Title
Distributed Tracking Algorithms for Multi-Agent Systems to Solve the Leader-Follower Flocking of Lagrange Networks and Dynamic Average Tracking Problem of Second-Order Systems

Permalink
https://escholarship.org/uc/item/2b746882

Author
Ghapani, Sheida

Publication Date
2016-01-01

License
CC BY 4.0

Peer reviewed|Thesis/dissertation
Distributed Tracking Algorithms for Multi-Agent Systems to Solve the Leader-Follower Flocking of Lagrange Networks and Dynamic Average Tracking Problem of Second-Order Systems

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

Doctor of Philosophy

in

Electrical Engineering

by

Sheida Ghapani

December 2016

Dissertation Committee:

Prof. Wei Ren, Chairperson
Prof. Jay Farrell
Dr. Fabio Pasqualetti
The Dissertation of Sheida Ghapani is approved:

__________________________________________

__________________________________________

__________________________________________

Committee Chairperson

University of California, Riverside
Acknowledgments

Firstly, I would like to express my sincere gratitude to my advisor Prof. Wei Ren for his continuous support of research, for his patience, motivation, enthusiasm, and immense knowledge.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Farrell and Dr. Pasqualetti, for their insightful comments and encouragement, but also for the hard question which incented me to widen my research from various perspectives. Also I thank my friend Salar for the stimulating discussions we had and for his help and support in the last four years.

Last but not the least, I would like to thank my family: my parents, my sister, Leila, and my brothers, Mehrdad, Mohammad and Mehran for supporting me spiritually throughout my Ph.D. and my life in general.
To my parents for their love and support for all these year.
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 solve problems that are difficult or impossible for an individual agent. Coordination in multi-agent systems attracts significant interest in the realm of engineering. Examples of cooperative tasks include mobile sensor networks, automated parallel delivery of payloads, region following formation control and coordinated path planning. One common feature for these systems is acting the agents in a distributed manner (using only local information from their neighbors) to complete global tasks cooperatively so as to increase flexibility and robustness. In this work, two distributed tracking issues in multi-agent systems are investigated in detail: leader-follower flocking with a moving leader for Lagrange networks and distributed average tracking of physical agents.

Flocking of multi-agent systems is the motion of a group of agents cohesively to maintain connectivity and avoid collisions. This dissertation proposes novel distributed tracking algorithms to solve the leader-follower flocking problem with a moving leader. The problem is investigated for networked Lagrange systems with parametric uncertainties under a proximity graph. Two cases
are considered: i) the leader moves with a constant velocity, and ii) the leader moves with a varying velocity. In the first case, a distributed continuous adaptive control algorithm accounting for unknown parameters is proposed in combination with a distributed continuous estimator for each follower. In the second case, a distributed discontinuous adaptive control algorithm and estimator are proposed to track the varying leader, where only a group of followers have access to the leader. However, in the proposed algorithm the agents use the two-hop neighbors’ information and need some global information to determine the control gains. Thus, the algorithm is improved in the next step to use one-hop neighbors’ information and to be fully distributed with the introduction of gain adaptation laws. In all proposed algorithms, flocking is achieved as long as the connectivity and collision avoidance are ensured at the initial time and the control gains are designed properly.

In the distributed average tracking problem, each agent uses local information to calculate the average of individual varying input signals, one per agent. In this dissertation, two distributed average tracking problems for physical second-order agents are investigated. First, distributed average tracking problem is studied for double-integrator agents with reduced requirement on velocity measurements and in the absence of correct position and velocity initialization. Two algorithms are introduced, where in both algorithms a distributed discontinuous control input and a filter are proposed. In the first algorithm, the requirement for either absolute or relative velocity measurements is removed. The algorithm is robust to initialization errors and can deal with a wide class of input signals with bounded deviations in input signals, input velocities, and input accelerations. In the second algorithm, the requirement for communication. However, the algorithm can deal with a smaller group of input signals. Second, distributed average tracking problem of physical second-order agents with heterogeneous nonlinear dynamics is investigated, where there is no constraint on
input signals. The agents’ dynamics satisfy a Lipschitz-like condition that will be defined later and is more general than the Lipschitz-type condition. In the proposed algorithm, a control input and a filter are designed for each agent. Since the input signals are arbitrary and the nonlinear terms in agents’ dynamics can be unbounded, novel state-dependent time varying gains are employed in agents’ filters and control inputs to overcome these unboundedness effects.

