))H_0^{-1}(x,t)\Big]\big(\frac{\partial \nabla f_0(x,t)}{\partial t} + \nabla f_0(x,t)\big). \end{aligned} \tag{3.25}$$

Theorem 25 *Suppose that f_0 satisfies Assumptions 8 and 9. Using* (3.25) *for* (3.24)*, $x(t)$ converges to the optimal trajectory $x_0^*(t)$, the minimizer of* (3.3)*, i.e.* $\lim_{t\to\infty}[x(t) - x_0^*(t)] = 0$.

Proof: Define the Lyapunov function candidate

$$W = \frac{1}{2}\nabla f_0^T(x,t)\nabla f_0(x,t) + \frac{1}{2}(S_0 - v)^T(S_0 - v),$$

where $S_0 = {H_0}^{-1}(x,t)\big(\frac{\partial}{\partial t}\nabla f_0(x,t) + \nabla f_0(x,t)\big)$. The derivative of W along the system (3.24) with the control input (3.25) is obtained as

$$\begin{aligned} \dot{W} &= \nabla f_0^T(x,t)(H_0(x,t)v + \frac{\partial}{\partial t}\nabla f_0(x,t)) \\ &+ (S_0 - v)^T(\dot{S}_0 - u) = -\nabla f_0^T(x,t)\nabla f_0(x,t) \end{aligned} \tag{3.26}$$

Therefore, $\dot{W} < 0$ for $\nabla f_0(x,t) \neq 0$. Now, having $W \geq 0$ and $\dot{W} \leq 0$, we can conclude that $\nabla f_0(x,t), S_0 - v \in \mathcal{L}_\infty$. By integrating both sides of (3.26), we can see that $\nabla f_0(x,t) \in \mathcal{L}_2$. Now, applying Barbalat's Lemma [72], we obtain that $\nabla f_0(x,t)$ will asymptomatically converge to zero when $t \to \infty$. Then by using Lemma 11 and under Assumption 8, it is easy to see that f_0 will be minimized. ■

The results from Lemma 25 can be extended to minimize the convex function $f(x,t) = \sum_{i=1}^N f_i(x,t)$. If Assumption 9 holds, with

$$\begin{aligned} u = &-\Big(\sum_{i=1}^N H_i(x,t)\Big)^{-1}\Big(\sum_{i=1}^N \frac{\partial}{\partial t}\frac{d\nabla f_i(x,t)}{dt} + \sum_{i=1}^N \frac{d\nabla f_i(x,t)}{dt}\Big) - \sum_{i=1}^N H_i(x,t)\sum_{i=1}^N \nabla f_i(x,t) \\ &+ \Big[\Big(\sum_{i=1}^N H_i(x,t)\Big)^{-1}\sum_{i=1}^N \frac{d}{dt}H_i(x,t)\sum_{i=1}^N H_i(x,t)\Big]\Big(\sum_{i=1}^N \frac{\partial\nabla f_i(x,t)}{\partial t} + \sum_{i=1}^N \nabla f_i(x,t)\Big) \end{aligned} \tag{3.27}$$

for (3.24), the function $f(x,t)$ is minimized. Unfortunately, (3.27) is a centralized solution relying on the knowledge of all $f_i(x,t), i = 1, ..., N$. In Subsections 3.2.2 and 3.2.3, (3.27) will be exploited to propose two algorithms for solving the time-varying convex optimization problem for double-integrator dynamics in a distributed manner.

3.2.2 Distributed Time-Varying Convex Optimization Using Neighbors' Positions and Velocities

In what follows, we focus on solving the distributed time-varying convex optimization problem (3.7) for agents with double-integrator dynamics

$$\dot{x}_i(t) = v_i(t) \quad \dot{v}_i(t) = u_i(t), \tag{3.28}$$

where $x_i, v_i \in \mathbb{R}^m$ are, respectively, the position and velocity of agent i, and $u_i \in \mathbb{R}^m$ is the control input of agent i. In this subsection, an algorithm with adaptive gains will be proposed to solve this problem, where each agent has access to only its own position and the relative positions and velocities between itself and its neighbors. The control input is proposed for (3.28) as

$$\begin{aligned} u_i =& -\sum_{j\in N_i} \mu(x_i - x_j) + \alpha(v_i - v_j) - \sum_{j\in N_i} \beta_{ij}\mathrm{sgn}(\gamma(x_i - x_j) + \zeta(v_i - v_j)) + \phi_i \\ \dot{\beta}_{ij} =& \left\|\gamma(x_i - x_j) + \zeta(v_i - v_j)\right\|_1, \;\; j \in N_i, \end{aligned} \tag{3.29}$$

where

$$\begin{aligned} \phi_i =& -H_i^{-1}(x_i,t)\big(\frac{\partial}{\partial t}\frac{d\nabla f_i(x_i,t)}{dt} + \frac{d\nabla f_i(x_i,t)}{dt}\big) - H_i(x_i,t)\nabla f_i(x_i,t) \\ &+ \left[H_i^{-1}(x_i,t)(\frac{d}{dt}H_i(x_i,t))H_i^{-1}(x_i,t)\right]\big(\frac{\partial\nabla f_i(x_i,t)}{\partial t} + \nabla f_i(x_i,t)\big), \end{aligned} \tag{3.30}$$

where μ, α, γ and ζ are positive coefficients, and β_{ij} is a varying gain with $\beta_{ij}(0) = \beta_{ji}(0) \geq 0$. Note that ϕ_i depends on only agent i's position and velocity, furthermore all terms in above expression are assumed to exist. Define agent i's position and velocity consensus error as, respectively, $e_{X_i} = x_i - \frac{1}{N}\sum_{\ell=1}^{N} x_\ell$ and $e_{V_i} = v_i - \frac{1}{N}\sum_{\ell=1}^{N} v_\ell$. Let $X = [x_1^T, x_2^T, ..., x_N^T]^T$ and $V = [v_1^T, v_2^T, ..., v_N^T]^T$. Define the consensus error vectors for position and velocity as

$$e_X(t) = (\Pi \otimes I_m)X, \; e_V(t) = (\Pi \otimes I_m)V. \tag{3.31}$$

As discussed in Subsection 3.1.2, it is easy to see that $e_X(t) = 0, e_V(t) = 0$ if and only if $x_i = x_j, v_i = v_j,\ \forall i, j \in \mathcal{I}$. Also let $\Phi = [\phi_1^T, \phi_2^T, ..., \phi_N^T]^T$.

Assumption 26 *With ϕ_i defined in* (3.30)*, there exists a positive constant $\bar{\phi}$ such that $\|\phi_i - \phi_j\| \leq \bar{\phi},\ \forall i, j \in \mathcal{I}$, and $\forall t$.*

In Proposition 2, we will show that the agents reach consensus using (3.29). Then this result will be used in Theorem 27 to show that the agents minimize the team cost function as $t \to \infty$.

Proposition 2 *Suppose that the graph $\mathcal{G}$ is connected, and Assumption 26 and $\frac{\gamma}{\alpha\zeta} < \lambda_2[L]$ hold. System* (3.28) *with the algorithm (3.29) reaches consensus, i.e, $x_i = x_j, v_i = v_j,\ \forall i, j \in \mathcal{I}$, as $t \to \infty$.*

Proof: The closed-loop system (3.28) with the control input (3.29) can be recast into a compact form as

$$\begin{cases} \dot{X} = & V \\ \dot{V} = & -(L \otimes I_m)(\mu X + \alpha V) + \Phi \\ & -(D' \otimes I_m)\text{sgn}\big[(D^T \otimes I_m)(\gamma X + \zeta V)\big], \end{cases} \tag{3.32}$$

where D' is defined in Definition 16, and D and L are defined in Section 1.2. Now, we can rewrite (3.32) as

$$\begin{cases} \dot{e}_X = & e_V \\ \dot{e}_V = & -(L \otimes I_m)(\mu e_X + \alpha e_V) + (\Pi \otimes I_m)\Phi \\ & -(D' \otimes I_m)\text{sgn}\big[(D^T \otimes I_m)(\gamma X + \zeta V)\big]. \end{cases} \tag{3.33}$$

Define the function

$$W = \frac{1}{2}\begin{pmatrix} e_X \\ \\ e_V \end{pmatrix}^T P \begin{pmatrix} e_X \\ \\ e_V \end{pmatrix} + \frac{1}{2}\sum_{i=1}^{N}\sum_{j\in N_i}(\beta_{ij}-\bar{\beta})^2, \tag{3.34}$$

where $P = \begin{pmatrix} (\alpha\gamma+\mu\zeta)(L\otimes I_m) & \gamma I_{mN} \\ \\ \gamma I_{mN} & \zeta I_{mN} \end{pmatrix}$. To prove the positive definiteness of P, we define $\hat{P} = \begin{pmatrix} (\alpha\gamma+\mu\zeta)(\lambda_2[L]I_{mN}) & \gamma I_{mN} \\ \\ \gamma I_{mN} & \zeta I_{mN} \end{pmatrix}$. By using Lemma 1, we obtain that $\hat{P} \leq P$ and we just need to show $\hat{P} > 0$. Now, applying Lemma 12, $\hat{P} > 0$ if $\zeta(\alpha\gamma+\mu\zeta)\lambda_2[L]I_N - \gamma^2 I_N > 0$, which is equivalent to $\frac{\gamma}{\alpha\zeta} < \lambda_2[L]$.

The time derivative of W along (3.33) can be obtained as

$$\begin{aligned}
\dot{W} =& -\gamma\mu e_X^T(L\otimes I_m)e_X + e_V^T(\gamma I_{mN} - \alpha\zeta L\otimes I_m)e_V \\
& - \big(\gamma e_X + \zeta e_V\big)^T\Big\{(D'\otimes I_m)\mathrm{sgn}\big[(D^T\otimes I_m)(\gamma e_X + \zeta e_V)\big]\Big\} \\
& + \big(\gamma e_X + \zeta e_V\big)^T(\Pi\otimes I_m)\Phi + \frac{1}{2}\sum_{i=1}^{N}\sum_{j\in N_i}(\beta_{ij} - \bar{\beta})\dot{\beta}_{ij} \\
& = -\gamma\mu e_X^T(L\otimes I_m)e_X + e_V^T(\gamma I_{mN} - \alpha\zeta L\otimes I_m)e_V \\
& - \frac{1}{2}\sum_{i=1}^{N}\sum_{j\in N_i}\beta_{ij}\left\|\gamma(e_{X_i} - e_{X_j}) + \zeta(e_{V_i} - e_{V_j})\right\|_1 \\
& + \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}\big[\gamma(e_{X_i} - e_{X_j}) + \zeta(e_{V_i} - e_{V_j})\big]\phi_i + \frac{1}{2}\sum_{i=1}^{N}\sum_{j\in N_i}(\beta_{ij} - \bar{\beta})\dot{\beta}_{ij} \\
& \le -\gamma\mu e_X^T(L\otimes I_m)e_X + e_V^T(\gamma I_{mN} - \alpha\zeta L\otimes I_m)e_V \\
& - \frac{1}{2}\sum_{i=1}^{N}\sum_{j\in N_i}\beta_{ij}\left\|\gamma(e_{X_i} - e_{X_j}) + \zeta(e_{V_i} - e_{V_j})\right\|_1 \\
& + \frac{1}{2N}\sum_{i=1}^{N}\sum_{j=1}^{N}\left\|\gamma(e_{X_i} - e_{X_j}) + \zeta(e_{V_i} - e_{V_j})\right\|_1\|\phi_i - \phi_j\| \\
& + \frac{1}{2}\sum_{i=1}^{N}\sum_{j\in N_i}(\beta_{ij} - \bar{\beta})\left\|\gamma(e_{X_i} - e_{X_j}) + \zeta(e_{V_i} - e_{V_j})\right\|_1 \\
& \le -\gamma\mu e_X^T(L\otimes I_m)e_X + e_V^T(\gamma I_{mN} - \alpha\zeta L\otimes I_m)e_V \\
& + (\frac{(N-1)\bar{\phi}}{4} - \frac{\bar{\beta}}{2})\sum_{i=1}^{N}\sum_{j\in N_i}\left\|\gamma(e_{X_i} - e_{X_j}) + \zeta(e_{V_i} - e_{V_j})\right\|,
\end{aligned}$$

where the last inequality is obtained because $\mathcal{G}$ is connected and Assumption 26 holds. The term $e_V^T(\gamma I_{mN} - \alpha\zeta L\otimes I_m)e_V < 0$, if $\gamma I_N - \alpha\zeta L < 0$. By applying Lemma 1, we know that $\gamma I_N - \alpha\zeta L < 0$ holds if $\gamma - \alpha\zeta\lambda_2[L] < 0$. Select $\bar{\beta}$ such that $\bar{\beta} \ge \frac{(N-1)\bar{\phi}}{2}$. Using an argument similar to (3.13), we have

$$
\begin{aligned}
\dot{W} &\leq (\frac{(N-1)\bar{\phi}}{4} - \frac{\bar{\beta}}{2}) \sum_{i=1}^{N} \sum_{j \in N_i} \left\| \gamma(e_{X_i} - e_{X_j}) + \zeta(e_{V_i} - e_{V_j}) \right\|_1 \\
&= (\frac{(N-1)\bar{\phi}}{2} - \bar{\beta}) \big(\gamma e_X + \zeta e_V\big)^T \Big((D' \otimes I_m) \mathrm{sgn}[(D^T \otimes I_m)(\gamma e_X + \zeta e_V)] \Big) \\
&\leq (\frac{(N-1)\bar{\phi}}{2} - \bar{\beta}) \left\| (D^T \otimes I_m)(\gamma e_X + \zeta e_V) \right\|_1 \\
&\leq (\frac{(N-1)\bar{\phi}}{2} - \bar{\beta}) \sqrt{\lambda_2[L]} \, \|(\gamma e_X + \zeta e_V)\|_2 < 0,
\end{aligned}
\tag{3.35}
$$

where in the last inequality the fact that $L = DD^T$ and Lemma 1 have been used. Therefore, having $W \geq 0$ and $\dot{W} \leq 0$, we can conclude that $e_X, e_V \in \mathcal{L}_\infty$. By integrating both sides of (3.35), we can see that $e_X, e_V \in \mathcal{L}_2$. Now, applying Barbalat's Lemma [72], we obtain that e_X and e_V will asymptotically converge to zero when $t \to \infty$ and hence the agents reach consensus as $t \to \infty$. ■

Theorem 27 *Suppose that the graph $\mathcal{G}$ is connected, Assumptions 8, 10 and 26 hold. If $H_i(x_i, t) = H_j(x_j, t)$, $\forall i, j \in \mathcal{I}$, and $\frac{\gamma}{\alpha\zeta} < \lambda_2[L]$ hold, by employing the algorithm (3.29) for the system (3.28), the optimization goal* (3.8) *is achieved.*

Proof: Define the Lyapunov function candidate

$$
W = \frac{1}{2}(\sum_{j=1}^{N} \nabla f_j(x_j, t))^T(\sum_{j=1}^{N} \nabla f_j(x_j, t)) + \frac{1}{2}(\sum_{j=1}^{N} v_j - \sum_{j=1}^{N} S_j)^T(\sum_{j=1}^{N} v_j - \sum_{j=1}^{N} S_j), \tag{3.36}
$$

where $S_j = H_j^{-1}(x_j, t)\big(\frac{\partial}{\partial t}\nabla f_j(x_j, t) + \nabla f_j(x_j, t)\big)$. The time derivative of W along with the system defined in (3.28) and (3.29) can be obtained as

$$\begin{aligned}\dot{W} =&(\sum_{j=1}^{N} \nabla f_j(x_j,t))^T(\sum_{j=1}^{N} H_j(x_j,t)v_j + \sum_{j=1}^{N} \frac{\partial}{\partial t}\nabla f_j(x_j,t)) + (\sum_{j=1}^{N} v_j - \sum_{j=1}^{N} S_j)^T(\sum_{j=1}^{N} \dot{v}_j - \sum_{j=1}^{N} \dot{S}_j)\\ =&(\sum_{j=1}^{N} \nabla f_j(x_j,t))^T(\sum_{j=1}^{N} H_j(x_j,t)v_j + \sum_{j=1}^{N} \frac{\partial}{\partial t}\nabla f_j(x_j,t)) + (\sum_{j=1}^{N} v_j - \sum_{j=1}^{N} S_j)^T(\sum_{j=1}^{N} \phi_j - \sum_{j=1}^{N} \dot{S}_j)\\ =&(\sum_{j=1}^{N} \nabla f_j(x_j,t))^T(\sum_{j=1}^{N} H_j(x_j,t)v_j + \sum_{j=1}^{N} \frac{\partial}{\partial t}\nabla f_j(x_j,t))\\ &-(\sum_{j=1}^{N} v_j - \sum_{j=1}^{N} S_j)^T(\sum_{j=1}^{N} H_j(x_j,t)\nabla f_j(x_j,t)),\end{aligned}$$

where in the second equality, we have used the fact that $\mathcal{G}$ is undirected and hence by summing both sides of the closed-loop system (3.28) with the controller (3.29), we have $\sum_{j=1}^{N} \dot{v}_j = \sum_{j=1}^{N} \phi_j$. Now, under the assumption of identical Hessians, we have

$$\begin{aligned}\dot{W} =&(\sum_{j=1}^{N} \nabla f_j(x_j,t))^T\big(H_j(x_j,t)\sum_{j=1}^{N} v_j + \sum_{j=1}^{N} \frac{\partial}{\partial t}\nabla f_j(x_j,t)\big)\\ &-(\sum_{j=1}^{N} v_j - \sum_{j=1}^{N} S_j)^T\big(H_j(x_j,t)\sum_{j=1}^{N} \nabla f_j(x_j,t)\big)\\ =&-(\sum_{j=1}^{N} \nabla f_j(x_j,t))^T(\sum_{j=1}^{N} \nabla f_j(x_j,t)).\end{aligned}$$

Therefore, $\dot{W} < 0$ for $\sum_{j=1}^{N} \nabla f_j(x_j,t) \neq 0$. Now, having $W \geq 0$ and $\dot{W} \leq 0$, we can conclude that $\sum_{j=1}^{N} \nabla f_j(x_j,t), \big(\sum_{j=1}^{N} v_j - \sum_{j=1}^{N} S_j\big) \in \mathcal{L}_\infty$. By integrating both sides of $\dot{W} = -(\sum_{j=1}^{N} \nabla f_j(x_j,t))^T(\sum_{j=1}^{N} \nabla f_j(x_j,t))$, we can see that $\sum_{j=1}^{N} \nabla f_j(x_j,t) \in \mathcal{L}_2$. Now, applying Barbalat's Lemma, we obtain that that $\sum_{j=1}^{N} \nabla f_j(x_j,t)$ will asymptomatically converge to zero. We also have $x_i = x_j, v_i = v_j,\ \forall i,j \in \mathcal{I}$ as $t \to \infty$ from Proposition 2. Now, under the assumption that $f(x,t)$ is convex and using Lemma 11 and under Assumption 8, it is easy to see that as $t \to \infty$, $\sum_{j=1}^{N} f_i(x_j,t)$ will be minimized, where $x_i = x_j, v_i = v_j,\ \forall i,j \in \mathcal{I}$. ■

Remark 28 *In Appendix .1, we show that Assumption 26 holds if the cost functions with identical Hessians satisfy certain conditions such that the boundedness of* $\|x_i(t) - x_j(t)\|$ *and* $\|v_i(t) - v_j(t)\|$

for all t guarantees the boundedness of $\|\nabla f_j(x_j,t) - \nabla f_i(x_i,t)\|$, $\left\|\frac{d\nabla f_j(x_j,t)}{dt} - \frac{d\nabla f_i(x_i,t)}{dt}\right\|$ and $\left\|\frac{\partial^2 \nabla f_j(x_j,t)}{\partial t^2} - \frac{\partial^2 \nabla f_i(x_i,t)}{\partial t^2}\right\|$ for all t. For example, for the cost functions $f_i(x_i(t),t) = (ax_i(t) + g_i(t))^2$ introduced in Remark 20, the boundedness of $\|x_i(t) - x_j(t)\|$ and $\|v_i(t) - v_j(t)\|$ for all t guarantees the boundedness of $\|\nabla f_j(x_j,t) - \nabla f_i(x_i,t)\|$, $\left\|\frac{d\nabla f_j(x_j,t)}{dt} - \frac{d\nabla f_i(x_i,t)}{dt}\right\|$ and $\left\|\frac{\partial^2 \nabla f_j(x_j,t)}{\partial t^2} - \frac{\partial^2 \nabla f_i(x_i,t)}{\partial t^2}\right\|$ for all t, if $\|g_i(t) - g_j(t)\|$, $\|\dot{g}_i(t) - \dot{g}_j(t)\|$ and $\|\ddot{g}_i(t) - \ddot{g}_j(t)\|$ are bounded. Hence Assumption 26 holds for $f_i(x_i(t),t) = (ax_i(t) + g_i(t))^2$, if $\|g_i(t) - g_j(t)\|$, $\|\dot{g}_i(t) - \dot{g}_j(t)\|$ and $\|\ddot{g}_i(t) - \ddot{g}_j(t)\|$, $\forall t$ and $\forall i,j \in \mathcal{I}$ are bounded.

Remark 29 *The result in Theorem 4.2 can be extended to a class of cost functions whose Hessians have the same structure rather than being identical under a certain additional assumption. Particularly, the assumption that $H_i(x_i,t) = H_j(x_j,t), \forall t$ and $\forall i,j \in \mathcal{I}$, can be replaced with $H_i(z,t) = H_j(z,t)$, $\forall z,t$ and $\forall i,j \in \mathcal{I}$, with an additional assumption that $\|\nabla f_i(x_i,t)\|$ and $\left\|\frac{\partial}{\partial t}\nabla f_i(x_i,t)\right\|$, $\forall i \in \mathcal{I}$, and $\forall t$ are bounded.*

To relax the assumption on Hessians, an estimator-based algorithm will be introduced in Subsection 3.2.3, where the agents can have cost functions with nonidentical Hessians.

3.2.3 Estimator-Based Distributed Time-Varying Convex Optimization

In this subsection, an estimator-based algorithm is designed to minimize the time-varying convex optimization (3.7) for double-integrator dynamics (3.28). In this algorithm, each agent calculates (3.27) in a distributed manner. Similar to Subsection 3.1.3, distributed average tracking is used as a tool to estimate the unknown variables in (3.27). Each agent generates an estimate of (3.27). Then using control input $u_i(t)$, each agent tracks its estimated signal while reaching consensus. The proposed algorithm for the system (3.28) is given by

$$\dot{\xi}_i = \kappa \sum_{j \in N_i(t)} \text{sgn}(w_j - w_i), \quad w_i = \xi_i + \psi_i \tag{3.37}$$

$$\dot{\phi}_i = \rho \sum_{j \in N_i(t)} \text{sgn}(\varsigma_j - \varsigma_i), \quad \varsigma_i = \phi_i + \theta_i \tag{3.38}$$

$$S_i = \varsigma_{i1}^{-1} \varsigma_{i2} \varsigma_{i1} \big(w_{i1} + w_{i2}\big) - \varsigma_{i1}^{-1} \big(w_{i3} + w_{i4}\big) - \varsigma_{i1} w_{i1} \tag{3.39}$$

$$u_i = -\sum_{j \in N_i(t)} \text{sig}(x_i - x_j)^{\alpha_1} - \sum_{j \in N_i(t)} \text{sig}(v_i - v_j)^{\alpha_1} + S_i, \tag{3.40}$$

where

$$0 < \alpha_1 < 1 \quad \alpha_2 = \frac{2\alpha_1}{\alpha_1 + 1} \tag{3.41}$$

and

$$\psi_i = \begin{pmatrix} \nabla f_i(x_i, t) \\ \frac{\partial}{\partial t}(\nabla f_i(x_i, t)) \\ \frac{d}{dt}(\nabla f_i(x_i, t)) \\ \frac{\partial}{\partial t}(\frac{d}{dt} \nabla f_i(x_i, t)) \end{pmatrix}, \; \theta_i = \begin{pmatrix} \nabla^2 f_i(x_i, t) \\ \frac{d}{dt}(\nabla^2 f_i(x_i, t)) \end{pmatrix},$$

and κ and ρ are positive constant coefficients to be selected. Eqs. (3.37) and (3.38) are distributed average tracking estimators, where the estimated variables w_i and ς_i can be redefined as $w_i^T = (w_{i1}^T, \dots, w_{i4}^T)$ and $\varsigma_i^T = (\varsigma_{i1}^T, \varsigma_{i2}^T)$ with $w_{ij} \in \mathbb{R}^m, \varsigma_{ik} \in \mathbb{R}^{m \times m}, \forall i \in \mathcal{I}, j = 1, ..., 4, k = 1, 2$. The initial values of the internal states ξ_i and ϕ_i are chosen such that[2]

$$\sum_{j=1}^{N} \xi_j(0) = \sum_{j=1}^{N} \phi_j(0) = 0. \tag{3.42}$$

[2]As a special case the initial values can be chosen as $\xi_j(0) = \phi_j(0) = 0, \forall j \in \mathcal{I}$.

Assumption 30 *The coefficients κ, and ρ, satisfy the following inequalities: $\kappa > \sup_t \|\psi_i\|_\infty$ and $\rho > \sup_t \|\theta_i\|_\infty$, $\forall i \in \mathcal{I}$.*

These assumptions can be satisfied if the graph $\mathcal{G}(t)$ is connected $\forall t$, the gradients, the derivatives of the Hessians and gradients, and the partial derivatives of the gradients' derivatives are bounded. Although these assumptions seem restrictive, they can be satisfied for many cost functions. For example, for $f_i(x_i(t), t) = (a_i x_i(t) + g_i(t))^2$ introduced in Remark 23, the above conditions are equivalent to the conditions that $\|g_i(t)\|$, $\|\dot{g}_i(t)\|$, and $\|\ddot{g}_i(t)\|$ are bounded.

Theorem 31 *Suppose that the graph $\mathcal{G}(t)$ is connected for all t and Assumptions 8 and 9 hold. If Assumption 30 and the initial condition* (3.42) *hold, by employing the algorithm* (3.37)-(3.40) *for the system (3.28), the optimization goal* (3.8) *is achieved.*

Proof: **Estimator**: It follows from Theorem 2 in [13] that if $\kappa > \sup_t \|\psi_i\|_\infty$, $\rho > \sup_t \|\theta_i\|_\infty$, $\forall i \in \mathcal{I}$ and the graph $\mathcal{G}(t)$ is connected for all t, then employing (3.37) and (3.38), there exists a $T > 0$ such that for all $t \geq T$, we have $\left\|w_i - \frac{1}{N}\sum_{i=1}^N \psi_i\right\|_2 = 0$, $\left\|\varsigma_i - \frac{1}{N}\sum_{i=1}^N \theta_i\right\|_2 = 0$. Now from (3.39), the estimated signal S_i satisfies

$$\begin{aligned} S_i = &-\sum_{i=1}^N H_i(x_i,t)\sum_{i=1}^N \nabla f_i(x_i,t) - (\sum_{i=1}^N H_i(x_i,t))^{-1}(\sum_{i=1}^N \frac{\partial}{\partial t}\frac{d\nabla f_i(x_i,t)}{dt} + \sum_{i=1}^N \frac{d\nabla f_i(x_i,t)}{dt}) \\ &+ \Big((\sum_{i=1}^N H_i(x_i,t))^{-1}\sum_{i=1}^N \frac{d}{dt}H_i(x_i,t)\sum_{i=1}^N H_i(x_i,t)\Big)(\sum_{i=1}^N \nabla f_i(x_i,t) + \sum_{i=1}^N \frac{\partial \nabla f_i(x_i,t)}{\partial t}), \end{aligned} \tag{3.43}$$

which shows that each agent has an estimate of (3.27), where $S_i = S_j,\ \forall i,j \in \mathcal{I}, \forall t \geq T$.

Controller: Note that for $\forall\ t \geq T,\ S_i = S_j,\ \forall i,j \in \mathcal{I}$, denoted as $\bar{S}$. For $t \geq T$ using (3.40) for (3.28), we have

$$\dot{v}_i = -\sum_{j\in N_i(t)} \mathrm{sig}(x_i - x_j)^{\alpha_1} - \sum_{j\in N_i(t)} \mathrm{sig}(v_i - v_j)^{\alpha_2} + \bar{S} \tag{3.44}$$

For $t \geq T$, rewriting (3.44) using new variables $\tilde{x}_i = x_i - \int_T^t \int_T^t \bar{S} dt$ and $\tilde{v}_i = v_i - \int_T^t \bar{S} dt$, we have

$$\dot{\tilde{v}}_i = - \sum_{j \in N_i(t)} \text{sig}(\tilde{x}_i - \tilde{x}_j)^{\alpha_1} - \sum_{j \in N_i(t)} \text{sig}(\tilde{v}_i - \tilde{v}_j)^{\alpha_2}. \tag{3.45}$$

It is proved in [81] that using (3.45) for the system (3.28), there exists a time T' such that $\tilde{x}_i = \tilde{x}_j,\ \tilde{v}_i = \tilde{v}_j,\ \forall i, j \in \mathcal{I}$. As a result we have $x_i = x_j,\ v_i = v_j,\ \forall i, j \in \mathcal{I}$, and $\dot{v}_i = \bar{S}, \forall\, t \geq T + T'$. Now, it is easy to see that according to (3.27) and (3.43) and Assumption 8, the optimization goal (3.8) is achieved. ■

Remark 32 *In the algorithm* (3.37)-(3.40)*, it is just required that Assumptions 9 and 30 hold, where Assumptions 9 does not necessarily hold for each agent's cost function $f_i(x(t), t)$. However, each agent must communicate two variables $\psi_i \in \mathbb{R}^{4m}$ and $\theta_i \in \mathbb{R}^{2m \times m}$ with its neighbors which necessitates the communication requirement. On other hand, for the algorithm* (3.29)*, not only Assumptions 10 and the conditions in Remark 28 have to be satisfied for each individual function $f_i(x_i(t), t)$, it requires the agents' Hessians to be equal. In spite of these restrictive assumptions, using the algorithm* (3.29)*, we can eliminate the necessity of communication between agents when the relative positions and the velocities between each agent and its neighbors can be obtained by sensing.*

In what follows we will study how to overcome the possible chattering effect of implementing signum function in the algorithm (3.29). In Subsections 3.2.4 and 3.2.5, two continuous control algorithms will be proposed to extend the controller (3.29).

3.2.4 Distributed Time-Varying Convex Optimization Using Time-Varying Approximation of Signum Function

In this subsection, we focus on distributed time-varying convex optimization for double-integrator dynamics (3.28), where a continuous control algorithm based on the boundary layer concept will be introduced. Using a continuous approximation of the signum function will reduce chattering in real applications and will make the controller easier to implement. In this algorithm, each agent needs to know its own position, velocity and the relative positions and velocities between itself and its neighbors. Define the nonlinear function $h(\cdot)$ as

$$h(z) = \frac{z}{\|z\| + \epsilon e^{-\xi t}}, \tag{3.46}$$

where ξ and ϵ are positive coefficients and $z \in \mathbb{R}^m$. The nonlinear function $h(z)$ is a continuous approximation, using the boundary layer concept [24], of the discontinuous function sgn$(\cdot)$. The size of boundary layer, $\epsilon e^{-\xi t}$ is time-varying and as $t \to \infty$ the continuous function $h(z)$ approaches the signum function. The idea of using this continuous approximation is borrowed from [86].

By replacing the signum function with a continuous approximation (3.46), the results presented in Subsection 3.2.2 are not valid anymore and it is not clear whether the new algorithm works. The reason is that the results (and the proofs) in Subsection 3.2.2 build upon the property of the ideal discontinuous signum function, which switches instantaneously at 0, so that it can compensate for the effect of the inconsistent internal time-varying optimization signals among the agents so that the agents can reach consensus. However, this can no longer be achieved by its continuous replacement and further careful analysis is needed. The results and the proofs presented in this subsection are not just a simple replacement of the signum function with its approximation.

Here in particular we show that with the signum function replaced with the time-varying continuous approximation, (3.46), it is possible to still achieve distributed optimization with zero error under certain different assumptions and conditions. The reason is that (3.46) approaches the signum function as $t \to \infty$.

The continuous control input with adaptive gains is proposed for (3.28) as

$$\begin{aligned} u_i =& - \sum_{j \in N_i} \mu(x_i - x_j) + \alpha(v_i - v_j) - \sum_{j \in N_i} \beta_{ij}\, h[\gamma(x_i - x_j) + \zeta(v_i - v_j)] + \phi_i, \\ \dot{\beta}_{ij} =& \big[\gamma(x_i - x_j) + \zeta(v_i - v_j)\big] h[\gamma(x_i - x_j) + \zeta(v_i - v_j)], \end{aligned} \tag{3.47}$$

where μ, α, γ and ζ are positive coefficients, β_{ij} is a varying gain with $\beta_{ij}(0) = \beta_{ji}(0) \geq 0$, and ϕ_i is defined as in (3.29).

Theorem 33 *Suppose that the graph $\mathcal{G}$ is connected, and*

$$\frac{\gamma}{\alpha\zeta} + \frac{\psi}{\alpha} < \lambda_2[L] \tag{3.48}$$

$$\frac{\mu}{2\alpha} > \psi \tag{3.49}$$

$$\frac{\gamma}{2\zeta} > \psi \tag{3.50}$$

hold, where $\psi > 0$ is a parameter to be selected. If Assumptions 8, 10 and 26 hold and $H_i(x_i, t) = H_j(x_j, t), \forall t$ and $\forall i, j \in \mathcal{I}$, by employing the algorithm (3.47) for the system (3.28), the optimization goal (3.8) *is achieved.*

Proof: Define e_X and e_V as in (3.31) and y as $y^T = (y_1^T, \ldots, y_N^T) = \gamma e_X^T + \zeta e_V^T$, with $y_i \in \mathbb{R}^m$.

Rewriting the closed-loop system (3.28) with the (3.47) in terms of the consensus errors, we have

$$\begin{cases} \dot{e}_X = & e_V \\ \dot{e}_V = & -(L \otimes I_m)(\mu e_X + \alpha e_V) \\ & - \begin{pmatrix} \sum_{j \in N_1} \beta_{1j} h\big(y_1 - y_j\big) \\ \vdots \\ \sum_{j \in N_N} \beta_{Nj} h\big(y_N - y_j\big) \end{pmatrix} + (\Pi \otimes I_m)\Phi. \end{cases} \tag{3.51}$$

Define the function

$$W = \frac{1}{2} \begin{pmatrix} e_X \\ e_V \end{pmatrix}^T P \begin{pmatrix} e_X \\ e_V \end{pmatrix} + \frac{1}{2} \sum_{i=1}^{N} \sum_{j \in N_i} (\beta_{ij} - \bar{\beta})^2, \tag{3.52}$$

where $P = \begin{pmatrix} -2\psi\gamma I_{Nm} + (\alpha\gamma + \mu\zeta)(L \otimes I_m) & \gamma I_{mN} \\ \gamma I_{mN} & \zeta I_{mN} \end{pmatrix}$. To prove the positive definiteness of P, we also define $\hat{P} = \begin{pmatrix} -2\psi\gamma I_{mN} + (\alpha\gamma + \mu\zeta)(\lambda_2[L] I_{mN}) & \gamma I_{mN} \\ \gamma I_{mN} & \zeta I_{mN} \end{pmatrix}$. By using Lemma 1, we obtain that $\hat{P} \leq P$ and we just need to show $\hat{P} > 0$. Using Lemma 12, we know that $\hat{P} > 0$ if

$$\begin{aligned} & -2\psi\gamma I_N + (\alpha\gamma + \mu\zeta)\lambda_2[L]) I_N > 0 \\ & -2\zeta\psi\gamma I_N + \zeta(\alpha\gamma + \mu\zeta)\lambda_2[L] - \gamma^2 I_N > 0. \end{aligned} \tag{3.53}$$

Now, Using conditions (3.48) and (3.49), respectively, we have

$$\begin{aligned} & -2\psi\gamma I_N + (\alpha\gamma + \mu\zeta)\lambda_2[L] I_N > -2\psi\gamma I_N + \frac{(\alpha\gamma + \mu\zeta)\gamma}{\alpha\zeta} I_N \\ & = \frac{\gamma}{\alpha\zeta}(-2\psi\alpha\zeta + \zeta\mu + \alpha\gamma^2) I_N > \frac{\gamma}{\alpha\zeta}(-\zeta\mu + \zeta\mu + \alpha\gamma) I_N = \frac{\gamma^2}{\zeta} > 0 \end{aligned}$$

which guarantees that the first inequality in (3.53) holds. Applying a similar procedure we have

$$
\begin{aligned}
&-2\zeta\psi\gamma I_N+\zeta(\alpha\gamma+\mu\zeta)\lambda_2[L]-\gamma^2 I_N\\
&>-2\zeta\psi\gamma I_N+(\alpha\gamma+\mu\zeta)\frac{\gamma}{\alpha}I_N-\gamma^2 I_N=\frac{\zeta\gamma}{\alpha}(-2\alpha\psi+\mu)I_N>0.
\end{aligned}
$$

Hence W is positive definite.

The time derivative of W along (3.51) can be obtained as

$$
\begin{aligned}
\dot{W}=&-\gamma\mu e_X^T(L\otimes I_m)e_X+e_V^T(\gamma I_{mN}-\alpha\zeta L\otimes I_m)e_V-2\psi\gamma e_X^T e_V\\
&-y^T\begin{pmatrix}\sum_{j\in N_1}\beta_{1j}h\big(y_1-y_j\big)\\ \vdots\\ \sum_{j\in N_N}\beta_{Nj}h\big(y_N-y_j\big)\end{pmatrix}+y^T(\Pi\otimes I_m)\Phi+\frac{1}{2}\sum_{i=1}^{N}\sum_{j\in N_i}(\beta_{ij}-\bar{\beta})\dot{\beta}_{ij}.
\end{aligned}
\tag{3.54}
$$

We rewrite (3.54) as

$$
W=\bar{W}-\sum_{i=1}^{N}\sum_{j\in N_i}\beta_{ij}y_i h(y_i-y_j)+y^T(\Pi\otimes I_m)\Phi+\frac{1}{2}\sum_{i=1}^{N}\sum_{j\in N_i}(\beta_{ij}-\bar{\beta})\dot{\beta}_{ij},
$$

where $\bar{W}=\begin{pmatrix}e_X\\ e_V\end{pmatrix}^T\bar{P}\begin{pmatrix}e_X\\ e_V\end{pmatrix}$ and $\bar{P}=\begin{pmatrix}-\mu\gamma(L\otimes I_m) & -\psi\gamma I_{mN}\\ -\psi\gamma I_{mN} & \gamma I_{mN}-\alpha\zeta(L\otimes I_m)\end{pmatrix}$. Because the graph $\mathcal{G}$ is connected, we have

$$
\begin{aligned}
\dot{W}=&\bar{W}-\frac{1}{2}\sum_{i=1}^{N}\sum_{j\in N_i}\beta_{ij}(y_i-y_j)h(y_i-y_j)\\
&+\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}(y_i-y_j)\phi_i+\frac{1}{2}\sum_{i=1}^{N}\sum_{j\in N_i}(\beta_{ij}-\bar{\beta})(y_i-y_j)h(y_i-y_j)\\
&=\bar{W}+\frac{1}{2N}\sum_{i=1}^{N}\sum_{j=1}^{N}(y_i-y_j)(\phi_i-\phi_j)-\frac{\bar{\beta}}{2}\sum_{i=1}^{N}\sum_{j\in N_i}(y_i-y_j)h(y_i-y_j)\\
&\le\bar{W}+\frac{1}{2N}\sum_{i=1}^{N}\sum_{j=1}^{N}\|y_i-y_j\|\,\|\phi_i-\phi_j\|-\frac{\bar{\beta}}{2}\sum_{i=1}^{N}\sum_{j\in N_i}\frac{\|y_i-y_j\|^2}{\|y_i-y_j\|+\epsilon e^{-\xi t}}\\
&\le\bar{W}+\frac{(N-1)\bar{\phi}}{4}\sum_{i=1}^{N}\sum_{j\in N_i}\|y_i-y_j\|-\frac{\bar{\beta}}{2}\sum_{i=1}^{N}\sum_{j\in N_i}\frac{\|y_i-y_j\|^2}{\|y_i-y_j\|+\epsilon e^{-\xi t}},
\end{aligned}
$$

where in the last inequality Assumption 26 is used. Selecting a $\bar{\beta}$ such that $\bar{\beta} \geq \frac{(N-1)\bar{\phi}}{2}$, we obtain

$$\dot{W} < \bar{W} + \frac{\bar{\beta}}{2} \sum_{i=1}^{N} \sum_{j \in N_i} \epsilon e^{-\xi t}. \tag{3.55}$$

If we can show $\bar{W} < -\psi W$ (or equivalently $\bar{W} + \psi W < 0$) holds, then knowing that $e^{-\xi t} \to 0$ as $t \to \infty$, Lemma 2.19 in [59] implies that the system (3.51) is asymptotically stable. Note that $\bar{W} + \psi W =$

$$\begin{pmatrix} (-\gamma\mu + \alpha\gamma\psi + \zeta\mu\psi)L - 2\psi^2\gamma I_N & 0 \\ 0 & (\gamma + \zeta\psi)I_N - \alpha\zeta L \end{pmatrix}. \tag{3.56}$$

Applying Lemma 12, we obtain that (3.56) is negative definite if

$$(-\gamma\mu + \alpha\gamma\psi + \zeta\mu\psi)L - 2\psi^2\gamma I_N < 0(\gamma + \zeta\psi)I_N - \alpha\zeta L < 0. \tag{3.57}$$