The dissertation provides a rigorous stability analysis of the introduced control algorithms for both issues and presents simulations that validate the effectiveness of the proposed algorithms.
# Contents

List of Figures xi

1 Introduction 1
   1.1 Flocking 2
      1.1.1 Overview of Related Works 3
   1.2 Distributed Average Tracking 5
      1.2.1 Overview of Related Works 7
   1.3 Contributions of Dissertation 10
   1.4 Preliminaries 14
      1.4.1 Notations 14
      1.4.2 Lagrange Dynamics 15
      1.4.3 Graph Theory 16
      1.4.4 Nonsmooth Analysis 17

2 Distributed Flocking with a Moving Leader for Lagrange Networks with Parametric Uncertainties 19
   2.1 Problem Statement 20
      2.1.1 Flocking of Lagrange Networks, Where the Leader Moves with a Constant Velocity 21
      2.1.2 Flocking of Lagrange Networks, Where the Leader Moves with a Varying Velocity 26
      2.1.3 Fully Distributed Flocking of Lagrange Networks with One-hop Neighbors’ Information, Where the Leader Has a Varying velocity 32
   2.2 Simulation 37

3 Distributed Average Tracking for Physical Second-Order Linear and Nonlinear Systems 48
   3.1 Distributed Average Tracking for Double-Integrator Multi-Agent Systems with Reduced Requirement on Velocity Measurements 49
      3.1.1 Problem statement 49
      3.1.2 Velocity Free Distributed Average Tracking in the Absence of Correct Initialization 50
| 3.1.3 | Distributed Average Tracking in the Absence of Relative Velocity Information and Communication | 59 |
| 3.1.4 | Simulation | 66 |

| 3.2 | Distributed Average Tracking of Physical Second-order Agents Without Constraint on Input Signals | 70 |
| 3.2.1 | Problem Statement | 71 |
| 3.2.2 | Distributed Average Tracking for Physical Heterogeneous Nonlinear Second-order Agents | 72 |
| 3.2.3 | Distributed Average Tracking for Physical Double-Integrator Agents | 79 |
| 3.2.4 | Simulation | 82 |

| 4 | Conclusions | 85 |

| Bibliography | 87 |
List of Figures

1.1 Three heuristic rules of the flocking of multi-agent systems, [66]. 3
1.2 Example of region following formation control [36]. 6

2.1 Potential function [5]. 23
2.2 The trajectories of the followers and the leader in the first case. The leader is represented as a square while the followers are represented as circles. An edge between two followers denotes that the two are neighbors, and an arrow from the leader to a follower denotes that the leader is a neighbor of the follower. 40
2.3 The velocity errors between the followers and the leader using (2.3)-(2.6). 41
2.4 The trajectories of the followers and the leader in the second case. 42
2.5 The velocity errors between the followers and the leader using (2.16)-(2.19). 43
2.6 The trajectories of the followers and the leader in the second case. 44
2.7 The velocities of the followers and the leader in the second case. 45
2.8 The trajectories of the followers and the leader in the third case. 46
2.9 The velocity errors between the followers and the leader using (2.29)-(2.34). 47

3.1 The positions of 10 agents and the average of the input signals, where the states are initialized correctly. The initial average of input signals is represented as a dot while the initial positions of the agents are represented as squares. 67
3.2 The velocities of 10 agents and the average of input velocities, where the agents’ velocity are initialized correctly. 68
3.3 The positions of 10 agents and the average of input signals in the absence of correct initialization. The initial average of input signals is represented as a dot while the initial positions of the agents are represented as squares. 69
3.4 The velocities of 10 agents and the average of input velocities in the absence of correct initialization. 70
3.5 The positions of 5 agents and the average of the reference inputs. The solid lines and the dashed lines describe, respectively the average of reference inputs and the position of agents. 83
3.6 The velocities of 5 agents and the average of reference inputs. The solid lines and the dashed lines describe, respectively the average of reference velocities and the velocity of agents. 84
Chapter 1

Introduction

The objective of this dissertation is the study of collective behaviors in multi-agent systems. A multi-agent system is a system or network of agents, where each agent has an associated dynamical equation describing its behavior in relation to itself, a subset of the other agents in the network and the environment. 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 (decentralized) approach. In the centralized approach, a powerful central station collects all of the information from other 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. Compared to the centralized approach, the distributed approach brings a number of benefits, such as increased robustness, efficiency and flexibility as well as easy maintenance, to many engineering systems. However, the inherently distributed nature of these systems makes the design of effective algorithms very challenging as the global performance depends significantly on the complex interactions among the agents. Due to the cooperation among
agents, control of distributed multi-agent coordination may also be referred to as cooperative control.