To satisfy the first condition in (3.57), we just need to show $-\gamma\mu + \alpha\gamma\psi + \zeta\mu\psi < 0$. Using conditions (3.49) and (3.50) we have

$$\gamma\mu + \alpha\gamma\psi + \zeta\mu\psi < -\gamma\mu + \frac{\mu\gamma}{2} + \zeta\mu\psi < -\gamma\mu + \frac{\mu\gamma}{2} + \frac{\mu\gamma}{2} < 0.$$

To satisfy the second condition in (3.57), we have

$$(\gamma + \zeta\psi)I_N - \alpha\zeta L < (\gamma + \zeta\psi)I_N - \alpha\zeta\lambda_2[L]I_N < 0,$$

where Lemma 1 and condition (3.48) are employed, respectively. Hence, $\bar{W} < -\psi W$ holds and the agents reach consensus as $t \to \infty$. Now, similar to the proof of Theorem 27, if $H_i(x_i, t) = H_j(x_j, t), \forall t$ and $\forall i, j \in \mathcal{I}$, it can be shown that $\sum_{j=1}^{N} \nabla f_j(x_j, t)$ will asymptotically converge to zero. Now, under Assumption 8 and the assumption that $f(x, t)$ is convex, Lemma 11 is employed. Using the fact that $x_i(t) \to x_j(t), \forall i, j \in \mathcal{I}$ as $t \to \infty$, it is easy to see that the optimization goal (3.8) is achieved. ■

Remark 34 *It is worth mentioning that if we have* $\frac{\gamma}{\alpha\zeta} < \lambda_2[L]$*, there always exists a positive coefficient* ψ *such that conditions* (3.48)-(3.50) *hold. However, selecting* ψ *based on conditions* (3.48)-(3.50) *affects the convergence speed, where by having a larger* ψ *the agents reach consensus faster. To satisfy conditions* (3.48) *and* (3.50)*, it is sufficient to have* $\frac{2\lambda_2[L]\alpha}{3} > \frac{\gamma}{2\zeta} > \psi$ *(e.g, selecting a large* α *and choosing proper* γ, ζ *and* ψ*). It can also be seen that selecting a large enough* μ*,* (3.49) *can be satisfied.*

Remark 35 *The results in Appendix .1 can be modified for Theorem 33, where we can show that Assumption 26 holds under the same conditions mentioned in Remark 28.*

3.2.5 Distributed Time-Varying Convex Optimization Using Time-Invariant Approximation of Signum Function

In this subsection, our focus is on replacing the signum function, with a time-invariant approximation

$$h(z) = \frac{z}{\|z\| + \epsilon}, \tag{3.58}$$

where $\epsilon > 0$. Here, the boundary layer ϵ, is constant. Employing (3.58), instead of (3.46) in the control algorithm (3.47) makes the controller easier to implement in real applications. The trade-off is that the agents will no longer reach consensus, which introduces additional complexities in convergence. Establishing analysis on the optimization error bound in this case is a nontrivial task, which is introduced in this subsection. The reason that the time-invariant continuous approximation (3.58) cannot ensure distributed optimization with zero error is that the global optimal trajectory is not even an equilibrium point of the closed-loop system whenever a time-invariant continuous approximation is introduced. It is worthwhile to mention that if the signum function were replaced with a different time-invariant continuous approximation other than (3.58), there would be no guarantee that the conclusion in this subsection would still hold and further careful analysis would be needed.

Theorem 36 *Suppose that the graph $\mathcal{G}$ is connected, Assumptions 8, 10 and 26 hold and the gradient of the cost functions can be written as $\nabla f_i(x_i,t) = \sigma x_i + g_i(t),\ \forall i \in \mathcal{I}$. If conditions* (3.48)-(3.50) *hold, the system (3.28) with the algorithm (3.47) with $h(\cdot)$ given by* (3.58)*, tracks the optimal trajectory with bounded errors such that as $t \to \infty$*

$$\begin{aligned} \|x_i - x^*\| &< \sqrt{\frac{\bar{\phi}N(N-1)^2\epsilon}{4\psi\lambda_{min}[P]}}, \\ \|v_i - v^*\| &< \sqrt{\frac{\bar{\phi}N(N-1)^2\epsilon}{4\psi\lambda_{min}[P]}},\ \ \forall i \in \mathcal{I}, \end{aligned} \tag{3.59}$$

where x^* and v^* are the position and the velocity of the optimal point and P is defined in (3.52)*. We also have*

$$\lim_{t\to\infty}[\frac{1}{N}\sum_{j=1}^{N} x_i - x^*] = 0, \quad \lim_{t\to\infty}[\frac{1}{N}\sum_{j=1}^{N} v_i - v^*] = 0. \tag{3.60}$$

Proof: The proof will be separated into two parts. In the first part, we show that the consensus error will remain bounded. Then in the second part, we show that the error between the agents' states and the optimal trajectory will remain bounded.

Define the Lyapunov function candidate W as in (3.52). Similar to the proof in Theorem 33, with $h(\cdot)$ given by (3.58) instead of (3.46), we obtain $\dot{W} < \bar{W} + \frac{\bar{\beta}}{2}\sum_{i=1}^{N}\sum_{j\in N_i}\epsilon < -\psi W + \frac{\bar{\beta}}{2}N(N-1)\epsilon$, where $\bar{\beta}$ is selected such that $\bar{\beta} \geq \frac{(N-1)\bar{\phi}}{2}$. Then we have $0 \leq W(t) \leq \frac{\bar{\beta}N(N-1)\epsilon}{2\psi}(1 - e^{-\psi t}) + W(0)e^{-\psi t}$. Therefore, as $t \to \infty$, we have

$$\left\|\begin{pmatrix} e_X \\ e_V \end{pmatrix}\right\|^2 \lambda_{\min}[P] \leq W = \begin{pmatrix} e_X \\ e_V \end{pmatrix}^T P \begin{pmatrix} e_X \\ e_V \end{pmatrix} \leq \frac{\bar{\beta}N(N-1)\epsilon}{2\psi}.$$

Now, it can be seen that there exists a bound on the position and velocity consensus errors as $t \to \infty$, that is,

$$\begin{aligned} \left\| x_i - \frac{1}{N}\sum_{j=1}^{N} x_j \right\| &< \sqrt{\frac{\bar{\phi} N(N-1)^2 \epsilon}{4\psi \lambda_{\min}[P]}}, \\ \left\| v_i - \frac{1}{N}\sum_{j=1}^{N} v_j \right\| &< \sqrt{\frac{\bar{\phi} N(N-1)^2 \epsilon}{4\psi \lambda_{\min}[P]}}, \end{aligned} \tag{3.61}$$

where it is easy to see that by selecting larger ψ satisfying conditions (3.48)-(3.50), the error bound will be smaller.

In what follows, we focus on finding the relation between the optimal trajectory and the agents' states. According to Assumption 8 and using Lemma 11, we know $\sum_{j=1}^{N} \nabla f_j(x^*, t) = 0$. Hence, Under the assumption of $\nabla f_i(x_i, t) = \sigma x_i + g_i(t)$, the optimal trajectory is

$$x^* = \frac{-\sum_{j=1}^{N} g_j}{N\sigma}, \qquad v^* = \frac{-\sum_{j=1}^{N} \dot{g}_j}{N\sigma}. \tag{3.62}$$

Similar to the proof of Theorem 27 we can show that, regardless of whether consensus is reached or not, it is guaranteed that $\sum_{j=1}^{N} \nabla f_j(x_j, t)$ will asymptomatically converge to zero. As a result, we have $\sum_{j=1}^{N} x_i \to \frac{-\sum_{j=1}^{N} g_j}{\sigma}$ and $\sum_{j=1}^{N} v_i \to \frac{-\sum_{j=1}^{N} \dot{g}_j}{\sigma}$. By using (3.62), we can conclude (3.60). It is also shown in (3.61) that the error between the agents' states and the optimal trajectory is bounded and (3.59) holds. ■

Remark 37 *Using the invariant approximation of the signum function* (3.58)*, instead of the time-varying one* (3.46)*, makes the implementation easier in real applications. However, the results show that the team cost function is not exactly minimized and the agents track the optimal trajectory with a bounded error. It also restricts the acceptable cost functions to a class that takes in the form* $\nabla f_i(x_i, t) = \sigma x_i + g_i(t)$.

Remark 38 *The results in Appendix .1 can be modified for Theorem 33, where under the assumption of $\nabla f_i(x_i, t) = \sigma x_i + g_i(t)$, it is easy to show that Assumption 26 holds, if $\|g_i(t) - g_j(t)\|$, $\|\dot{g}_i(t) - \dot{g}_j(t)\|$ and $\|\ddot{g}_i(t) - \ddot{g}_j(t)\|$, $\forall t$ and $\forall i, j \in \mathcal{I}$ are bounded.*

Remark 39 *The algorithms introduced Subsections 3.1.2, 3.1.3, 3.2.2, 3.2.4 and 3.2.5 are still valid in the case of strongly connected weight balanced directed graph $\mathcal{G}$. In our proofs, L can be replaced with symmetric matrix $L + L^T$, as $x^T L x = \frac{1}{2} x^T (L + L^T) x$. Since $\mathcal{G}$ is strongly connected weight balanced, $L + L^T$ is positive semi-definite with a simple eigenvalue at zero. Note that applying the introduced algorithms for directed graphs, we need to redefine λ_2 as the smallest non-zero eigenvalue of matrix $L + L^T$.*

Remark 40 *The algorithms proposed in Subsections 3.1.3, 3.2.3 and Section 3.3 are provided for time-varying graphs. For algorithms introduced in Subsections 3.1.2, 3.2.2, 3.2.4 and 3.2.5, the current results are demonstrated for static graphs. However, the results are valid for time-varying graphs, if the graph $\mathcal{G}(t)$ is connected for all t and a sufficiently large constant gain is used, instead of the time-varying adaptive gains. In particular, the constant gain β should satisfy $\beta > \frac{\|(\Pi \otimes I_m)\Phi\|_2}{\sqrt{\lambda_2[L(t)]}}$. To select such a β, we need to know $\bar{\phi}$, defined in Assumption 17 (or Assumption 26 in the case of double-integrator dynamics), and β_x (and β_v in case of double-integrator dynamics), defined in Appendix .1.*

3.3 Distributed Time-Varying Convex Optimization with Swarm tracking behavior

In this section, we introduce two distributed optimization algorithms with swarm tracking behavior, where the center of the agents tracks the optimal trajectory defined by (3.7) for single-integrator and double-integrator dynamics while the agents avoid inter-agent collisions and maintaining connectivity.

3.3.1 Distributed Optimization with Swarm Tracking for Single-Integrator Dynamics

In this subsection, we focus on distributed optimization problem with swarm tracking behavior for single-integrator dynamics (3.1). To solve this problem, we propose the algorithm

$$u_i(t) = -\beta \text{sgn}\Big(\sum_{j \in N_i(t)} \frac{\partial V_{ij}}{\partial x_i}\Big) + \phi_i, \tag{3.63}$$

where V_{ij} is a potential function between agents i and j to be designed, β is positive, and ϕ_i is defined in (3.9). It is worth mentioning that in this subsection, we assume each agent has a communication/sensing radius R, where if $\|x_i - x_j\| < R$ agent i and j become neighbors. Our proposed algorithm guarantees connectivity maintenance which means that if the graph $\mathcal{G}(0)$ is connected, then for all $t, \mathcal{G}(t)$ will remain connected. Before our main result in this subsection, we need to define the potential function V_{ij}.

Definition 41 *[8] The potential function V_{ij} is a differentiable nonnegative function of $\|x_i - x_j\|$, which satisfies the following conditions*

1) $V_{ij} = V_{ji}$ *has a unique minimum in* $\|x_i - x_j\| = d_{ij}$*, where* d_{ij} *is a desired distance between agents* i *and* j *and* $R > \max_{i,j} d_{ij}$*.*

2) $V_{ij} \to \infty$ *if* $\|x_i - x_j\| \to 0$*.*

3) $V_{ii} = c$*, where* c *is a constant.*

4) $$\begin{cases} \frac{\partial V_{ij}}{\partial(\|x_i - x_j\|)} = 0 & \|x_i(0) - x_j(0)\| \geq R,\ \|x_i - x_j\| \geq R, \\ \frac{\partial V_{ij}}{\partial(\|x_i - x_j\|)} \to \infty & \|x_i(0) - x_j(0)\| < R,\ \|x_i - x_j\| \to R. \end{cases}$$

The motivation of the last condition in Definition 5.1 is to maintain the initially existing connectivity patterns. It guarantees that two agents which are neighbors at $t = 0$, remain neighbors. However, if two agents are not neighbors at $t = 0$, they do not need to be neighbors at $t > 0$. For example, such a potential function V_{ij} is presented in [8].

Theorem 42 *Suppose that graph* $\mathcal{G}(0)$ *is connected, Assumptions 8, and 10 hold and the gradient of the cost functions can be written as* $\nabla f_i(x_i, t) = \sigma x_i + g_i(t),\ \forall i \in \mathcal{I}$*. If* $\beta \geq \|\phi_i\|_1,\ \forall i \in \mathcal{I}$*, for the system* (3.1) *with the algorithm* (3.63)*, the center of the agents tracks the optimal trajectory while the agents maintain connectivity and avoid inter-agent collisions.*

Proof: Define the positive semi-definite Lyapunov function candidate

$$W = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} V_{ij}. \tag{3.64}$$

The time derivative of W is obtained as

$$\dot{W} = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \left(\frac{\partial V_{ij}}{\partial x_i} \dot{x}_i + \frac{\partial V_{ij}}{\partial x_j} \dot{x}_j\right) = \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{\partial V_{ij}}{\partial x_i} \dot{x}_i,$$

where in the second equality, Lemma 3.1 in [8] has been used. Now, rewriting $\dot{W}$ along with the closed-loop system (3.63) and (3.1), we have

$$\dot{W} = -\beta \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{\partial V_{ij}}{\partial x_i} \text{sgn}(\sum_{j=1}^{N} \frac{\partial V_{ij}}{\partial x_i}) + \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{\partial V_{ij}}{\partial x_i} \phi_i \leq \sum_{i=1}^{N} \Bigg[\Bigg\| \sum_{j=1}^{N} \frac{\partial V_{ij}}{\partial x_i} \Bigg\|_1 (\|\phi_i\|_1 - \beta) \Bigg].$$

It is easy to see that if $\beta \geq \|\phi_i\|_1$, $\forall i \in \mathcal{I}$, then $\dot{W}$ is negative semi-definite. Therefore, having $W \geq 0$ and $\dot{W} \leq 0$, we can conclude that $V_{ij} \in \mathcal{L}_\infty$. Since V_{ij} is bounded, based on Definition 41, it is guaranteed that there will not be inter-agent collision and the connectivity is maintained.

In what follows, we focus on finding the relation between the optimal trajectory and the agents' positions. Based on Definition 41, we can obtain that

$$\frac{\partial V_{ij}}{\partial x_i} = \frac{\partial V_{ji}}{\partial x_i} = -\frac{\partial V_{ij}}{\partial x_j}. \tag{3.65}$$

Now, by summing both sides of the closed-loop system (3.1) with the control algorithm (3.63), for $i = 1, ..., N$ we have $\sum_{j=1}^{N} \dot{x}_j = \sum_{j=1}^{N} \phi_j$. We also know that agents have identical Hessians since it is assumed that $\nabla f_i(x_i, t) = \sigma x_i + g_i(t)$. Now, similar to the proof of Theorem 27, regardless of whether consensus is reached or not, we can show that $\sum_{j=1}^{N} \nabla f_j(x_j, t)$ will asymptomatically converge to zero. Hence we have $\sum_{j=1}^{N} x_i \to \frac{-\sum_{j=1}^{N} g_j}{\sigma}$. On the other hand, using Lemma 11 and under Assumption 8, we know $\sum_{j=1}^{N} \nabla f_j(x^*, t) = 0$. Hence, the optimal trajectory is $x^* = \frac{-\sum_{j=1}^{N} g_j}{N\sigma}$. This implies that $\frac{1}{N} \sum_{j=1}^{N} x_i \to x^*$, where we have shown that the center of the agents will track the team cost function minimizer. ■

Remark 43 *In Appendix .2, it is shown that a constant β can be selected such that $\beta \geq \|\phi_i\|_1, \forall t$ and $\forall i \in \mathcal{I}$, if $\|g_i(t)\|$ and $\|\dot{g}_i(t)\|, \forall t$ and $\forall i \in \mathcal{I}$, are bounded. In particular, it is shown that such a constant β can be determined, at time $t = 0$, using the agents' initial states and the upper bounds on $\|g_i(t)\|$ and $\|\dot{g}_i(t)\|, \forall t$ and $\forall i \in \mathcal{I}$.*

3.3.2 Distributed Convex Optimization with Swarm tracking behavior for Double-Integrator Dynamics

In this subsection, we focus on distributed time-varying optimization problem with swarm tracking behavior for double-integrator dynamics (3.28). We will propose an algorithm, where each agent has access to only its own position and the relative positions and velocities between itself and its neighbors. We propose the algorithm

$$u_i(t) = -\sum_{j\in N_i(t)} \frac{\partial V_{ij}}{\partial x_i} - \beta \sum_{j\in N_i(t)} \mathrm{sgn}(v_i - v_j) + \phi_i, \tag{3.66}$$

where V_{ij} is defined in Definition 41, β is a positive coefficient, and ϕ_i is defined in (3.29).

Theorem 44 *Suppose that the graph $\mathcal{G}(0)$ is connected, Assumptions 8, and 10 hold and the gradient of the cost functions can be written as $\nabla f_i(x_i, t) = \sigma x_i + g_i(t),\ \forall i \in \mathcal{I}$. If $\beta > \frac{\|(\Pi\otimes I_m)\Phi\|_2}{\sqrt{\lambda_2[L(t)]}}$ holds, for the system (3.28) with the algorithm* (3.66)*, the center of the agents tracks the optimal trajectory, the agents' velocities track the optimal velocity, and the agents maintain connectivity and avoid inter-agent collision.*

Proof: Writing the closed-loop system (3.28) with the control algorithm (3.66), based on the consensus error e_X and e_V defined in (3.31), we have

$$\begin{cases} \dot{e}_X = & e_V \\ \dot{e}_V = & -\alpha(L(t)\otimes I_m)e_V - \beta(D(t)\otimes I_m)\mathrm{sgn}\big[(D^T(t)\otimes I_m)e_V\big] \\ & \begin{pmatrix} \sum_{j\in N_1} \frac{\partial V_{1j}}{\partial e_{X_1}} \\ \vdots \\ \sum_{j\in N_N} \frac{\partial V_{Nj}}{\partial e_{X_N}} \end{pmatrix} + (\Pi\otimes I_m)\Phi. \end{cases} \tag{3.67}$$

Define the positive semi-definite Lyapunov function candidate

$$W = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} V_{ij} + \frac{1}{2}e_V^T e_V.$$

The time derivative of W along with the closed-loop system (3.67) can obtain as

$$\dot{W} = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\big(\frac{\partial V_{ij}}{\partial e_{X_i}}e_{V_i} + \frac{\partial V_{ij}}{\partial e_{X_j}}e_{V_j}\big) + e_V^T\dot{e}_V.$$

Using Lemma 3.1 in [8], $\dot{W}$ can be rewritten as

$$\dot{W} = -\,\alpha e_V^T(L(t)\otimes I_m)e_V - \beta e_V^T(D(t)\otimes I_m)\text{sgn}\big[(D^T(t)\otimes I_m)e_V\big] + e_V^T(\Pi\otimes I_m)\Phi.$$

Using a similar argument to that in (3.13), we obtain that if $\beta\sqrt{\lambda_2[L(t)]} > \|(\Pi\otimes I_m)\Phi\|_2$, then $\dot{W}$ is negative semi-definite. Therefore, having $W \geq 0$ and $\dot{W} \leq 0$, we can conclude that $V_{ij}, e_v \in \mathcal{L}_\infty$. By integrating both sides of $\dot{W} \leq -\alpha e_V^T(L(t)\otimes I_m)e_V$, we can see that $e_v \in \mathcal{L}_2$. Now, applying Barbalat's Lemma, we obtain that e_V asymptotically converges to zero which means that the agents' velocities reach consensus as $t \to \infty$. Since V_{ij} is bounded, it is guaranteed that there will not be inter-agent collision and the connectivity is maintained.