A basic behavior in cooperative control is consensus. Consensus is the agreement on certain quantities, such as position and velocity, by a group of agents. Some cooperative tasks in which consensus algorithms are employed include formation control [25, 57, 62], flocking theory [56, 77], rendezvous [30, 73], and distributed average tracking problems [28, 72]. For example, in the formation control, all agents maintain their relative positions to one another (consensus on relative position), thus maintaining the desired formation configuration. In this dissertation, we address the following subjects which are listed in the order appearance

1. Leader-follower flocking with a moving leader for Lagrange networks,

2. Distributed average tracking for physical second-order multi-agent systems with linear and nonlinear dynamics.

Thus, we continue with a short introduction and overview of flocking and distributed average tracking problems.

1.1 Flocking

Flocking is a collective behavior of large number of interacting agents with a common group objective. This collective behavior can be observed in nature like flock of birds, swarm of insects, and school of fish. The key idea behind flocking relies on substituting a complex and expensive single agent such as a robot with a more affordable and flexible set of agents. In 1987, Reynolds characterized the flocking of multi-agent systems by describing the following three heuristic rules:
Figure 1.1: Three heuristic rules of the flocking of multi-agent systems, [66].

(1) Cohesion: attempt to stay close to nearby flockmates; (2) Separation: avoid collisions with nearby flockmates; and (3) Alignment: attempt to match velocity with nearby flockmates, [66]. In 1995, Vicsek proposed a flocking model in which mobile agents regulate their headings to match the average of the headings of their nearest neighbors. The flocking of multi-agent systems has engineering applications such as massive mobile sensing in an environment; parallel and simultaneous transportation of vehicles or delivery of payloads; and performing military missions such as reconnaissance, surveillance, and combat using a cooperative group of Unmanned Aerial Vehicles (UAVs). In the reminder of this subsection, we talk about the works that have been done on flocking of multi-agent systems.

1.1.1 Overview of Related Works

The focus of many flocking-related research works has been on multi-agent systems in which the agents are governed by linear dynamics. For example in [78], a flocking algorithm is introduced for a group of agents, where there is no leader. However, in some applications there is an objective of interest for the whole group that can be represented by a real or virtual leader. Therefore, [56] studies the flocking problem with a leader with constant velocity, where all of the followers have access to the leader. Ref. [74] considers a more general condition, where the leader
has a constant and a varying velocity. When the leader has a constant velocity, [74] relaxes the constraint that the leader is a neighbor of all followers. However, if the leader has a varying velocity, it still requires that the leader be a neighbor of all followers. Unfortunately, this is an unrealistic restriction on the distributed control design, especially when the number of the followers becomes large. In [5], distributed control algorithms for swarm tracking are studied via a variable structure approach, where the moving leader is a neighbor of only a subset of the followers. In [44], the flocking control and communication optimization problem are considered for multi-agent systems in a realistic communication environment and the desired separation distances between neighboring agents are calculated in real time.

Note that all above references focus on linear multi-agent systems with single- or double-integrator dynamics. However, in reality, many physical systems are inherently nonlinear and cannot be described by linear equations. Among the nonlinear systems, Lagrange models can be used to describe a large class of physical systems of practical interests such as autonomous vehicles, walking robots, and rotation and translation of spacecraft formation flying. But due to the existence of nonlinear terms with parametric uncertainties, the algorithms for linear models cannot be directly used to solve the coordination problem for multi-agent systems with Lagrange dynamics.

Recent results on distributed coordination of networked Lagrange systems focus on the consensus without a leader [16, 37, 51, 61, 80, 82], coordinated tracking with one leader [20, 22, 47], containment control with multiple leaders [46, 48, 50], and flocking or swarming without or with a leader [8, 17, 49]. Ref. [17] proposes a control algorithm based on potential functions for networked Lagrange systems to achieve collision avoidance and velocity matching simultaneously in both time-delay and switching-topology scenarios. However, parametric uncertainties are not con-
sidered and there is no leader. Ref. [8] presents a region-based shape controller for a swarm of Lagrange systems. By utilizing potential functions, the authors design a control scheme that can force multiple robots to move as a group inside a desired region with a common velocity while maintaining a minimum distance among themselves. However, the algorithm relies on the strict assumption that all followers have access to the information of the desired region and the common velocity. A leader-follower swarm tracking framework is established in [49] in the presence of multiple leaders. However, only a compromised result can be obtained, where the group dispersion, cohesion, and containment objectives are considered together. In the proposed algorithms, the variables of the estimators must be communicated among the followers. Furthermore, more information is used in the controller design, for example, the second-order derivatives of the potential functions.