In the next step, using (3.65), by summing both sides of the closed-loop system (3.28) with the control algorithm (3.66), for $i = 1, ..., N$ we have $\sum_{j=1}^{N}\dot{v}_j = \sum_{j=1}^{N}\phi_j$. Now, if the team cost function $f(x,t)$ is convex and $\nabla f_i(x_i,t) = \sigma x_i + g_i(t)$, applying a procedure similar to the proof of Theorem 36, we can show that $\sum_{j=1}^{N}\nabla f_j(x_j,t)$ will asymptomatically converge to zero and (3.60) holds. Particularly, we have shown that the average of agents' states tracks the optimal trajectory. We also have shown that the agents' velocities reach consensus as $t \to \infty$. Thus we have $v_i(t)$ approaches $v^*(t)$ as $t \to \infty$. This completes the proof. ■

Remark 45 *The assumption $\beta > \frac{\|(\Pi \otimes I_m)\Phi\|_2}{\sqrt{\lambda_2[L(t)]}}$ can be interpreted as a bound on the difference between the agents' internal signals. Using the fact that $\lambda_2[L(t)]$ is lower bounded above 0, there always exists a β which satisfies $\beta > \frac{\|(\Pi \otimes I_m)\Phi\|_2}{\sqrt{\lambda_2[L(t)]}}$, if Assumption 26 holds. Here, to satisfy Assumption 26, with $\nabla f_i(x_i, t) = \sigma x_i + g_i(t),\ \forall i \in \mathcal{I}$, the boundlessness of $\|g_i(t) - g_j(t)\|, \|\dot{g}_i(t) - \dot{g}_j(t)\|$, and $\|\ddot{g}_i(t) - \ddot{g}_j(t)\|, \forall t$ and $\forall i, j \in \mathcal{I}$ is sufficient.*

Remark 46 *All the proposed algorithms are also applicable to non-convex functions. However, in this case it is just guaranteed that the agents converge to a local optimal trajectory of the team cost function.*

3.4 Simulation and Discussion

In this section, we present various simulations to illustrate the theoretical results in previous sections. Consider a team of six agents. The interaction among the agents is described by the undirected connected graph shown in Fig. 4.1. The agents' goal is to minimize the team cost function $\sum_{i=1}^{6} f_i(x_i(t), t)$, where $x_i(t) = (r_{x_i}(t), r_{y_i}(t))^T$ is the coordinate of agent i in $2D$ plane.

In our first example, we apply the algorithm (3.9) for single-integrator dynamics (3.1). The local cost function for agent i is chosen as

$$f_i(x_i(t), t) = (r_{x_i}(t) - i\sin(t))^2 + (r_{y_i}(t) - i\cos(t))^2, \tag{3.68}$$

where it is easy to see that the optimal point of the team cost function creates a trajectory of a circle whose center is at the origin and radius is equal to $\frac{21}{6}$. For local cost functions (3.68), Assumption 10 and the conditions for agents' cost functions in Remark 20 hold. In addition they have identical Hessians and the team cost function is convex. $\beta_{ij}(0) = \beta_{ji}(0)$ are chosen randomly within (0.1,2)

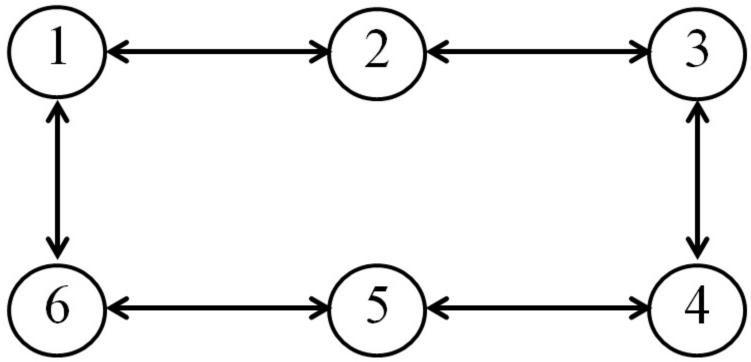

Figure 3.1: Undirected graph

in the algorithm (3.9). The trajectory of the agents and the optimal point is shown in Fig. 3.2. It can be seen that agents reach consensus and the agents track the optimal trajectory which minimizes the team cost function.

In case of double-integrator dynamics, we first give an example to illustrate the algorithm (3.29) for (3.28) with the local cost functions defined by (3.68). Choosing the coefficients in (3.29) as $\mu = 5, \alpha = 12, \gamma = 5, \zeta = 12$, and $\beta_{ij}(0) = \beta_{ji}(0)$ randomly within (0.1,2), the team cost function will be minimized as it can be seen in Fig. 3.3 while the agents reach consensus.

In our next example, we illustrate the results obtained in Subsection 3.2.3, where it has been clarified that the algorithm (3.37) -(3.40) can be used for local cost functions with nonidentical Hessians. Here, the local cost function

$$f_i(x_i(t), t) = (\frac{r_{x_i}(t)}{i} - \sin(t))^2 + (\frac{r_{y_i}(t)}{i} - \cos(t))^2, \tag{3.69}$$

will be used, where $H_i(x_i, t) = \frac{2}{i^2} I_2,\ \forall i \in \mathcal{I}$. It can be obtained that the team cost function's optimal trajectory creates a circle whose center is at the origin and radius is equal to 1.64. Algorithm

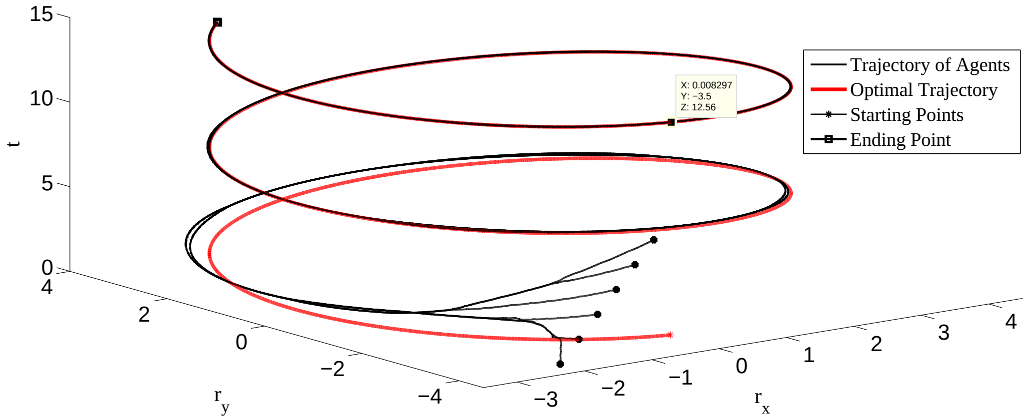

Figure 3.2: Trajectories of all agents along with optimal trajectory using the algorithm (3.9) for local cost function (3.68)

(3.37) -(3.40) with $\kappa = 12, \rho = 2, \alpha_1 = 0.1$, and $\alpha_2 = \frac{0.2}{1.1}$ is used for the system (3.28). Fig. 3.4 shows that the team cost function is minimized.

In our next example, the results in Subsection 3.2.5 is illustrated, where the invariant approximation of the signum function is employed. Here, the algorithm (3.47) and (3.58) are used to minimize the agents' team cost function for local cost functions defined as (3.68). The coefficients are chosen as $\mu = 5, \alpha = 10, \gamma = 5, \zeta = 5, \epsilon = 2$ and $\beta_{ij}(0) = \beta_{ji}(0)$ randomly within (0.1,2). Fig. 3.5 shows the agents' trajectories along with the optimal one. It is shown that the agents track the optimal trajectory with a bounded error.

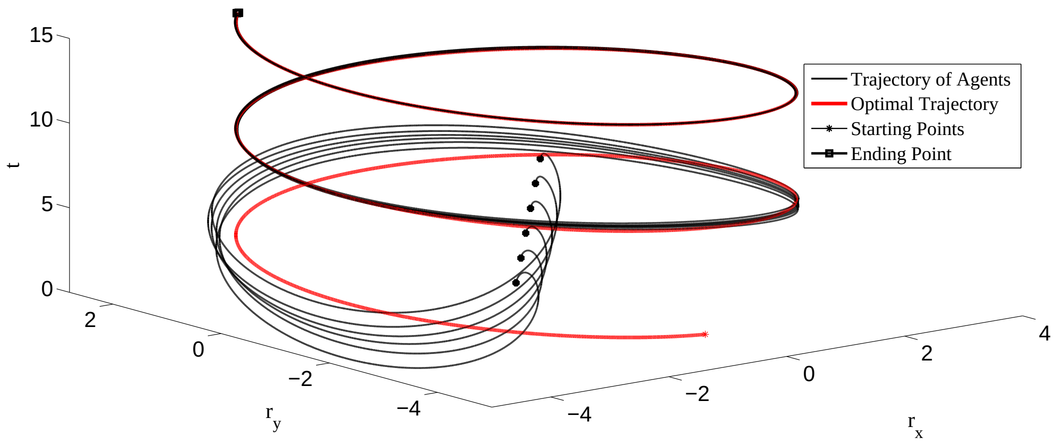

Figure 3.3: Trajectories of all agents along with optimal trajectory using the algorithm (3.29) for local cost function (3.68)

In our last illustration, the swarm tracking control algorithm (3.66) is employed, where the local cost functions are defined as

$$f_i(x_i(t), t) = (r_{x_i}(t) + 2i\frac{\sin(0.5t)}{t+1})^2 + (r_{y_i}(t) + i\sin(0.1t))^2, \tag{3.70}$$

In this case, we let $R = 5$. The parameter of the control algorithm (3.66) is chosen as $\beta = 20$. To guarantee the collision avoidance and connectivity maintenance, the potential function partial derivative is chosen as Eqs. (36) and (37) in [8], where $d_{ij} = 0.5$, $\forall i, j$. Fig. 3.6 shows that the center of the agents' positions tracks the optimal trajectory while the agents remain connected and avoid collisions.

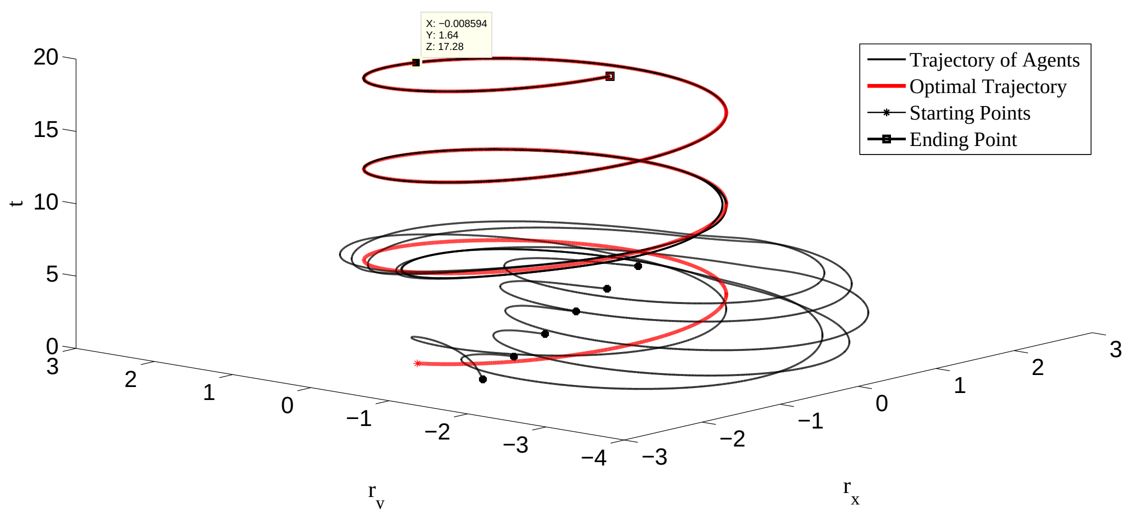

Figure 3.4: Trajectories of all agents along with optimal trajectory using the algorithm (3.37) -(3.40) for local cost function (3.69)

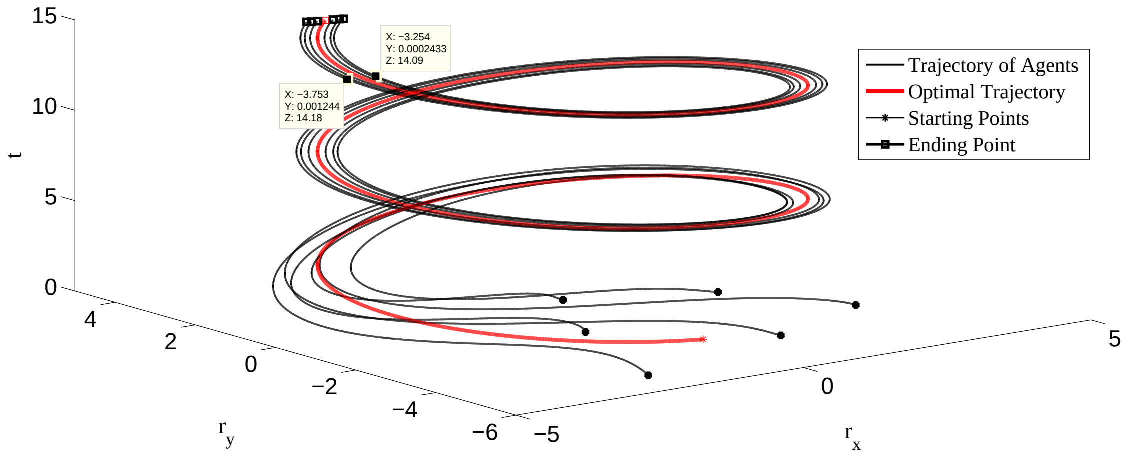

Figure 3.5: Trajectories of all agents using the algorithm (3.47) and (3.58) for local cost function (3.68)

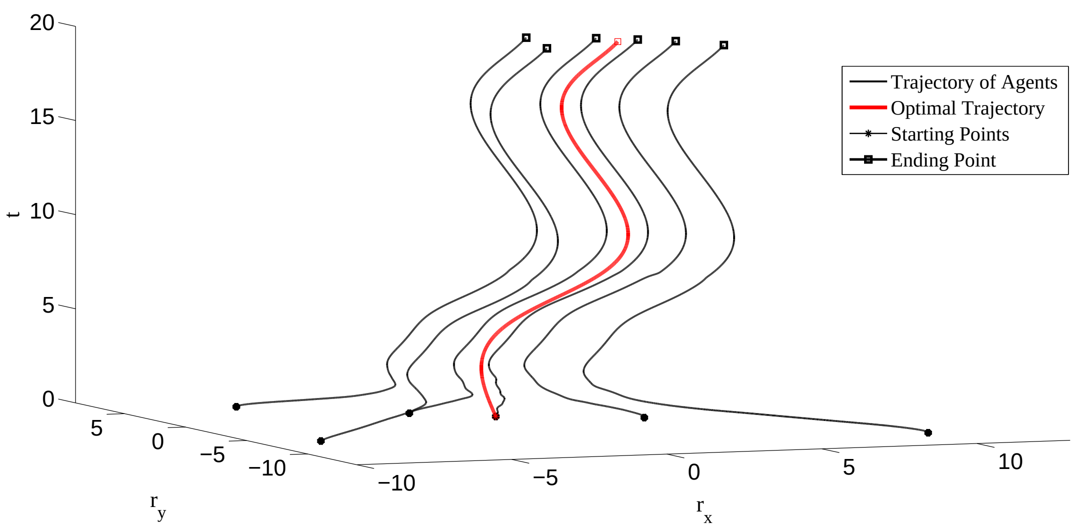

Figure 3.6: Trajectories of all agents using the algorithm (3.66) for local cost function (3.70)

Chapter 4

Heterogeneous Distributed Average Tracking and Its Application in Distributed Optimization

4.1 Distributed Average Tracking

Consider a heterogeneous multi-agent system consisting of N physical agents, where $\mathcal{I}$ denotes the index set $\{1, \cdots, N\}$. The agents' are described by single-integrator, double-integrator and Euler-Lagrange dynamics. Without loss of generality, we label single-integrator agents as $1, \ldots, M-1$, where their dynamics is described by

$$\dot{x}_i = u_i, \qquad i = 1, \ldots, M-1. \tag{4.1}$$

We also label double-integrator agents as $M, \ldots, N'-1$, with dynamics described by

$$\dot{x}_i = v_i, \quad \dot{v}_i = u_i, \qquad i = M, \ldots, N'-1. \tag{4.2}$$

Agents with Euler-Lagrange dynamics are labeled as $N', \ldots, N$, and their dynamic is described by

$$M_i(x_i)\ddot{x}_i + C_i(x_i, \dot{x}_i)\dot{x}_i + g_i(x_i) = u_i \qquad i = N', \ldots, N, \tag{4.3}$$

where $x_i(t) \in \mathbb{R}^{\mathfrak{p}}$, $v_i(t) \in \mathbb{R}^{\mathfrak{p}}$ and $u_i(t) \in \mathbb{R}^{\mathfrak{p}}$ are, respectively, ith agent's position, velocity and control input. $M_i(x_i)$ is the $\mathfrak{p} \times \mathfrak{p}$ symmetric inertia matrix, $C_i(x_i, \dot{x}_i)\dot{x}_i$ is the Coriolis and centrifugal force, and $g_i(x_i)$ is the vector of gravitational force. The dynamics of the Lagrange systems satisfy the following properties [74]:

(P1) There exist positive constants $k_{\underline{M}}, k_{\overline{M}}, k_{\overline{C}}, k_{\overline{g}}$ such that $k_{\underline{M}} I_p \leq M_i(x_i) \leq k_{\overline{M}} I_{\mathfrak{p}}$, $||C_i(x_i, \dot{x}_i)\dot{x}_i|| \leq k_{\overline{C}}||\dot{x}_i||$ and $||g_i(x_i)|| \leq k_{\overline{g}}$.

(P2) $\dot{M}_i(x_i) - 2C_i(x_i, \dot{x}_i)$ is skew symmetric.

(P3) The Lagrange dynamics can be rewritten as, i.e., $M_i(x_i)\chi + C(x_i, \dot{x}_i)\psi + g_i(x_i) = Y_i(x_i, \dot{x}_i, \chi, \psi)\theta_i$, $\forall \chi, \psi \in \mathbb{R}^{\mathfrak{p}}$, where $Y_i \in \mathbb{R}^{\mathfrak{p} \times \mathfrak{p}_\theta}$ is the regression matrix and $\theta_i \in \mathbb{R}^{\mathfrak{p}_\theta}$ is the unknown but constant parameter vector.

In our framework the agents' interaction topology is described by an undirected graph G.

Assumption 47 *Graph G is connected.*

Suppose that each agent has a time-varying reference input $r_i(t) \in \mathbb{R}^{\mathfrak{p}}$, $i \in \mathcal{I}$, satisfying

$$\begin{aligned}\dot{r}_i(t) =& v_i^r(t), \\ \dot{v}_i^r(t) =& a_i^r(t),\end{aligned} \tag{4.4}$$

where $v_i^r(t) \in \mathbb{R}^{\mathfrak{p}}$ and $a_i^r(t) \in \mathbb{R}^{\mathfrak{p}}$ are, respectively, the reference velocity and the reference acceleration for agent i at time t.

Assumption 48 *The reference input $r_i(t), \forall i \in \mathcal{I}$ and its velocity $v_i^r(t)$ are bounded. It is assumed that $\|r_i(t)\| < \bar{r}$, and $\|v_i^r(t)\| < \bar{v^r}, \forall i \in \mathcal{I}$, where $\bar{r}$ and $\bar{v^r}$ are positive constants.*

Here the goal is to design $u_i(t)$ for agent $i \in \mathcal{I}$, to track the average of the reference inputs, i.e.,

$$\lim_{t\to\infty} ||x_i(t) - \frac{1}{N}\sum_{j=1}^{N} r_j(t)|| = 0, \tag{4.5}$$

where each agent has only local interaction with its neighbors.

Corollary 49 *[26] Consider the system,*

$$\dot{x} = f(x,t), \tag{4.6}$$

where $x(t) \in \mathcal{D} \subset \mathbb{R}^n$ and $f : \mathcal{D} \times [0,\infty] \to \mathbb{R}^n$ and $\mathcal{D}$ is an open and connected set containing $x = 0$, and suppose is Lebesgue measurable and is essentially locally bounded, uniformly in t. Let $V : \mathcal{D} \times [0,\infty] \to \mathbb{R}$ be locally Lipschitz and regular such that

$$\begin{aligned} W_1(x) \leq & V(x,t) \leq W_2(x) \\ & \dot{\tilde{V}} \leq -W(x), \end{aligned} \tag{4.7}$$

$\forall t \geq 0, \forall x \in \mathcal{D}$, where W_1 and W_2 are continuous positive definite functions, and W is a continuous positive semi-definite function on $x \in \mathcal{D}$ and $\dot{\tilde{V}}$ is the generalized gradient of function V. Choose $r > 0$ and $c > 0$ such that $\mathcal{B}_r \subset \mathcal{D}$ and $c < \min_{\|x\|=r} W_1(x)$. Then, all Filippov solutions of (4.6) *such that $x(t_0) \in \{x \in \mathcal{B}_r | W_2(x) \leq c\}$ are bounded and satisfy $W(x) \to 0$ as $t \to \infty$.*

4.1.1 Distributed Average Tracking for Heteregeous Physical Agents Using Neighbors' Positions

In this subsection, we study the distributed average tracking problem for heterogeneous multi-agent system consisting of three different dynamics, single-integrator, double-integrator and Euler-Lagrange dynamics. Here, we propose an algorithm to achieve goal (4.5), where each agent is required to have access to only its own position and the relative positions between itself and its neighbors. Note that in some applications, these pieces of information can be obtained by sensing; hence the communication necessity might be eliminated. For notational simplicity, we will remove the index t from variables in the reminder of the paper.