In our introduced algorithm, a group of followers move cohesively with a moving leader to maintain connectivity and avoid collisions for whole time and also eventually achieve velocity matching. The leader can be a physical or virtual vehicle, which encapsulates the group trajectory. We consider two cases: i) the leader moves with a constant velocity, and ii) the leader moves with a varying velocity. In the first case, a distributed continuous adaptive control algorithm accounting for unknown parameters and a distributed continuous estimator are proposed for each follower. In the second case, we propose distributed discontinuous adaptive control algorithms to drive the agents to track the varying leader, where only a group of agents have access to the leader.

1.2 Distributed Average Tracking

The idea of distributed average tracking is that multiple agents track the average of multiple varying input signals, one input signal per each agent, under local interactions. The problem has
found applications in distributed sensor fusion [58], feature-based map merging [2], and distributed Kalman filtering [3], where the scheme has been mainly used as an estimator. However, there are some applications such as region following formation control [8] or coordinated path planning [76] that require the agents’ physical states instead of estimator states to converge to a varying network quantity, where each agent only has a local and incomplete copy of that quantity. For example, in region-following formation control each agent has the knowledge of a sub-area $S_i(t) \subseteq S(t)$, where $S(t) \in \mathbb{R}^p$ is a dynamic region that the agents should move into. Let $p_i(t) \in S_i(t)$ be the varying center of $S_i(t)$, then $\frac{1}{n} \sum_{j=1}^{n} p_j(t)$ can be viewed as the center of the dynamic region (Fig. 1.2). Therefore, the distributed average tracking algorithm can be employed to drive $y_i(t)$ to $\frac{1}{n} \sum_{j=1}^{n} p_j(t)$ for the $i$th agent, where $y_i(t) = x_i(t) - \delta_i$, $x_i$ is the $i$th agent’s position and $\delta_i$ is a constant vector used by the $i$th agent to specify the formation of the agents. Compared with the consensus problem, distributed average tracking poses more theoretical challenges, since the tracking objective is varying and is not available to any agent. In the reminder of this subsection, we talk about the works that have been done on distributed average tracking problem.
1.2.1 Overview of Related Works

In the literature, linear distributed algorithms have been employed for special types of varying input signals. Ref. [72] uses frequency domain analysis to study consensus on the average of multiple input signals with steady-state values. In [28], a proportional algorithm and a proportional-integral algorithm are proposed to achieve distributed average tracking with a bounded tracking error, where accurate estimator initialization is relaxed in the proportional-integral algorithm. In [4], the internal model principle is employed to extend the proportional-integral algorithm to a special group of varying input signals with a common denominator in their Laplace transforms, where the denominator also needs to be used in the estimator design. In [89] and [52], two discrete-time distributed average tracking algorithms are addressed, where [52] extends the proposed algorithm in [89] by introducing a varying sequence of damping factors to achieve robustness to initialization errors. In [43], the authors propose first- and second-order-input consensus algorithms to allow the agents to track the average of their dynamic input signals with a pre-specified rate, where the interaction is described by a strongly connected and weight-balanced directed graph.

However, linear algorithms cannot ensure distributed average tracking for general input signals. Therefore, some researchers employ nonlinear tracking algorithms. In [53], a class of nonlinear algorithms is proposed for input signals with bounded deviations, where the tracking error is proved to be bounded. A nonsmooth algorithm is proposed in [9], which is able to track arbitrary varying input signals with bounded derivatives. All the above references primarily study the distributed average tracking problem from a distributed estimation perspective, where the agents implement local estimators through communication with neighbors freely without the need for obeying certain physical agent dynamics. However, there are applications where the distributed average
tracking problem is relevant for designing distributed control laws for physical agents. One application is the region-following formation control [8], where a swarm of robots is required to move inside a dynamic region while keeping a desired formation. Here, the dynamics of the physical agents must also be taken into account in the control law design and the dynamics themselves introduce further challenges to the distributed average tracking problem. For example, the control law designed for physical agents with single-integrator dynamics can no longer be used directly for physical agents subject to more complicated dynamic equations. Distributed average tracking for physical agents with double-integrator dynamics is studied in [14], where the input signals are allowed to have bounded accelerations. Distributed average tracking for physical agents with general linear dynamics is addressed in [87] and [12]. Ref. [12] proposes a discontinuous algorithm, while a continuous algorithm is employed in [87] with, respectively, static and adaptive coupling strengths.