Three controllers are proposed, where each agent according to its dynamic will employ the proper control u_i. Consider the control input

$$u_i = -\beta_i \mathrm{sgn}\big[\sum_{j=1}^{N} a_{ij}(x_i - x_j)\big] - (x_i - r_i) + v_i^r, \qquad i = 1, \dots, M-1, \tag{4.8}$$

for agents with single-integrator dynamics and

$$\begin{aligned} u_i = &- \beta_i \mathrm{sgn}\big[\sum_{j=1}^{N} a_{ij}(x_i - x_j)\big] - \sum_{j=1}^{N} a_{ij}(x_i - x_j) \\ &- (x_i - r_i) - 2(v_i - v_i^r) + a_i^r, \qquad i = M, \dots, N'-1 \end{aligned} \tag{4.9}$$

for agents with double-integrator dynamics and

$$
\begin{aligned}
u_i =& Y_i(x_i, \dot{x}_i, \upsilon_i, \nu_i)\hat{\theta}_i - \alpha s_i - \sum_{j=1}^{N} a_{ij}(x_i - x_j), \qquad i = N', \ldots, N \\
\nu_i =& - \beta_i \text{sgn}\big[\sum_{j=1}^{N} a_{ij}(x_i - x_j)\big] - (x_i - r_i) + v_i^r, \\
s_i =& \dot{x}_i - \nu_i, \\
\dot{\hat{\theta}}_i =& - Y_i(x_i, \dot{x}_i, \upsilon_i, \nu_i)^T s_i,
\end{aligned} \tag{4.10}
$$

for agents with Euler-Lagrange dynamics, where α and β_i are positive constant gains to be designed, and $\hat{\theta}_i$ is the estimate of the unknown but constant parameters θ_i. Using the definition of the generalized gradient [18], the generalized time-derivative of s_i and ν_i are defined, respectively, as ϑ_i and υ_i, where $\zeta_i \in \vartheta_i$, and $\mu_i \in \upsilon_i$. Let ξ_i denotes the minimum norm element of υ_i.

Theorem 50 *Under the control law given by* (4.8)-(4.10) *for system defined in* (4.1)-(4.3)*, distributed average tracking goal* (4.5) *is achieved asymptotically, provided that Assumptions 47 and 48 hold and the control gain* β_i *is chosen such that* $\min_{i \in \mathcal{I}} \beta_i > \bar{r} + \bar{v^r}$ *and* $\alpha > 0$.

Proof: Rewrite the Laplacian matrix as $L = [L_s^T \ L_d^T \ L_e^T]^T$, where $L_s \in \mathbb{R}^{(M-1)\times N}, L_d \in \mathbb{R}^{(N'-M)\times N}$ and $L_e \in \mathbb{R}^{(N-N'+1)\times N}$ and subscripts $_s$, $_d$ and $_e$, respectively, are used for single-integrator, double-integrator and Euler-Lagrange dynamics, i.e, L_s describes the interaction among single-integrator agents and other agents. Let x denotes the column stack vectors of all x_i's $i = 1, \ldots, N$, and it can be rewritten as $x = [x_s^T \ x_d^T \ x_e^T]^T$, where x_s, x_d and x_e are, respectively, the column stack vectors of the positions for single-integrator, double-integrator and Euler-Lagrange dynamics.

System (4.1) with control input (4.8) can be rewritten in vector form as

$$\dot{x}_s = -\beta_s \text{sgn}[(L_s \otimes I_{\mathfrak{p}})x] - (x_s - r_s) + v_s^r, \tag{4.11}$$

where $r_s = [r_1^T, ..., r_{M-1}^T]^T$, and $v_s^r = [v_1^{rT}, ..., v_{M-1}^{r}{}^T]^T$, denote, respectively, the aggregated reference inputs and reference velocities of the single-integrator dynamic (4.1) and $\beta_s = \text{diag}(\beta_1, \cdots, \beta_{M-1})$. System (4.2) with control input (4.9) can be rewritten in vector form as

$$\begin{aligned}\dot{x}_d =& v_d \\ \dot{v}_d =& -\beta_d \text{sgn}[(L_d \otimes I_{\mathfrak{p}})x] - (L_d \otimes I_{\mathfrak{p}})x - (x_d - r_d) - 2(v_d - v_d^r) + a_d^r,\end{aligned} \tag{4.12}$$

where $v_d = [v_M{}^T, ..., v_{N'-1}{}^T]^T$, $r_d = [r_M^T, ..., r_{N'-1}^T]^T$, $v_d^r = [v_M^r{}^T, ..., v_{N'-1}^r{}^T]^T$, and $a_d^r = [a_M^r{}^T, ..., a_{N'-1}^r{}^T]^T$, denote, respectively, the aggregated velocities, reference inputs, reference velocities and reference accelerations of the double-integrator system (4.2) and $\beta_d = \text{diag}(\beta_M, \cdots, \beta_{N'-1})$.

It follows form (P3) that $M(x_e)\zeta + C(x_e, \dot{x}_e)s + Y(x_e, \dot{x}_e, v, \nu)\theta = u_e$, where $M(x_e) \triangleq \text{diag}\{M_{N'}(x_{N'}), \cdots, M_N(x_N)\}$, $C(x_e, \dot{x}_e) \triangleq \text{diag}\{C_{N'}(x_{N'}, \dot{x}_{N'}), \cdots, C_N(x_N, \dot{x}_N)\}$, and $u_e = [u_{N'}^T, ..., u_N^T]^T$. Now, by replacing the control input (4.10), we have

$$M(x_e)\zeta + C(x_e, \dot{x}_e)s + Y(x_e, \dot{x}_e, v, \nu)\theta = Y(x_e, \dot{x}_e, v, \nu)\hat{\theta} - \alpha s - (L_e \otimes I_{\mathfrak{p}})x,$$

where ζ, s, ν, θ and $\hat{\theta}$ are, respectively, the column stack vectors of all ζ_i's, s_i's, ν_i's, θ_i's and $\hat{\theta}_i$'s, $i = N', \ldots, N$. Let $\beta_e = \text{diag}(\beta_{N'}, \cdots, \beta_N)$. Let $r = [r_s^T \ r_d^T \ r_e^T]^T$, and $v^r = [v^r{}_s^T v^r{}_d^T \ v^r{}_e^T]^T$ denote, respectively, the aggregated reference inputs, and reference velocities for all agents.

Define the Lyapunov function V_t as

$$V_t = \frac{1}{2}x^T(L \otimes I_{\mathfrak{p}})x + \frac{1}{2}(v_d - \Phi)^T(v_d - \Phi) + \frac{1}{2}sMs + \frac{1}{2}\tilde{\theta}^T\tilde{\theta}, \tag{4.13}$$

where $\Phi(x) = -\beta_d \text{sgn}[(L_d \otimes I_{\mathfrak{p}})x] - (x_d - r_d) + v_d^r$, and $\tilde{\theta} = \hat{\theta} - \theta$. It is easy to see that we have $V_1 = \frac{1}{2}x^T L x = \frac{1}{2}e^T e$, where $e = D^T x$ and D is defined in Section **??**. Hence V_1 is a positive definite function corresponding to e. The candidate Lyapunov function V_t satisfies the following inequalities:

$$W_1(y) \leq V(y,t) \leq W_2(y), \tag{4.14}$$

where $y = \begin{bmatrix} x \\ v_d - \Phi \\ s \\ \tilde{\theta} \end{bmatrix}$ and W_1 and W_2 are positive-definite continuous functions defined as $W_1 = \Lambda_1 \|y\|^2$ and $W_2 = \Lambda_2 \|y\|^2$, where Λ_1 and Λ_2 are positive constants.

Define the generalized gradient of V_t and Φ by $\dot{\tilde{V}}_t$ and $\dot{\tilde{\Phi}}$, respectively. Every element of $\eta \in \dot{\tilde{V}}$ satisfies

$$\eta \leq -x^T(L \otimes I_{\mathfrak{p}})\bar{\beta}\begin{bmatrix} \text{sgn}[(L_s \otimes I_{\mathfrak{p}})x] \\ \text{sgn}[(L_d \otimes I_{\mathfrak{p}})x] \\ \text{sgn}[(L_e \otimes I_{\mathfrak{p}})x] \end{bmatrix} + x^T(L \otimes I_{\mathfrak{p}})\Bigg(\begin{bmatrix} 0 \\ v_d - \Phi \\ s \end{bmatrix} + \begin{bmatrix} -x_s + r_s + v_s^r \\ -x_d + r_d + v_d^r \\ -x_e + r_e + v_e^r \end{bmatrix}\Bigg)$$

$$+ (v_d - \Phi)^T \times \Big(- \beta_d \text{sgn}[(L_d \otimes I_{\mathfrak{p}})x] - (L_d \otimes I_{\mathfrak{p}})x - (x_d - r_d) - 2(v_d - v_d^r) + a_d^r - \varrho\Big)$$
$$+ \frac{1}{2}s\dot{M}s + \tilde{\theta}^T\dot{\hat{\theta}} + s^T\big(- C(x_e, \dot{x}_e)s - Y(x_e, \dot{x}_e, v, \nu)\tilde{\theta} - \alpha s\big) - s^T(L_e \otimes I_{\mathfrak{p}})x,$$

where $\bar{\beta} = \text{diag}(\beta_s, \beta_d, \beta_e)$, and $\varrho \in \dot{\tilde{\Phi}}$ and we used the fact that we can rewrite equations (4.12) and (4.10), respectively, as

$$x_d = \Phi + v_d - \Phi$$

and

$$\dot{x}_e = -\beta_e \text{sgn}[(L_e \otimes I_{\mathfrak{p}})x] - (x_e - r_e) + v_e^r + s. \tag{4.15}$$

Employing (4.10), we have

$$\begin{aligned}
\eta \leq & -(\min_{i \in \mathcal{I}} \beta_i) \left\|(L \otimes I_{\mathfrak{p}})x\right\|_1 - x^T(L \otimes I_{\mathfrak{p}})x + x^T(L \otimes I_{\mathfrak{p}})(r + v^r) + x^T(L_d \otimes I_{\mathfrak{p}})^T(v_d - \Phi) \\
& + x^T(L_e \otimes I_{\mathfrak{p}})^T s + (v_d - \Phi)^T[\Phi - v_d - (L_d \otimes I_{\mathfrak{p}})x - (v_d - v_d^r) + a_d^r - \varrho]X + \frac{1}{2} s\dot{M}s - \alpha s^T s \\
& + s^T[-C(x_e, \dot{x}_e)s - Y(x_e, \dot{x}_e, \upsilon, \nu)\tilde{\theta}] - \alpha s^T s - s^T(L_e \otimes I_{\mathfrak{p}})x - \tilde{\theta}^T Y(x_e, \dot{x}_e, \upsilon, \nu)^T s \\
= & -(\min_{i \in \mathcal{I}} \beta_i) \left\|(L \otimes I_{\mathfrak{p}})x\right\|_1 - x^T(L \otimes I_{\mathfrak{p}})x + x^T(L \otimes I_{\mathfrak{p}})(r + v^r) + (v_d - \Phi)^T(\Phi - v_d) \\
& + (v_d - \Phi)^T[(v_d^r - v_d) + a_d^r - \varrho - \chi + \chi] - \alpha s^T s,
\end{aligned}$$

where χ is the minimum norm element of $\dot{\tilde{\Phi}}$ and we have used property (P2) to obtain the last equality.

Under assumption 48 and by selecting $\min_{i \in \mathcal{I}} \beta_i > \bar{r} + \bar{v^r}$, we know that $-(\min_{i \in \mathcal{I}} \beta_i) \times \left\|(L \otimes I_{\mathfrak{p}})x\right\|_1 + x^T(L \otimes I_{\mathfrak{p}})(r + v^r) < 0$. Now, using the fact that $\chi = -(v_d - v_d^r) + a_d^r$, it follows $\eta \leq -x^T(L \otimes I_{\mathfrak{p}})x + (v_d - \Phi)^T(\Phi - v_d) + (v_d - \Phi)^T(\chi - \varrho) - \alpha s^T s$. Using an argument similar to [45], $\chi - \varrho$ is zero wherever $\nu(x, v, t)$ is differentibale. Also at points of non-differentiability, we will have $\chi - \varrho = 0$ [45]. Hence, $\eta \leq -x^T(L \otimes I_{\mathfrak{p}})x - (v_d - \Phi)^T(v_d - \Phi) - \alpha s^T s$. Now, we can see that $\dot{\tilde{V}}_t \leq -W(y)$, where W is a positive semi-definite defined on the domain $\mathcal{D} = \mathbb{R}^{N(3\mathfrak{p} + \mathfrak{p}_\theta)}$. As a result $V_t \in \mathcal{L}_\infty$, and $\tilde{\theta}, s, e, (v_d - \Phi) \in \mathcal{L}_\infty$.

By calling $z = x_e - r_e$, we can rewrite (4.15) as $\dot{z} = -z - \beta_e \text{sgn}[(L \otimes I_{\mathfrak{p}})x] + s$, where we know that $(L \otimes I_{\mathfrak{p}})x$ and s are bounded. Hence it is easy to see that z will remain bounded. Also $\dot{z}$ will be bounded because $z, (L \otimes I_{\mathfrak{p}})x$ and s are bounded. Now, using (4.13) and under assumptions (P1) and (P3), it is easy to see that ζ is bounded.

Knowing the fact that s is continuous and bounded, we can use the mean value theorem for nonsmooth functions [34], where we have

$$\frac{s(t_1) - s(t_0)}{t_1 - t_0} \in S, \ \forall t_0, t_1 \tag{4.16}$$

and S denotes the set $\partial s(t) \cup -\partial(-s)(t)$ for $t \in (t_0, t_1)$. Because ζ is bounded for every $\zeta \in \vartheta$, we know that there exists a κ such that $\partial s(t) \leq \kappa, \forall t$. Hence, the members of the set S are all bounded and we have $s(t_1) - s(t_0) \leq \kappa(t_1 - t_0), \forall t_0, t_1$, which shows s is lipschitz and therefore it is uniformly continuous.

Now, choose $\rho > 0$ such that $\mathcal{B}_\rho \subset \mathcal{D}$ denotes a closed ball. Define $\mathcal{M} \subset \mathcal{D}$ as $\mathcal{M} \triangleq \{\varpi \subset \mathcal{M} | W_2(\varpi) \leq \min\limits_{\|\varpi\|=\rho} W_1(\varpi) = \lambda_1 \rho^2\}$. Then, all conditions in Corollary 49, LaSalle-Yoshizawa for nonsmooth systems, are provided and we have $W(y) \to 0$ as $t \to \infty, \forall y(0) \in \mathcal{M}$. Because ρ can be selected arbitrarily large to include all initial conditions, the region of attraction is $\mathcal{M} = \mathbb{R}^{N(3\mathfrak{p}+\mathfrak{p}_\theta)}$.

Now, having $W(y) \to 0$, it follows that $s \to 0$, $v_d - \Phi \to 0$ and $(L \otimes I_\mathfrak{p})x \to 0$. Since $v_d - \Phi \to 0$, we will have

$$\dot{x}_d = -\beta_d \text{sgn}[(L_d \otimes I_\mathfrak{p})x] - (x_d - r_d) + v_d^r + \epsilon_d, \tag{4.17}$$

where $\epsilon_d \to 0$ as $t \to \infty$. Also using (4.15), and because $s \to 0$, we have

$$\dot{x}_e = -\beta_e \text{sgn}[(L_e \otimes I_\mathfrak{p})x] - (x_e - r_e) + v_e^r + \epsilon_e, \tag{4.18}$$

where $\epsilon_e \to 0$ as $t \to \infty$.

Hence, it turns out that using (4.11), (4.17) and (4.18), we have

$$\dot{x}_s = -\beta_s \mathrm{sgn}[(L_s \otimes I_{\mathfrak{p}})x] - (x_s - r_s) + v_s^r, \tag{4.19}$$

$$\dot{x}_d = -\beta_d \mathrm{sgn}[(L_d \otimes I_{\mathfrak{p}})x] - (x_d - r_d) + v_d^r + \epsilon_d, \tag{4.20}$$

$$\dot{x}_e = -\beta_e \mathrm{sgn}[(L_e \otimes I_{\mathfrak{p}})x] - (x_e - r_e) + v_e^r + \epsilon_e, \tag{4.21}$$

where we can rewrite it as

$$\dot{x} = -\bar{\beta} \mathrm{sgn}[(L \otimes I_{\mathfrak{p}})x] - (x - r) + v^r + \epsilon, \tag{4.22}$$

where $\epsilon = \begin{bmatrix} 0 \\ \epsilon_d \\ \epsilon_e \end{bmatrix}$. Define the Lyapunov candidate function $V_1 = x^T(L \otimes I_{\mathfrak{p}})x$, where its time-derivative along the system (4.22) is

$$\begin{aligned} \dot{V}_1 = &- \bar{\beta} x^T (L \otimes I_{\mathfrak{p}}) \mathrm{sgn}[(L \otimes I_{\mathfrak{p}})x - x^T(L \otimes I_{\mathfrak{p}})x + x^T(L \otimes I_{\mathfrak{p}})(r + v^r) + x^T(L \otimes I_{\mathfrak{p}})\epsilon \\ &\leq -\bar{\beta} \left\| (L \otimes I_{\mathfrak{p}})x \right\|_1 - x^T(L \otimes I_{\mathfrak{p}})x + x^T(L \otimes I_{\mathfrak{p}})(r + v^r) + x^T(L \otimes I_{\mathfrak{p}})\epsilon \end{aligned} \tag{4.23}$$

Now, by by selecting $\min_{i \in \mathcal{I}} \beta_i > \bar{r} + \bar{v^r}$ and knowing that $\epsilon \to 0$ as $t \to \infty$, we can employ Lemma 2.19 in [59]. As a result we can show that the agents' positions reach consensus, i.e, $x_i = x_j$, as $t \to \infty$. Define the variable $S_1 = (\mathbf{1}_N^T \otimes I_{\mathfrak{p}})(x - r) = \sum_{i=1}^N x_i - \sum_{i=1}^N r_i$, then we can rewrite (4.22) as

$$\dot{S}_1 = -(\mathbf{1}_N^T \otimes I_{\mathfrak{p}}) \bar{\beta} \mathrm{sgn}(Lx) - S_1 + \epsilon. \tag{4.24}$$

Then we can use input-to-state stability to analyze the system (4.24) by treating the term $(\mathbf{1}_N^T \otimes I_{\mathfrak{p}})\bar{\beta}\mathrm{sgn}(Lx)$ as the input and S_1 as the state. The system (4.24) with zero input is exponentially stable and hence input-to-state stable. Since $Lx \to 0$ as $t \to \infty$ for each agent, it follows that

$S_1 \to 0$ as $t \to \infty$. This implies that $\sum_{i=1}^{N} x_i \to \sum_{i=1}^{N} r_i$, where combining it with the consensus result, we will have

$$x_i \to \frac{1}{N}\sum_{j=1}^{N} r_j, \ \forall i \in \mathcal{I}. \tag{4.25}$$

■

Remark 51 *Note that the controllers in* (4.8)-(4.10) *are proposed precisely for our heterogeneous framework and they are not just a simple combination of the controllers in the literature. The interaction among agents with different dynamics is one of the challenge that we have faced. The only common state among our agents is position; hence we cannot use the well-known algorithms for double-integrator or Euler-Lagrange dynamics, which they require velocity measurement or communication. It is worthwhile to mention that algorithm* (4.9) *is proposed based on the intuition behind Backstepping approach.*

4.1.2 Estiamtor Based Distributed Average Tracking for Heteregeous Physical Agents

In this subsection, we propose an estimator based algorithm to address the distributed average tracking problem (4.5) for heterogeneous multi-agent systems (4.1)-(4.3). Here, a filter is used to generate the average of the inputs in a distributed manner, where each agent tracks its own generated signal. In some frameworks, the estimator based algorithm is able to relax the restrictive assumptions mentioned in Subsection 4.1.1. As a trade-off the estimator based algorithm necessitates communication between neighbors.

First, a filter is introduced for each agent to estimate the average of the reference inputs and reference velocities. Then the control input u_i, $i = 1, \ldots, N$, is designed for each agent such that x_i tracks p_i, where $p_i \in \mathbb{R}^{\mathfrak{p}}$ is the filter's output. The filter, adapted from [29], is proposed as following

$$\begin{aligned}
\dot{p}_i =& q_i \\
\dot{q}_i =& -\beta_i \text{sgn}\big[\sum_{j=1}^{N} a_{ij}\{(p_i+q_i)-(p_j+q_j)\}\big] \\
& -\kappa(p_i - r_i) - \kappa(q_i - v_i^r) + a_i^r, \qquad i = 1, \dots, N
\end{aligned} \tag{4.26}$$

where $\beta_i = \eta_i \|r_i\|_1 + \eta_i \|v_i^r\|_1 + \|a_i^r\|_1 + \gamma$ is a state based gain and η_i, γ and κ are positive constants to be designed. The controllers are given by

$$u_i = -\eta_i \text{sgn}(x_i - p_i) + q_i \qquad i = 1, \dots, M-1, \tag{4.27}$$

for agents with single-integrator dynamics and

$$\begin{aligned}
u_i =& -\eta_i \text{sgn}[(x_i - p_i) + (v_i - q_i)] - \eta_i(x_i - p_i) \\
& -\eta_i(v_i - q_i) + \dot{q}_i, \qquad i = M, \dots, N'-1,
\end{aligned} \tag{4.28}$$

for agents with double-integrator dynamics and

$$\begin{aligned}
u_i =& Y_i(x_i, \dot{x}_i, p_i, q_i, \dot{q}_i)\hat{\theta}_i - \alpha s_i \qquad i = N', \dots, N \\
s_i =& \mu(x_i - p_i) + (\dot{x}_i - q_i), \\
\dot{\hat{\theta}}_i =& -Y_i(x_i, \dot{x}_i, p_i, q_i, \dot{q}_i)^T s_i,
\end{aligned} \tag{4.29}$$

for agents with Euler-Lagrange dynamics, where α and μ are positive constants.

Theorem 52 *Under the control law given by* (4.26)-(4.29) *for system defined in* (4.1)-(4.3)*, the distributed average tracking goal* (4.5) *is achieved asymptotically, provided that Assumptions 47 and 48 hold and the control gains are chosen such that $\eta_i > \kappa > 1$ and γ, α and μ are positive constants.*

Proof: **Filter**: Here, it is proved that, $\forall i = 1, \cdots, N$, we have

$$\lim_{t\to\infty} p_i = \frac{1}{N}\sum_{j=1}^{N} r_j$$
$$\lim_{t\to\infty} q_i = \frac{1}{N}\sum_{j=1}^{N} v_j^r. \tag{4.30}$$

Let $p = [p_1^T, \ldots, p_N^T]^T$, and $q = [q_1^T, \ldots, q_N^T]^T$, denote the aggregated states of the filters. Let $M \triangleq I_N - \frac{1}{N}\mathbf{1}_N^T\mathbf{1}_N$. Note that M has one simple zero eigenvalue with $\mathbf{1}_N$ as its right eigenvector and has 1 as its other eigenvalue with the multiplicity $N-1$. Define the consensus error vectors $\tilde{p} = (M \otimes I_\mathfrak{p})p$ and $\tilde{q} = (M \otimes I_\mathfrak{p})q$. Then it is easy to see that $\tilde{p} = 0$ (respectively, $\tilde{q} = 0$) if and only if $p_i = p_j,\ \forall i, j \in \mathcal{I}$ ($q_i = q_j, \forall i, j \in \mathcal{I}$).

Now, the estimator dynamics (4.26) can be rewritten in vector form as

$$\begin{aligned}
\dot{\tilde{p}} =& \tilde{q},\\
\dot{\tilde{q}} =& -\alpha(M\beta \otimes I_\mathfrak{p})\mathrm{sgn}[(L \otimes I_\mathfrak{p})(\tilde{p} + \tilde{q})] - \kappa\tilde{p}\\
&+ \kappa(M \otimes I_\mathfrak{p})r - \kappa\tilde{q} + \kappa(M \otimes I_\mathfrak{p})v^r + (M \otimes I_\mathfrak{p})a^r,
\end{aligned}$$

where $r(t) = [r_1^T, \ldots, r_N^T]^T$, $v^r(t) = [v_1^{rT}, \ldots, v_N^{r\,T}]^T$ and $a^r(t) = [a^r{}_1^T, \ldots, a^r{}_N^T]^T$, are respectively, the aggregated reference inputs, reference velocities and reference accelerations, and $\beta = \mathrm{diag}(\beta_1, \ldots, \beta_N)$.

Consider the Lyapunov function candidate $V_1 = \frac{1}{2}\begin{bmatrix}\tilde{p}^T & \tilde{q}^T\end{bmatrix}(L \otimes \begin{bmatrix}2\kappa & 1\\ 1 & 1\end{bmatrix} \otimes I_\mathfrak{p})\begin{bmatrix}\tilde{p}\\ \tilde{q}\end{bmatrix}$. Since $(\mathbf{1}_N \otimes I_\mathfrak{p})^T\tilde{p} = \mathbf{0}_{N\mathfrak{p}}$ and $(\mathbf{1}_N \otimes I_\mathfrak{p})^T\tilde{q} = \mathbf{0}_{N\mathfrak{p}}$, by using Lemma 1, we have $V_1 \geq \frac{\lambda_2(L)}{2}\begin{bmatrix}\tilde{p}^T & \tilde{q}^T\end{bmatrix}(\begin{bmatrix}2\kappa & 1\\ 1 & 1\end{bmatrix} \otimes I_{N\mathfrak{p}})\begin{bmatrix}\tilde{p}\\ \tilde{q}\end{bmatrix}$, where $\lambda_2(L)$ is defined in Lemma 1. Now, using the fact that $\begin{bmatrix} 2\kappa & 1 \\ 1 & 1 \end{bmatrix} > 0$, for $\kappa > \frac{1}{2}$, it is easy to see that V_1 is positive definite. The derivative of V_1 is given as

$$\begin{aligned}
\dot{V}_1 =& 2\kappa\tilde{p}^T(L\otimes I_{\mathfrak{p}})\tilde{q} + \tilde{q}^T(L\otimes I_{\mathfrak{p}})\tilde{q} - \kappa\tilde{p}^T(L\otimes I_{\mathfrak{p}})\tilde{p} + \tilde{p}^T(L\otimes I_{\mathfrak{p}})\big(\kappa r + \kappa v^r + a^r\big) \\
& - \kappa\tilde{p}^T(L\otimes I_{\mathfrak{p}})\tilde{q} - \tilde{p}^T(L\beta\otimes I_{\mathfrak{p}})\mathrm{sgn}[(L\otimes I_{\mathfrak{p}})(\tilde{p}+\tilde{q})] + \tilde{q}^T(L\otimes I_{\mathfrak{p}})\big(\kappa r + \kappa v^r + a^r\big) \\
& - \kappa\tilde{q}^T(L\otimes I_{\mathfrak{p}})\tilde{p} - \kappa\tilde{q}^T(L\otimes I_{\mathfrak{p}})\tilde{q} - \tilde{q}^T(L\beta\otimes I_{\mathfrak{p}})\mathrm{sgn}[(L\otimes I_{\mathfrak{p}})(\tilde{p}+\tilde{q})] \\
=& -\kappa\tilde{p}^T(L\otimes I_{\mathfrak{p}})\tilde{p} - (\kappa-1)\tilde{q}^T(L\otimes I_{\mathfrak{p}})\tilde{q} + (\tilde{p}+\tilde{q})^T(L\otimes I_{\mathfrak{p}})\big(\kappa r + \kappa v^r + a^r\big) \\
& - (\tilde{p}+\tilde{q})^T(L\beta\otimes I_{\mathfrak{p}})\mathrm{sgn}[(L\otimes I_{\mathfrak{p}})(\tilde{p}+\tilde{q})],
\end{aligned}$$

where we have used $LM = L$. Now using the triangular inequality, we have

$$\begin{aligned}
\dot{V}_1 \leq& -\kappa\tilde{p}^T(L\otimes I_{\mathfrak{p}})\tilde{p} - (\kappa-1)\tilde{q}^T(L\otimes I_{\mathfrak{p}})\tilde{q} + \sum_{i=1}^{N}\Big\|\sum_{j=1}^{N}a_{ij}\Big\{(\tilde{p}_i+\tilde{q}_i)-(\tilde{p}_j+\tilde{q}_j)\Big\}\Big\|_1 \times \\
& (\kappa\|r_i\|_1 + \kappa\|v_i^r\|_1 + \|a_i^r\|_1) - \sum_{i=1}^{N}\beta_i\Big\|\sum_{j=1}^{N}a_{ij}\Big\{(\tilde{p}_i+\tilde{q}_i)-(\tilde{p}_j+\tilde{q}_j)\Big\}\Big\|_1 \\
=& -\kappa\tilde{p}^T(L\otimes I_{\mathfrak{p}})\tilde{p} - (\kappa-1)\tilde{q}^T(L\otimes I_{\mathfrak{p}})\tilde{q} + \sum_{i=1}^{N}\Big((\kappa-\eta_i)\|r_i\|_1 + (\kappa-\eta_i)\|v_i^r\|_1 - \gamma\Big)\times \\
& \Big\|\sum_{j=1}^{N}a_{ij}\Big\{(\tilde{p}_i+\tilde{q}_i)-(\tilde{p}_j+\tilde{q}_j)\Big\}\Big\|_1,
\end{aligned}$$

where $\tilde{p}_i$ and $\tilde{q}_i$ are, respectively, the ith components of $\tilde{p}$ and $\tilde{q}$ and we have used the definition of β_i to obtain the last equality. Since $\eta_i > \kappa$, we will have

$$\begin{aligned}
\dot{V}_1 \leq& -\kappa\tilde{p}^T(L\otimes I_{\mathfrak{p}})\tilde{p} - (\kappa-1)\tilde{q}^T(L\otimes I_{\mathfrak{p}})\tilde{q} \\
\leq& -\kappa\lambda_2(L)\tilde{p}^T\tilde{p} - (\kappa-1)\lambda_2(L)\tilde{q}^T\tilde{q} < 0,
\end{aligned}$$

where we have used Lemma 1, and $\kappa > 1$ in second inequality. Now, it is easy to see that $\tilde{p}$ and $\tilde{q}$

are globally exponentially stable, which means

$$\begin{aligned}\lim_{t\to\infty} p_i =& \frac{1}{N}\sum_{j=1}^{N} p_j,\\ \lim_{t\to\infty} q_i =& \frac{1}{N}\sum_{j=1}^{N} q_j.\end{aligned} \tag{4.31}$$

Now, using a procedure similar to proof of Theorem 50, the variables $S_1 = \sum_{i=1}^{N}(p_i - r_i)$ and $S_2 = \sum_{i=1}^{N}(q_i - v_i^r)$ are defined. By summing both sides of (4.26), for $i = 1, ..., N$ we have

$$\begin{aligned}\dot{S}_1 =& S_2,\\ \dot{S}_2 =& -\kappa S_1 - \kappa S_2 - \sum_{i=1}^{N}\beta_i \mathrm{sgn}\Big[\sum_{j=1}^{N} a_{ij}\Big\{(p_i + q_i) - (p_j + q_j)\Big\}\Big].\end{aligned} \tag{4.32}$$

Then we can use input-to-state stability to analyze the system (4.32) by treating the term $\sum_{i=1}^{N}\beta_i \mathrm{sgn}\Big[\sum_{j=1}^{N} a_{ij}\Big\{(p_i + q_i) - (p_j + q_j)\Big\}\Big]$ as the input and S_1 and S_2 as the states. Since $\kappa > 1$, the matrix $\begin{bmatrix}\mathbf{0}_\mathfrak{p} & I_\mathfrak{p}\\ -\kappa I_\mathfrak{p} & -\kappa I_\mathfrak{p}\end{bmatrix}$ is Hurwitz. Thus, the system (4.32) with zero input is exponentially stable and hence input-to-state stable and we have $S_1 \to 0$ and $S_2 \to 0$. Therefore, we have that $\lim_{t\to\infty}\sum_{i=1}^{N} p_i = \sum_{i=1}^{N} r_i$ and $\lim_{t\to\infty}\sum_{i=1}^{N} q_i = \sum_{i=1}^{N} v_i^r$. Now, using (4.31), it is easy to see that the estimation goal (4.30) is achieved.

Controller: Here, each agent tracks its own generated signal, its own estimator output, where it is shown that by using the control inputs (4.27)-(4.29), we have $\lim_{t\to\infty} x_i = p_i$ for $i = 1, \ldots, N$.

Single-integrator: Using the control input (4.27) for (4.1), we obtain the closed-loop dynamics for agents with single-integrator dynamics as

$$\dot{\tilde{x}}_i = -\eta_i \mathrm{sgn}(\tilde{x}_i), \qquad i = 1, \ldots, M-1, \tag{4.33}$$

where $\tilde{x}_i = x_i - p_i$. Consider the candidate Lyapunov function $V_s = \frac{1}{2}\tilde{x}_i^T\tilde{x}_i$. By taking the derivative of V_s, we have $\dot{V}_s = -\eta_i \|\tilde{x}_i\|_1$. It is now easy to conclude that $\tilde{x}_i$, for $i = 1, \ldots, M-1$, converges to zero.

Double-integrator: For agents with double-integrator dynamic the closed-loop system, using the control input (4.28) for (4.2), can be written as

$$\dot{\tilde{v}}_i = -\eta_i \text{sgn}(\tilde{x}_i + \tilde{v}_i) - \eta_i\tilde{x}_i - \eta_i\tilde{v}_i, \qquad i = M, \ldots, N'-1, \tag{4.34}$$

where $\tilde{v}_i = v_i - q_i$. Consider the candidate Lyapunov function $V_d = \frac{1}{2}\begin{bmatrix}\tilde{x}_i^T & \tilde{v}_i^T\end{bmatrix}\begin{bmatrix}2\eta_i I_{\mathfrak{p}} & I_{\mathfrak{p}} \\ I_{\mathfrak{p}} & I_{\mathfrak{p}}\end{bmatrix}$ $\times \begin{bmatrix}\tilde{x}_i \\ \tilde{v}_i\end{bmatrix}$. Since $\eta_i > \frac{1}{2}$, V_d is positive definite. The derivative of V_d along system (4.34) is obtained as

$$\begin{aligned}\dot{V}_d =& 2\eta_i\tilde{x}_i^T\tilde{v}_i + \tilde{v}_i^T\tilde{v}_i - \eta_i\tilde{x}_i^T(\tilde{x}_i + \tilde{v}_i) - \eta_i\tilde{x}_i^T\text{sgn}(\tilde{x}_i + \tilde{v}_i) - \eta_i\tilde{v}^T(\tilde{x}_i + \tilde{v}_i) - \eta\tilde{v}_i^T\text{sgn}(\tilde{x}_i + \tilde{v}_i) \\ =& -\eta_i\tilde{x}_i^T\tilde{x}_i + (1-\eta_i)\tilde{v}_i^T\tilde{v}_i - \eta_i\|\tilde{x}_i + \tilde{v}_i\|_1.\end{aligned}$$

Since $\eta_i > 1$, it is concluded that $\begin{bmatrix}\tilde{x}_i \\ \tilde{v}_i\end{bmatrix}$ for $i = M, \ldots, N'-1$, asymptotically converges to zero.

Euler-Lagrange: It follows form (P3) and (4.29) that the closed-loop dynamics for agent $i = N', \ldots, N$, can be written as

$$M_i(x_i)\dot{s}_i + C(x_i, \dot{x}_i)s_i + Y_i(x_i, \dot{x}_i, p_i, q_i, \dot{q}_i)\theta_i = Y_i(x_i, \dot{x}_i, p_i, q_i, \dot{q}_i)\hat{\theta}_i - \alpha s_i.$$

Consider the candidate Lyapunov function $V_e = \frac{1}{2}s_i^T M_i s_i + \frac{1}{2}\tilde{\theta}_i^T\tilde{\theta}_i$, where $\tilde{\theta}_i = \hat{\theta}_i - \theta_i$. By taking the derivative of V_e, we have

$$\begin{aligned}\dot{V}_e =&\frac{1}{2}s_i^T\dot{M}_i s_i + s_i^T M_i \dot{s}_i + \tilde{\theta}_i^T\dot{\tilde{\theta}}_i\\ =&\frac{1}{2}s_i^T\dot{M}_i s_i - s_i^T C(x_i,\dot{x}_i)s_i + s_i^T Y_i(x_i,\dot{x}_i,p_i,q_i,\dot{q}_i)\tilde{\theta}_i - \alpha s_i^T s_i - \tilde{\theta}_i^T Y_i(x_i,\dot{x}_i,p_i,q_i,\dot{q}_i)^T s_i\\ =&-\alpha s_i^T s_i,\end{aligned}$$

where (P2) is employed to obtain the last equality. Then we can get that $s_i, \tilde{\theta}_i \in \mathbb{L}_\infty$. Also under Assumption 48, it is easy to see that p_i and q_i are bounded. Therefore, using the boundedness of s_i, we know x_i and $\dot{x}_i$ remain bounded. Furthermore, from (4.26), we know that $\dot{q}_i$ is bounded. It follows from (P3) that

$$M_i(x_i)[\mu(q_i - \dot{x}_i) + \dot{q}_i] + C_i(x_i,\dot{x}_i)[\mu(p_i - x_i) + q_i] + g_i(x_i) = Y_i(x_i,\dot{x}_i,p_i,q_i,\dot{q}_i)\theta_i, \quad (4.35)$$

where using the boundedness of its components, we can see that $Y_i(x_i,\dot{x}_i,p_i,q_i,\dot{q}_i)$ is bounded for $i = N',\dots,N$. Now, it follows form (4.35) that $\dot{s}_i$ is bounded. This guarantees the boundedness of $\ddot{V}_e$. Thus by using Lyapunov-like lemma, we have $s_i \to 0$ for $i = N',\dots,N$. Using an argument similar to Lemma 5 in [11], it is obtained that $\lim_{t\to\infty} x_i = p_i$ for $i = N',\dots,N$. Till now it is proved that $\lim_{t\to\infty} x_i = p_i$ for $i = 1,\dots,N$. Now, it follows from (4.30) that the goal (4.5) is achieved. ■

Remark 53 *Note that the restriction in Theorem 52, Assumption 48, is placed by agents with Euler-Lagrange dynamic.*

Remark 54 *Both algorithms introduced in* (4.8)-(4.10) *and* (4.26)-(4.29) *require that Assumptions 48 hold. In algorithm* (4.8)-(4.10)*, the agents just need their own positions and the relative positions between themselves and their neighbors. In some applications, these pieces of information can be obtained by sensing; hence the communication necessity might be eliminated. However, in* *algorithm* (4.26)-(4.29) *each agent must communicate two variables p_i and q_i with its neighbors, which needs communication.*

Remark 55 *The restriction noted in Remark 53 is inevitable when we have an agent with Euler-Lagrange dynamics among our agents. However, for a multi-agent system consisting of agents with only single-integrator and double-integrator dynamics, Assumption 48 will be relaxed in algorithm* (4.26)-(4.29). *As a result, there will be no restriction on admissible reference inputs. Note that in this framework Assumptions 48 can not be relaxed for algorithm* (4.8)-(4.10).

Assumption 56 *The reference velocity $v_i^r(t), \forall i \in \mathcal{I}$ is bounded. It is assumed that $\|v_i^r(t)\| < \bar{v^r}, \forall i \in \mathcal{I}$, where $\bar{v^r}$ is a positive constant.*

Discussion

The algorithm (4.26)-(4.29) can be modified to relax the restriction on $r_i, \forall i = 1, \cdots, N$. Using the modified version, Assumption 48 will be relaxed and we have Assumption 56 instead. By defining $s_i = \dot{x}_i - q_i$ in (4.29) and using a procedure similar to proof of Theorem 52, we can easily show that $s_i \to 0, \forall i = 1, \cdots, N$ asymptotically. This means that $\dot{x}_i \to q_i$ asymptotically and we have

$$\lim_{t\to\infty} ||\dot{x}_i - \frac{1}{N}\sum_{j=1}^{N} v_j^r(t)|| = 0. \tag{4.36}$$

However, the trade-off is that using this modified version, we can only show that $x_i - p_i$ remains bounded and we can not prove that $x_i \to p_i$; hence the goal (4.5) is not achieved. There are applications such as distributed optimization, where by achieving goal (4.36), agents are able to find the optimal point of the team's cost function. In next subsection, it is shown how distributed average tracking can be employed to solve the distributed optimization problem.