It is noted that in [14] both relative position and relative velocity measurements are required in the control laws designed for double-integrator agents. While double-integrator dynamics can be viewed as a special case of general linear dynamics, the distributed average tracking algorithms in [11], when applied to double-integrator systems, still need relative position and velocity measurements. In practice, velocity measurements are usually less accurate and more expensive than position measurements. In addition, relative velocity measurements are often more challenging and expensive than absolute velocity measurements. We are hence motivated to solve the distributed average tracking problem for physical double-integrator agents with reduced requirement on velocity measurements. Two distributed algorithms (controller design combined with filter design) are introduced to achieve distributed average tracking with reduced requirement on velocity measurements. In the context of distributed average tracking, reducing velocity measurements poses
significant theoretical challenges. The reason is that unlike the consensus on single-leader coordinated tracking problems, there are significant additional inherent challenges in distributed average tracking as none of the agents has the tracking objective available. It will be explained that each algorithm has its own relative benefits and is feasible for different application scenarios.

However, in real applications, the agents have more complicated dynamics rather than single- or double-integrator dynamics. In [11] a proportional-integral control scheme is extended to achieve distributed average tracking for physical Euler-Lagrange systems for two different types of input signals with steady states and with bounded derivatives. It should be noted that the introduced algorithms in [11] work for a limited group of input signals. Thus, in next step, we introduced a distributed average distributed algorithm for physical second-order agents with heterogeneous nonlinear dynamics, where there is no constraint on the input signals. The nonlinear terms in agents’ dynamics satisfy a Lipschitz-like condition that will be defined later and is more general than the Lipschitz-type condition. Since the unknown terms in agents’ dynamics can be unbounded and the input signals are arbitrary, we are faced with extra challenges. Therefore, a local filter is introduced for each agent to estimate the average of the input signals and reference velocities, where time varying state-dependent gains are employed in each agent’s filter and control input to overcome these unboundedness effects. In the special case, where the agents have double-integrator dynamics, the filter is not required anymore. An algorithm is proposed, where the agents’ positions and velocities are driven directly to track the average of the input signals and the reference velocities. Here, by employing the novel time varying state-dependent gains in the control input, the distributed average tracking is achieved in the presence of arbitrary input signals.
1.3 Contributions of Dissertation

In this dissertation, we focus on the following two tracking problems:

1. Leader-follower flocking with a moving leader for Lagrange networks,

2. Distributed average tracking for second-order linear and nonlinear multi-agent systems.

Some materials from this dissertation have been published in three conferences [31, 33, 34] and the journal [32]. In this subsection, we briefly talk about the contributions of this dissertation as follow.

- In the first part of the dissertation, we focus on the distributed leader-follower flocking problem with a moving leader for networked Lagrange systems with unknown parameters under a proximity graph defined according to the relative distance between each pair of agents. Here a group of followers move cohesively with the moving leader to maintain connectivity and avoid collisions for all time and also eventually achieve velocity matching. The leader can be a physical or virtual vehicle, which encapsulates the group trajectory. We consider two cases: i) the leader moves with a constant velocity, and ii) the leader moves with a varying velocity. In the first case, a distributed continuous adaptive control algorithm accounting for unknown parameters and a distributed continuous estimator are proposed for each follower. In the second case, we first propose a distributed discontinuous adaptive control algorithm and estimator, where we use two-hop neighbors’ information and common control gain for all followers. Hence the system is not completely distributed. We then improve the algorithm to use one-hop neighbors’ information. Further, by proposing gain adaption schemes, the algorithm is improved as a fully distributed algorithm. In all proposed algorithms, flocking is achieved as long as the connectivity and collision avoidance are ensured at the initial time.
and the control gains are designed properly.

Compared with the results in the existing literature, this paper has the following novel features.

1) Each agent is considered as a nonlinear Lagrange system with parametric uncertainties which is more realistic. While in [5, 44, 56, 78], the agents’ dynamics are assumed to be single- or double-integrators. The results for single- or double-integrator dynamics are not applicable to Lagrange systems with parametric uncertainties.