4.1.3 The Application of The Distributed Average Tracking in Distributed Optimization

We are interested in a distributed convex optimization problem for heterogeneous multi-agent system. Consider a heterogeneous multi-agent system consisting of N physical agents. The agents' are described by single-integrator (4.1) and double-integrator dynamics (4.2). A local cost function $f_i : R^m \to R$ is assigned to agent i which is known to only agent i. The team cost function $f : R^m \to R$ is denoted by $f(x) \triangleq \sum_{i=1}^{N} f_i(x)$. Our objective is that $x(t)$ converges to x^*, the minimizer of the function $f(x)$, using only local information and local interaction with neighbors. Assume that using distributed average tracking algorithms proposed in previous subsections, goals (4.36) is achieved. Defining $v_i^r = -\nabla f_i(x_i)$, we have

$$\dot{x}_i \to \frac{1}{N}\sum_{j=1}^{N} -\nabla f_j(x_j),$$

which can be written as

$$\dot{x}_i = \frac{1}{N}\sum_{j=1}^{N} -\nabla f_j(x_j) + \zeta_i, \tag{4.37}$$

where $\zeta_i \to 0$. Define $X = \frac{1}{N}\sum_{j=1}^{N} x_j$ and $\eta_i = x_i - X$, where $\eta_i \to 0$. Now, we can rewrite (4.37) as

$$\dot{X} = -\frac{1}{N}\sum_{j=1}^{N} \left[\nabla f_j(X + \eta_j) + \zeta_j\right], \tag{4.38}$$

Assumption 57 *Function $f(x) \triangleq \sum_{i=1}^{N} f_i(x)$ is strongly convex, where we have $(\nabla f(x) - \nabla f(y))^T(x - y) \geq m \left\| x - y \right\|_2^2$.*

Under Assumption 57, there exists a unique optimal X^* for function $f(x)$. In the remaining, we will show that $X \to X^*$. Define the Lyapanuv candidate $W = (X - X^*)^T(X - X^*)$. The time derivative of W along system (4.38) is obtained as

$$\begin{aligned}
\dot{W} =& -\frac{1}{N}(X - X^*)^T\big(\sum_{j=1}^{N}\left[\nabla f_j(X+\eta_j)+\zeta_j\right]\big)\\
=& -\frac{1}{N}(X - X^*)^T\big(\sum_{j=1}^{N}\nabla f_j(X+\eta_j)+\sum_{j=1}^{N}\nabla f_j(X)+\sum_{j=1}^{N}\nabla f_j(X^*)-\sum_{j=1}^{N}\nabla f_j(X)+\sum_{j=1}^{N}\zeta_j\big)\\
\leq& \frac{1}{N}\left\|X - X^*\right\|\left\|\sum_{j=1}^{N}\nabla f_j(X+\eta_j)+\sum_{j=1}^{N}\nabla f_j(X)\right\|\\
&-\frac{1}{N}(X - X^*)^T\big(\sum_{j=1}^{N}\nabla f_j(X+\eta_j)+\sum_{j=1}^{N}\nabla f_j(X)+\sum_{j=1}^{N}\nabla f_j(X^*)-\sum_{j=1}^{N}\nabla f_j(X)+\sum_{j=1}^{N}\zeta_j\big),
\end{aligned}$$

where in the second equality $\sum_{j=1}^{N}\nabla f_j(X) - \sum_{j=1}^{N}\nabla f_j(X) = 0$ is added, and we also used the fact that $\sum_{j=1}^{N}\nabla f_j(X^*) = 0$.

Assumption 58 *The gradient of the local cost function, $\nabla f_i(x_i)$, is lipschitz for $\forall i \in \mathcal{I}$.*

Under Assumption 58, we have $\|\nabla f_i(x) - \nabla f_i(y)\| \leq \rho_i\|x - y\|$. Therefore, $\frac{1}{N}\left\|X - X^*\right\|$ $\left\|\sum_{j=1}^{N}\nabla f_j(X+\eta_j)+\sum_{j=1}^{N}\nabla f_j(X)\right\| \leq \frac{1}{N}\left\|X - X^*\right\|\big(\sum_{j=1}^{N}\rho_j\left\|\eta_j\right\|\big)$. Also under Assumption 57, we have $-\frac{1}{N}(X - X^*)^T\big(\sum_{j=1}^{N}\nabla f_j(X^*) - \sum_{j=1}^{N}\nabla f_j(X)\big) \leq \frac{m}{N}\left\|X - X^*\right\|^2$, where m is the strongly convex coefficient of the function $f(x)$ defined in Assumption 57. Therefore we have

$$\dot{W} \leq \frac{1}{N}\left\|X - X^*\right\|\big(\sum_{j=1}^{N}\rho_j\left\|\eta_j\right\| + \left\|\zeta_j\right\|\big) - \frac{1}{N}m\left\|X - X^*\right\|^2,$$

where $\|\eta_j\| \to 0$ and $\|\zeta_j\| \to 0, \forall j \in \mathcal{I}$. We can conclude that $X - X^* \to 0$.

4.2 Simulation and Discussion

In this section, we present a simulation to illustrate the theoretical result in Subsection 4.1.1. Consider a team of six agents. The interaction among the agents is described by an undirected graph shown in Fig. 4.1, where agents are colored based on their dynamics. Agents with single-integrator, double-integrator, and Euler-Lagrange dynamics are, respectively, colored red, blue and green.

The agents' goal is to track the average of their reference inputs. The reference input for agent i is defined as $r_i(t) = \begin{bmatrix} 3i\text{sin}(\frac{\pi}{25}t) \\ 4i\text{cos}(\frac{\pi}{50}t) \end{bmatrix}$. The reference input and its velocity is bounded and Assumption 48 is satisfied. The dynamic for agents with single-integrator and double-integrator dynamics is defined as (4.1) and (4.2). The dynamic equation for each Euler-Lagrange agent is modeled by $m_i\ddot{x}_i + c_i\dot{x}_i = u_i, i = 5, 6$, where $x_i(t)$ is the coordinate of agent i in $2D$ plane [10]. The parameters m_i and c_i represent, respectively, the mass and damping constants of the agent i, which are assumed to be constant but unknown. We let $m_1 = 1, c_1 = 0.5, m_2 = 1.5$, and $c_2 = 0.6$.

In our example, we apply the algorithm (4.8)-(4.10), where the controllers' parameters are selected as $\beta_i = 25, \forall i \in \mathcal{I}$, and $\alpha = 15$. The initial positions of the agents are selected as $[8\ 0]^T, [9\ 3]^T, [10\ 6]^T, [11\ 9]^T, [12\ 12]^T$, and $[13\ 15]^T$ and their initial velocities are selected as zero. Fig. 4.2 shows that the distributed average tracking is achieved and agents track the average of the reference inputs.

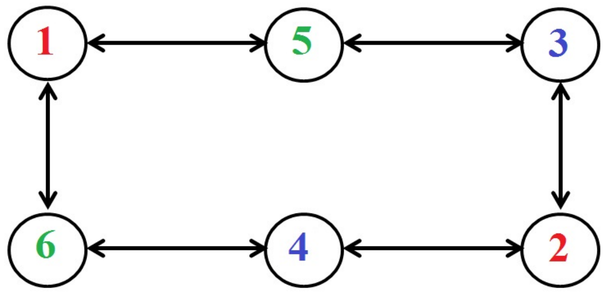

Figure 4.1: Undirected graph

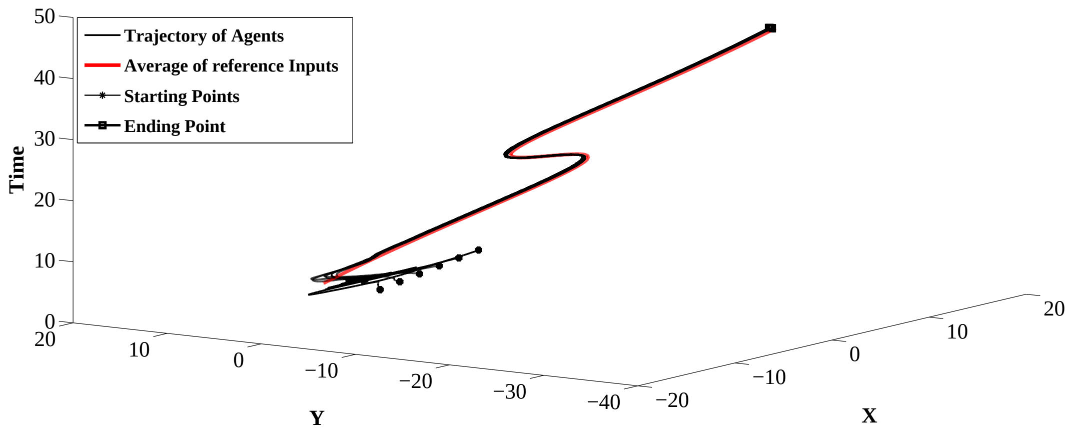

Figure 4.2: Trajectories of all agents along with the average of the reference inputs using the algorithm (4.8)-(4.10).

Chapter 5

Conclusions

In this dissertation, two optimization problems were investigated, where two different methods of game theory and convex optimization were employed to find optimal solutions for multi-agent systems' behaviors. Here, we addressed the following subjects which are listed in the order of appearance

1. Distributed Coverage Control of Mobile Sensor Networks in Unknown Environment Using Game Theory,
2. Distributed Continuous-Time Convex Optimization for Time-Varying Cost Functions.

To solve the first issue, we have discussed the sensor coverage problem using a MSN, where the mobile sensor agents' task is to reposition themselves from their random initial locations to their optimal configuration. We modelled the optimization problem as a non-cooperative game, where each agent as a selfish, logical, decision maker chooses its action to increase its own utility function. A utility function has been proposed based on the marginal contribution of each agent on sensing the area while considering the energy consumption of the MSN. Using the defined utility function, a game has been introduced and it has been proved that the game is a state-based potential game. In order to converge to the Nash equilibrium, each agent applied binary log-linear learning to update its own action at each iteration. Taking into account the fact that an agent needs the knowledge about the worth of its action set to use this learning algorithm, an estimation scheme has been introduced to predict the worth of the uncovered regions using the agents' previous observations. The GMM has been used to model the wroth of the area. The ML method has been employed to estimate the unknown parameters of the GMM, where the utility functions of future actions have been calculated using estimated parameters. In the last part of chapter 2, the mutual information term has been added to the agent's utility function in order to find an action which creates more informative observations. The simulation and experimental results illustrated that the algorithm could find the proper configuration for mobile sensors. The experimental videos can be found at Without Mutual Information and With Mutual Information.

In the second part of the dissertation, a time-varying distributed convex optimization problem was studied for continuous-time multi-agent systems, where the objective was to minimize the sum of the local time-varying cost functions. Each local cost function was only known to an individual agent. Control algorithms have been designed for the cases of single-integrator and double-integrator dynamics. In both cases, as a first step, a centralized approach has been introduced to solve the optimization problem for convex time-varying cost functions. Then this problem has been solved in a distributed manner and a discontinuous algorithm with adaption gains has been proposed, where it was possible to rely on only local sensing. To relax the restricted assumption imposed on the feasible cost functions, an estimator based algorithm has been proposed where the agents used dynamic average tracking as a tool to estimate the centralized control input. However, the necessity of communication between neighbors was the drawback of the estimator based algorithm. Then in the case of double-integrator dynamics, we have focused on extending our proposed algorithms to improve them for real applications. Two continuous algorithms have been proposed which employed continuous approximations of the signum function. The first continuous algorithm used a time-varying approximation of the signum function, where we have shown that the team cost function was minimized and the agents reach consensus. In the second continuous algorithm, a time-invariant approximation of the signum function has been used which was easier to implement. The trade-off was that there exists a bounded error between the agents and the optimal trajectory. To add the inter-agent collision avoidance capability into our algorithms, two distributed convex optimization algorithms with swarm tracking behavior have been proposed for single-integrator and double-integrator dynamics.

In the last part of the dissertation, a distributed average tracking was studied for a group of heterogeneous physical agents. The multi-agent system was consisted of the agents with single-integrator, double-integrator and Euler-Lagrange dynamics. Two nonsmooth algorithms were proposed to achieve the distributed average tracking goal. In our first proposed algorithm, each agent required to have access to only its own position and the relative positions between itself and its neighbors, where it was possible to rely on only local sensing. To relax some restrictive assumptions on admissible reference inputs, we proposed the second algorithm, where a filter was introduced for each agent to generate an estimation of the average reference inputs. Then, each agent tracked its own generated signal. Finally, it was shown that how the proposed algorithms can be employed to solve the distributed optimization problem for a heterogeneous framework.

Bibliography

[1] Rosario Aragues, Jorge Cortes, and Carlos Sagues. Distributed consensus on robot networks for dynamically merging feature-based maps. *IEEE Transactions on Robotics*, 28(4):840–854, 2012.

[2] H. Bai, R. A. Freeman, and K. M. Lynch. Distributed kalman filtering using the internal model average consensus estimator. In *Proceedings of the American Control Conference*, San Francisco, 2011.

[3] He Bai, R.A Freeman, and K.M. Lynch. Robust dynamic average consensus of time-varying inputs. In *Proceedings of the IEEE Conference on Decision and Control*. Atlanta, 2010.

[4] Mokhtar S. Bazaraa, Hanif D. Sherali, and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons, Inc., 2005.

[5] Lawrence E. Blume. The statistical mechanics of strategic interaction. *Games and Economic Behavior*, 5(3):387 – 424, 1993.

[6] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, June 1994.

[7] S. Boyd and L. Vandenbergher. *Convex Optimization*. Cambridge University Press, 2004.

[8] Yongcan Cao and Wei Ren. Distributed coordinated tracking with reduced interaction via a variable structure approach. *Automatic Control, IEEE Transactions on*, 57(1):33–48, Jan 2012.

[9] Chien Chern Cheah, Saing Paul Hou, and Jean Jacques E Slotine. Region-based shape control for a swarm of robots. *Automatica*, 45(10):2406–2411, 2009.

[10] Chien Chern Cheah, Saing Paul Hou, and Jean Jacques E. Slotine. Region-based shape control for a swarm of robots. *Automatica*, 45(10):2406 – 2411, 2009.

[11] F. Chen, W. Ren, W. Lan, and G. Chen. Tracking the average of time-varying nonsmooth signals for double-integrator agents with a fixed topology. In *American Control Conference*, June 2013.

[12] Fei Chen, Yongcan Cao, and Wei Ren. Distributed average tracking of multiple time-varying reference signals with bounded derivatives. *IEEE Transactions on Automatic Control*, 57(12):3169–3174, 2012.

[13] Fei Chen, Yongcan Cao, and Wei Ren. Distributed average tracking of multiple time-varying reference signals with bounded derivatives. *Automatic Control, IEEE Transactions on*, 57(12):3169–3174, Dec 2012.

[14] Fei Chen, Gang Feng, Lu Liu, and Wei Ren. Distributed average tracking of networked Euler-Lagrange systems. *IEEE Transactions on Automatic Control*, 60(2):547–552, 2015.

[15] Fei Chen, Wei Ren, Weiyao Lan, and Guanrong Chen. Tracking the average of time-varying nonsmooth signals for double-integrator agents with a fixed topology. In *American Control Conference*, 2013.

[16] Ashish Cherukuri and Jorge Cortes. Initialization-free distributed coordination for economic dispatch under varying loads and generator commitment. In *arXiv*, 2014.

[17] F.R.K. Chung. *Spectral Graph Theory*. Number 92 in CBMS Regional Conference Series. Conference Board of the Mathematical Sciences.

[18] F. Clarke. *Optimization and Nonsmooth Analysis*. Society for Industrial and Applied Mathematics, 1990.

[19] J. Cortes. Discontinuous dynamical systems. *Control Systems, IEEE*, 28(3):36–73, June 2008.

[20] J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation,*, 20(2):243–255, April 2004.

[21] Felipe Cucker and Steve Smale. Emergent behavior in flocks. *IEEE Transactions on Automatic Control*, pages 852–862, 2007.

[22] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society Series B*, 39(1):1–38, 1977.

[23] Greg Droge, Hiroaki Kawashima, and Magnus B. Egerstedt. Continuous-time proportional-integral distributed optimisation for networked systems. *Journal of Control and Decision*, 1(3):191–213, 2014.

[24] C. Edwards and S. Spurgeon. *Sliding Mode Control: Theory And Applications*. Series in Systems and Control. Taylor & Francis, 1998.

[25] A.F. Filippov. *Differential Equations with Discontinuous Righthand Sides*. Springer Netherlands, 1988.

[26] N. Fischer, R. Kamalapurkar, and W. E. Dixon. Lasalle-yoshizawa corollaries for nonsmooth systems. *IEEE Transactions on Automatic Control*, 58(9):2333–2338, Sept 2013.

[27] R. A. Freeman, Peng Yang, and K. M. Lynch. Stability and convergence properties of dynamic average consensus estimators. In *Proceedings of the IEEE Conference on Decision and Control*. San Diego, 2006.

[28] Drew Fudenberg and David K. Levine. *The Theory of Learning in Games*, volume 1 of *MIT Press Books*. The MIT Press, January 1998.

[29] S. Ghapani, S. Rahili, and W. Ren. Distributed average tracking for second-order agents with nonlinear dynamics. In *American Control Conference*, Boston, MA, July 2016.

[30] Sheida Ghapani, Wei Ren, and Fei Chen. Distributed average tracking for double-integrator agents without using velocity measurements. In *American Control Conference, 2015*. IEEE, 2015.

[31] B. Gharesifard and J. Cortes. Distributed continuous-time convex optimization on weight-balanced digraphs. *Automatic Control, IEEE Transactions on*, 59(3):781–786, March 2014.

[32] T. Goto, T. Hatanaka, and M. Fujita. Payoff-based inhomogeneous partially irrational play for potential game theoretic cooperative control: Convergence analysis. In *American Control Conference*, pages 2380–2387, June 2012.

[33] Carlos Guestrin, Andreas Krause, and Ajit Paul Singh. Near-optimal sensor placements in Gaussian processes. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 265–272, New York, 2005.

[34] J-B. Hiriart-Urruty. Mean value theorems in nonsmooth analysis. *Numerical Functional Analysis and Optimization*, 2(1):1–30, 1980.

[35] B. Johansson, T. Keviczky, M. Johansson, and K.H. Johansson. Subgradient methods and consensus algorithms for solving convex optimization problems. In *Decision and Control IEEE Conference on*, pages 4185–4190, Dec 2008.

[36] S. S. Kia, J. Cortés, and S. Martnez. Dynamic average consensus under limited control authority and privacy requirements. *International Journal of Robust and Nonlinear Control*, 2014.

[37] Solmaz S. Kia, Jorge Cortes, and Sonia Martinez. Distributed convex optimization via continuous-time coordination algorithms with discrete-time communication. *Automatica*, 55, 2015.

[38] K. Kvaternik and L. Pavel. A continuous-time decentralized optimization scheme with positivity constraints. In *IEEE Conference on Decision and Control*, pages 6801–6807, Dec 2012.

[39] V. Lakshmanan and J. S. Kain. A Gaussian Mixture Model Approach to Forecast Verification. *Weather and Forecasting*, 25:908–920, 2010.

[40] Wei Li and C.G. Cassandras. Distributed cooperative coverage control of sensor networks. In *The IEEE Conference on Decision and Control and European Control Conference*, pages 2542–2547, Dec. 2005.

[41] Yusun Lim and Jeff S. Shamma. Robustness of stochastic stability in game theoretic learning. In *American Control Conference*, pages 6145–6150, 2013.

[42] Peng Lin, Wei Ren, Yongduan Song, and J.A Farrell. Distributed optimization with the consideration of adaptivity and finite-time convergence. In *American Control Conference, 2014*, pages 3177–3182, June 2014.

[43] Qing Ling and Alejandro Ribeiro. Decentralized dynamic optimization through the alternating direction method of multipliers. *IEEE Transactions on Signal Processing*, 62(5):1185–1197, 2014.

[44] Yuan Liu, Haibo Min, Shicheng Wang, Zhiguo Liu, and Shouyi Liao. Distributed consensus of a class of networked heterogeneous multi-agent systems. *Journal of the Franklin Institute*, 351(3):1700 – 1716, 2014.

[45] S. G. Loizou and K. J. Kyriakopoulos. Navigation of multiple kinematically constrained robots. *IEEE Transactions on Robotics*, 24(1):221–231, Feb 2008.

[46] Jie Lu and Choon Yik Tang. Zero-gradient-sum algorithms for distributed convex optimization: The continuous-time case. In *American Control Conference , 2011*, pages 5474–5479, June 2011.

[47] Jason R. Marden. State based potential games. *Automatica*, 48(12):3075–3088, 2012.

[48] Jason R. Marden and Jeff S. Shamma. Revisiting log-linear learning: Asynchrony, completeness and payoff-based implementation. *Games and Economic Behavior*, 75(2):788 – 808, 2012.

[49] Jason R Marden and Adam Wierman. Distributed welfare games. *Operations Research*, 61(1):155–168, 2013.

[50] J.R. Marden and A. Wierman. Distributed welfare games with applications to sensor coverage. In *IEEE Conference on Decision and Control*, pages 1708–1713, Dec. 2008.

[51] G.J. McLachlan and K.E. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York, 1988.

[52] J. Mei, W. Ren, and J. Chen. Distributed consensus of second-order multi-agent systems with heterogeneous unknown inertias and control gains under a directed graph. *IEEE Transactions on Automatic Control*, 61(8):2019–2034, 2016.

[53] A Nedic, A Ozdaglar, and P.A Parrilo. Constrained consensus and optimization in multi-agent networks. *Automatic Control, IEEE Transactions on*, 55(4):922–938, April 2010.

[54] Shahram Nosrati, Masoud Shafiee, and Mohammad Bagher Menhaj. Dynamic average consensus via nonlinear protocols. *Automatica*, 48(9):2262 – 2270, 2012.

[55] R. Olfati-Saber and R.M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *Automatic Control, IEEE Transactions on*, 49(9):1520–1533, Sept 2004.

[56] Reza Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *Automatic Control, IEEE Transactions on*, 51(3):401–420, 2006.

[57] Reza Olfati-Saber and Richard M Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, 2004.

[58] D. Pescaru, C. Istin, D. Curiac, and A. Doboli. Energy saving strategy for video-based wireless sensor networks under field coverage preservation. In *IEEE International Conference on Automation, Quality and Testing, Robotics*, pages 289–294, 2008.

[59] Z. Qu. *Cooperative Control of Dynamical Systems: Applications to Autonomous Vehicles*. Springer, 2009.

[60] Michael Rabbat and Robert Nowak. Distributed optimization in sensor networks. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, pages 20–27, New York, NY, USA, 2004. ACM.

[61] S. Rahili and W. Ren. Game theory control solution for sensor coverage problem in unknown environment. In *IEEE Conference on Decision and Control*, pages 1173–1178, Dec 2014.

[62] S. Rahili and W. Ren. Distributed continuous-time convex optimization with time-varying cost functions. *IEEE Transactions on Automatic Control*, PP(99):1–1, 2016.

[63] S. Rahili, W. Ren, and S. Ghapani. Distributed convex optimization of time-varying cost functions with swarm tracking behavior for continuous-time dynamics. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 362–367, Dec 2015.

[64] Salar Rahili, Wei Ren, and Peng Lin. Distributed convex optimization of time-varying cost functions for double-integrator systems using nonsmooth algorithms. In *American Control Conference*, Chicago, IL, July 2015.

[65] S. Ram, A. Nedic, and V. Veeravalli. Incremental stochastic subgradient algorithms for convex optimization. *SIAM Journal on Optimization*, 20(2):691–717, 2009.

[66] William H. Sandholm. H. peyton young, strategic learning and its limits , oxford univ. press (2004) 165 pages. *Games and Economic Behavior*, 63(1):417–420, May 2008.

[67] M. Schwager, J.-J. Slotine, and D. Rus. Decentralized, adaptive control for coverage with networked robots. In *IEEE International Conference on Robotics and Automation*, pages 3289–3294, April 2007.

[68] Sambu Seo, Marko Wallat, Thore Graepel, and Klaus Obermayer. Gaussian process regression: Active data selection and test point rejection. In *IJCNN (3)*, pages 241–246, 2000.

[69] M. C. Shewry and H. P. Wynn. Maximum entropy sampling. *Journal of Applied Statistics*, 14(2):165–170, 1987.

[70] Guodong Shi, K.H. Johansson, and Yiguang Hong. Reaching an optimal consensus: Dynamical systems that compute intersections of convex sets. *Automatic Control, IEEE Transactions on*, 58(3):610–622, March 2013.

[71] Andrea Simonetto, Leon Kester, and Geert Leus. Distributed time-varying stochastic optimization and utility-based communication. In *arXiv*, 2014.

[72] J.J.E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice Hall, 1991.

[73] Demetri P Spanos, Reza Olfati-Saber, and Richard M Murray. Dynamic consensus on mobile networks. In *Proceedings of The 16th IFAC World Congress*. Prague Czech Republic, 2005.

[74] Seth Hutchinson Spong, Mark W. and M. Vidyasagar. *Robot Modeling and Control*. Hoboken, NJ: John Wiley and Sons, 2006.

[75] Mani Srivastava, Richard Muntz, and Miodrag Potkonjak. Smart kindergarten: sensor-based wireless networks for smart developmental problem-solving environments. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pages 132–138, New York, NY, 2001.

[76] Housheng Su, Xiaofan Wang, and Zongli Lin. Flocking of multi-agents with a virtual leader. *Automatic Control, IEEE Transactions on*, 54(2):293–307, Feb 2009.

[77] W. Su. *Traffic Engineering and Time-Varying Convex Optimization*. BiblioBazaar, 2011.

[78] Sheng Yuan Tu and A.H. Sayed. Mobile adaptive networks. *Selected Topics in Signal Processing, IEEE Journal of*, 5(4):649–664, Aug 2011.

[79] Jing Wang and N. Elia. A control perspective for centralized and distributed convex optimization. In *Decision and Control and European Control Conference*, pages 3800–3805, Dec 2011.

[80] Long Wang and Feng Xiao. Finite-time consensus problems for networks of dynamic agents. *Automatic Control, IEEE Transactions on*, 55(4):950–955, April 2010.

[81] Xiaoli Wang and Yiguang Hong. Finite-time consensus for multi-agent networks with second-order agent dynamics. In *Proceedings of IFAC World Congress*, 2008.

[82] David H. Wolpert and Kagan Tumer. An introduction to collective intelligence. Technical report, Handbook of Agent Technology. AAAI, 1999.

[83] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Computer Networks*, 52(12):2292 – 2330, 2008.

[84] Deming Yuan, Shengyuan Xu, and Huanyu Zhao. Distributed primal-dual subgradient method for multiagent optimization via consensus algorithms. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 41(6):1715–1724, Dec 2011.

[85] Yu Zhao, Zhisheng Duan, and Zhongkui Li. Distributed average tracking for multiple reference signals with general linear dynamics. *arXiv*, 2013.

[86] Yu Zhao, Zhisheng Duan, and Zhongkui Li. Distributed average tracking for multiple reference signals with general linear dynamics. In *arXiv*, 2013.

[87] Y. Zheng, Y. Zhu, and L. Wang. Consensus of heterogeneous multi-agent systems. *IET Control Theory Applications*, 5(16):1881–1888, 2011.

[88] M. Zhu and S. Martínez. Distributed coverage games for energy-aware mobile sensor networks. *SIAM Journal on Control and Optimization*, 51(1):1–27, 2013.

[89] Yi Zou and K. Chakrabarty. Sensor deployment and target localization based on virtual forces. In *The 22d Annual Joint Conference of the IEEE Computer and Communications.*, pages 1293–1303, March 2003.

.1 Appendix

In this appendix an explanation is given on how Assumptions 17 and 26 can be, respectively, satisfied in Theorem 18 and 27. We focus on the more involved case in Theorem 27 while a similar argument holds in the case in Theorem 18. Here we show that Assumption 26 holds if the cost functions with identical Hessians satisfy certain conditions, referred as Condition ($\star$) for convenience, such that the boundedness of $\|x_i(t) - x_j(t)\|$ and $\|v_i(t) - v_j(t)\|$ for all t guarantees the boundedness of $\|\nabla f_j(x_j, t) - \nabla f_i(x_i, t)\|$, $\left\|\frac{d\nabla f_j(x_j,t)}{dt} - \frac{d\nabla f_i(x_i,t)}{dt}\right\|$ and $\left\|\frac{\partial^2 \nabla f_j(x_j,t)}{\partial t^2} - \frac{\partial^2 \nabla f_i(x_i,t)}{\partial t^2}\right\|$ for all t. As a result, if Condition ($\star$) is satisfied, then Assumption 4.2 holds.

In particular, we will show that under Condition ($\star$), there always exists a finite $\bar{\phi}$ which can be determined at time $t = 0$. With $\bar{\phi}$ determined, using the algorithm (3.29), $\|x_i(t) - x_j(t)\|$ and $\|v_i(t) - v_j(t)\|, \forall t$, and $\forall i, j \in \mathcal{I}$ will remain bounded for all time, which implies that Assumption 26 holds. We show the argument in four steps.

1. With identical Hessians, Assumption 26 holds if $\|\nabla f_j(x_j, t) - \nabla f_i(x_i, t)\|$, $\left\|\frac{d\nabla f_j(x_j,t)}{dt} - \frac{d\nabla f_i(x_i,t)}{dt}\right\|$ and $\left\|\frac{\partial^2 \nabla f_j(x_j,t)}{\partial t^2} - \frac{\partial^2 \nabla f_i(x_i,t)}{\partial t^2}\right\|$, $\forall t$ and $\forall i, j \in \mathcal{I}$ are bounded. Assume that the boundedness of $\|x_i(t) - x_j(t)\|$ and $\|v_i(t) - v_j(t)\|$ for all t guarantees the boundedness of $\|\nabla f_j(x_j, t) - \nabla f_i(x_i, t)\|$, $\left\|\frac{d\nabla f_j(x_j,t)}{dt} - \frac{d\nabla f_i(x_i,t)}{dt}\right\|$ and

$\left\| \frac{\partial^2 \nabla f_j(x_j,t)}{\partial t^2} - \frac{\partial^2 \nabla f_i(x_i,t)}{\partial t^2} \right\|$, $\forall t$ and $\forall i, j \in \mathcal{I}$. This is ensured by Condition ($\star$).

Denote the upper bounds on $\|x_i(t) - x_j(t)\|$ and $\|v_i(t) - v_j(t)\|$, $\forall t$ and $\forall i, j \in \mathcal{I}$ as, respectively, β_x and β_v. It is easy to see that if there exist constant β_x and β_v, there exists a constant $\bar{\phi}$, which in turn guarantees the existence of bounded $\bar{\beta}$ where $\bar{\beta} > \frac{(N-1)\bar{\phi}}{2}$.

2. Our proof will be completed if we can show that there exist constant β_x and β_v. In the remaining, we will show that not only there exist constant β_x and β_v, but also it is sufficient to determine these constants using the agents' initial states. Two conservative β_x and β_v are selected using the initial states as

$$\begin{aligned}\beta_x =& 2\sqrt{\frac{m\lambda_{max}[P]}{N\lambda_{min}[P]}}\big(\max_i \left\| \sum_{j=1}^{N}(x_i(0) - x_j(0)) \right\|_\infty + \max_i \left\| \sum_{j=1}^{N}(v_i(0) - v_j(0)) \right\|_\infty \big) + \gamma \\ \beta_v =& 2\sqrt{\frac{m\lambda_{max}[P]}{N\lambda_{min}[P]}}\big(\max_i \left\| \sum_{j=1}^{N}(x_i(0) - x_j(0)) \right\|_\infty + \max_i \left\| \sum_{j=1}^{N}(v_i(0) - v_j(0)) \right\|_\infty \big) + \gamma,\end{aligned} \tag{.1}$$

where γ is a positive constant, $\lambda_{min}[P] > 0$, and $\lambda_{max}[P] > 0$, respectively, are the smallest and largest eigenvalues of P defined in (3.34). Now, we will show that $\|x_i(t) - x_j(t)\| \leq \beta_x$, $\|v_i(t) - v_j(t)\| \leq \beta_v$, $\forall i, j \in \mathcal{I}$ and $\forall t$.

3. We know that for W defined in (3.34), and for $\bar{\beta}$ selected based on the introduced constants β_x and β_v, we have $\dot{W}_{|t=0} < 0$. We will use a contradiction approach to show that such $\bar{\beta}$ ensures $\dot{W} \leq 0$ for all t. Assume that there exists a time $t = t_\epsilon$ at which $\dot{W}$ becomes positive, i.e, $\dot{W}_{|t=t_\epsilon} > 0$. By recalling the selected conservative $\bar{\beta}$, this is only possible if one or more of the constants β_x, and β_v do not exist. This means that there exist two agents k and l such that $\|x_k(t_\epsilon) - x_l(t_\epsilon)\| > \beta_x$ or $\|v_k(t_\epsilon) - v_l(t_\epsilon)\| > \beta_v$.

Let us first suppose that $\|x_k(t_\epsilon) - x_l(t_\epsilon)\| > \beta_x$. Note that that $\dot{W}(t) \leq 0$, $\forall t \in [0, t_\epsilon)$, which means that $W(t) \leq W(0)$, $\forall t \in [0, t_\epsilon)$. Using (3.34), it is easy to see that $\forall\, t \in [0, t_\epsilon)$

we have $\sqrt{\frac{\lambda_{max}[P]}{\lambda_{min}[P]}} \left\| \begin{pmatrix} e_X(0) \\ e_V(0) \end{pmatrix} \right\|_2 \geq \left\| \begin{pmatrix} e_X(t) \\ e_V(t) \end{pmatrix} \right\|_2$. Now, using the graph connectivity and the properties of the norms, it is easy to show that

$$\begin{aligned}
\beta_x - \gamma &= 2\sqrt{\frac{m\lambda_{max}[P]}{N\lambda_{min}[P]}} \Big(\max_i \left\| \sum_{j=1} (x_i(0) - x_j(0)) \right\|_\infty + \max_i \left\| \sum_{j=1} (v_i(0) - v_j(0)) \right\|_\infty \Big) \\
&= 2\sqrt{\frac{Nm\lambda_{max}[P]}{\lambda_{min}[P]}} (\|e_X(0)\|_\infty + \|e_V(0)\|_\infty) \geq 2\sqrt{\frac{Nm\lambda_{max}[P]}{\lambda_{min}[P]}} \left\| \begin{pmatrix} e_X(0) \\ e_V(0) \end{pmatrix} \right\|_\infty \\
&\geq 2\sqrt{\frac{\lambda_{max}[P]}{\lambda_{min}[P]}} \left\| \begin{pmatrix} e_X(0) \\ e_V(0) \end{pmatrix} \right\|_2 \geq 2 \left\| \begin{pmatrix} e_X(t) \\ e_V(t) \end{pmatrix} \right\|_2 \\
&\geq 2 \|e_X(t)\|_2 \geq 2 \max_i \left\| x_i(t) - \frac{1}{N} \sum_{j=1} x_j(t) \right\|_\infty \\
&\geq \|x_k(t) - x_l(t)\|, \forall k, l \in \mathcal{I}, \text{and } \forall\, t \in [0, t_\epsilon).
\end{aligned}$$

Now, under the assumption of $\|x_k(t_\epsilon) - x_l(t_\epsilon)\| > \beta_x$ and using the selected β_x in (.1), we have $\|x_k(t_\epsilon) - x_l(t_\epsilon)\| - \lim_{t \to t_\epsilon^-} \|x_k(t) - x_l(t)\| > \gamma$. However, $\lim_{t \to t_\epsilon^-} \|x_k(t) - x_l(t)\| \neq \|x_k(t_\epsilon) - x_l(t_\epsilon)\|$, contradicts with the continuity of the agents' positions which is mentioned in Remark 15. Therefore, we have $\|x_i(t) - x_j(t)\| \leq \beta_x,\ \forall i, j \in \mathcal{I}$ and $\forall t$. The same argument can be made for showing $\|v_i(t) - v_j(t)\| \leq \beta_v,\ \forall i, j \in \mathcal{I}$ and $\forall t$, which is omitted here. We thus conclude that there exists no time $t = t_\epsilon$, where $\dot{W}_{|t=t_\epsilon} > 0$ and it completes the proof.

For example, to satisfy Assumption 26 for the local cost function $f_i(x_i, t) = (ax_i + g_i(t))^2$ defined in Remark 20, it is only required to satisfy Condition ($\star$). It is easy to see that Condition ($\star$) boils down to the boundedness of $\|g_i(t) - g_j(t)\|$, $\|\dot{g}_i(t) - \dot{g}_j(t)\|$, and $\|\ddot{g}_i(t) - \ddot{g}_j(t)\|$, $\forall t$ and $\forall i, j \in \mathcal{I}$. Hence the boundedness of $\|g_i(t) - g_j(t)\|$, $\|\dot{g}_i(t) - \dot{g}_j(t)\|$, and $\|\ddot{g}_i(t) - \ddot{g}_j(t)\|$, $\forall t$ and $\forall i, j \in \mathcal{I}$ is sufficient to ensure that Assumption 26 holds.

A similar argument can be done for satisfying Assumption 17 in Theorem 18. Here for cost functions with identical Hessians, Condition ($\star$) is such that the boundedness of $\|x_i(t) - x_j(t)\|$ for all t guarantees the boundedness of $\|\nabla f_j(x_j, t) - \nabla f_i(x_i, t)\|$ and $\left\|\frac{\partial \nabla f_j(x_j,t)}{\partial t} - \frac{\partial \nabla f_i(x_i,t)}{\partial t}\right\|$ $\forall i, j \in \mathcal{I}$ for all t. As it is mentioned in Remark 20, for the local cost function $f_i(x_i, t) = (ax_i + g_i(t))^2$, Condition ($\star$) boils down to the boundedness of $\|g_i(t) - g_j(t)\|$ and $\|\dot{g}_i(t) - \dot{g}_j(t)\|, \forall t$ and $\forall i, j \in \mathcal{I}$. Hence the boundlessness of $\|g_i(t) - g_j(t)\|$ and $\|\dot{g}_i(t) - \dot{g}_j(t)\|$ is sufficient to ensure that Assumption 17 holds.

.2 Appendix

In this appendix, we clarify how the boundedness of $\phi_i, \forall i \in \mathcal{I}, \forall t$, in Theorem 42, can be satisfied. In particular, we show that for bounded $\|g_i(t)\|$ and $\|\dot{g}_i(t)\|, \forall t$ and $\forall i \in \mathcal{I}$, there exists a constant β such that $\beta \geq \|\phi_i\|_1, \ \forall i \in \mathcal{I}, \forall t$. The constant β can be determined, at time $t = 0$, using the agents' initial states and the upper bounds on $\|g_i(t)\|$ and $\|\dot{g}_i(t)\|, \forall t$ and $\forall i \in \mathcal{I}$.

Denote the upper bounds on $\|x_i(t)\|, \|g_i(t)\|$ and $\|\dot{g}_i(t)\|, \forall t$ and $\forall i \in \mathcal{I}$ as, respectively, $\beta_x, \bar{g}$ and $\bar{\dot{g}}$. It is easy to see that if there exist constant $\beta_x, \bar{g}$ and $\bar{\dot{g}}$, there exists a constant β, where $\beta \geq \|\phi_i\|_1, \forall i \in \mathcal{I}$, and $\forall t$. We show the argument in three steps.

1. It is assumed that $\|g_i(t)\|$ and $\|\dot{g}_i(t)\|, \forall t$ and $\forall i \in \mathcal{I}$ are bounded. Hence, our proof will be completed if we can show that there exists a constant β_x. In the remaining, we will show that not only there exists a constant β_x, but also it is sufficient to determine this constant using the agents' initial states. Define β_x as

$$\beta_x = \frac{1}{N}\left\|\sum_{i=1}^{N} x_i(0)\right\| + \frac{2}{\sigma}\bar{g} + (N-1)R + \gamma, \tag{.2}$$

where γ is a positive constant and R is defined in Definition 41.

We know that for W defined in (3.64), and for β selected based on the introduced constants $\beta_x, \bar{g}$ and $\bar{\dot{g}}$, we have $\dot{W}_{|t=0} < 0$. We will use a contradiction approach to show that such β ensures $\dot{W} \leq 0$ for all t. Assume that there exists a time $t = t_\epsilon$ at which $\dot{W}$ becomes positive, i.e, $\dot{W}_{|t=t_\epsilon} > 0$. By recalling the selected conservative β, this is only possible if there exists an agent i such that $\|x_i(t_\epsilon)\| > \beta_x$.

Note that that $\dot{W}(t) \leq 0,\ \forall t \in [0, t_\epsilon)$, which means that $V_{ij}(t)$ is bounded, $\forall t \in [0, t_\epsilon)$, which in turn implies that the agents remain connected. Hence, it is easy to see that $\forall\, t \in [0, t_\epsilon)$ we have

$$\|x_i(t) - x_j(t)\| < (N-1)R. \tag{.3}$$

2. Using W defined in (3.14) and similar to Theorem 18, we have $\dot{W}(t) < 0$ (no matter consensus is reached or not), which in turn implies that $\left\|\sum_{j=1}^N \nabla f_j(x_j)\right\|$ is decreasing. Now, for $\nabla f_i(x_i, t) = \sigma x_i + g_i(t)$, defined in Theorem 42, and using the properties of the norms, it is easy to show that

$$\sigma \left\|\sum_{i=1}^N x_i(t)\right\| - \left\|\sum_{i=1}^N g_i(t)\right\| \leq \left\|\sum_{i=1}^N \nabla f_i(x_i(t), t)\right\|$$
$$\leq \left\|\sum_{i=1}^N \nabla f_i(x_i(0), 0)\right\| \leq \sigma \left\|\sum_{i=1}^N x_i(0)\right\| + \left\|\sum_{i=1}^N g_i(0)\right\|.$$

Using the upper bound $\bar{g}$, we have

$$\left\|\sum_{i=1}^N x_i(t)\right\| \leq \left\|\sum_{i=1}^N x_i(0)\right\| + \frac{2N}{\sigma}\bar{g}, \tag{.4}$$

for all t.

3. Now, using (.3) and (.4), it is easy to see that

$$\|x_i(t)\| \leq \frac{1}{N}\left\|\sum_{i=1}^N x_i(0)\right\| + \frac{2}{\sigma}\bar{g} + (N-1)R, \tag{.5}$$

$\forall i \in \mathcal{I}$, and $\forall t \in [0, t_\epsilon)$. Now, under the assumption of $\|x_i(t_\epsilon)\| > \beta_x$ and using the selected β_x in (.2), we have $\|x_i(t_\epsilon)\| - \lim_{t \to t_\epsilon^-} \|x_i(t)\| > \gamma$. However, $\|x_i(t_\epsilon)\| \neq \lim_{t \to t_\epsilon^-} \|x_i(t)\|$, contradicts with the continuity of the agents' positions which is mentioned in Remark 15. Therefore, we have $\|x_i(t)\| \leq \beta_x, \ \forall i \in \mathcal{I}$ and $\forall t$. We thus conclude that there exists no time $t = t_\epsilon$, where $\dot{W}_{|t=t_\epsilon} > 0$ and it completes the proof.