2) This dissertation considers the combination of flocking (considering connectivity maintenance, collision avoidance, and velocity matching with a moving leader in the meantime) and the constraint that the leader’s information is available to only the followers in its proximity. The above constraint introduces further complexities since not all followers know the leader’s velocity. Even for the case with single- or double-integrator agents, the problem is very challenging [5], not to mention the case of nonlinear Lagrange systems with parametric uncertainties. In contrast, in [17], parametric uncertainties are not considered and there is no leader and in [8], it is assumed that the leader’s information is available to all followers (against the local interaction nature of the problem).

3) To overcome the coexistence and coupling of the above mentioned challenges, in this dissertation, we propose an adaptive control law in combination with a new distributed estimator for each follower. The novelty of the estimators is that the partial derivatives of the potential functions are integrated into the estimators. In [5, 49], the variables of the estimators must be communicated between the neighbors. For the case of a moving leader with varying velocity, the proposed algorithms in [5, 47] require both one-hop
and two-hop neighbors’ information. In contrast, in the second proposed algorithm for tracking the leader with varying velocity, only one-hop neighbors’ information (e.g., the relative position and velocity measurements between the neighbors and the absolute position and velocity measurements) is required. These measurements can be obtained by the sensing devices carried by the agents and hence the need for communication can be removed. Further, this algorithm is fully distributed and global information is not required, while the results in [5,47,49] rely on some global information.

• In the second part of dissertation, two groups of distributed average tracking algorithms are introduced for, respectively, second-order linear and nonlinear multi-agent systems.

First, two distributed algorithms (controller design combined with filter design) are introduced to achieve distributed average tracking with reduced requirement on velocity measurements and in the absence of correct position and velocity initialization. Each algorithm has its own relative benefits and is feasible for different application scenarios. In the first algorithm design, there is no need for either absolute or relative velocity measurements. Each agent’s algorithm employs its local relative positions with respect to neighbors, its own and neighbors’ filter outputs accessed through communication and its own input signal, input velocity and input acceleration. The algorithm allows the agents to track the average of a wide class of time-varying input signals with bounded deviations among the input signals, among the input velocities, and among the input accelerations. Using this algorithm, distributed average tracking can be achieved in the absence of velocity measurements and correct initialization.

In addition, it will be shown that if the agents can be correctly initialized, the algorithm can be modified to achieve the distributed average tracking with even a larger class of input sig-
nals with only bounded input accelerations’ deviations. In the second algorithm design, there is still no requirement for correct initialization and relative velocity measurements. Furthermore, inter-agent communication is not necessary and the algorithm can be implemented using only local sensing, which is desirable for certain applications (e.g., deep-space spacecraft formation flying), where communication might not be desirable or available. Each agent’s algorithm only employs its local relative positions with respect to neighbors, its own velocity, input signal, input velocity and input acceleration. In this algorithm, the input signals, input velocities and input accelerations should be all bounded.

Second, a distributed algorithm is introduced to achieve distributed average tracking for physical second-order agents with heterogeneous nonlinear dynamics, where there is no constraint on the input signals. Here, the nonlinear terms in agents’ dynamics satisfy a Lipschitz-like condition that will be defined later and is more general than the Lipschitz-type condition. Since the unknown terms in agents’ dynamics can be unbounded and the input signals are arbitrary, we are faced with extra challenges and hence a local filter is introduced for each agent to estimate the average of the input signals and reference velocities. The novelty of the local filters is that by employing the time varying state-dependent gains, the agents can track the average of a group of input signals with no constraint on their dynamics. In the special case, where the agents have double-integrator dynamics, the filter is not required anymore. Therefore, the algorithm is modified, to drive the agents’ positions and velocities directly to track the average of the input signals and the reference velocities.
1.4 Preliminaries

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

1.4.1 Notations

\( \mathbb{R} \) set of real numbers

\( \mathbb{R}^p \) set of \( p \times 1 \) real vectors

\( \mathbb{R}^{m \times n} \) set of \( m \times n \) real matrices

1\(_n \) \( n \times 1 \) column vector of all ones

0\(_n \) \( n \times 1 \) column vector of all zeros

\( \lambda_{\min}(\cdot) \) 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

\( \| \cdot \| \) Euclidean norm

\( \| \cdot \|_p \) \( p \)-norm of a real vector
Kronecker product

\[ \text{sgn}(\cdot) \]

signum function defined component-wise

\[ f(t) \in \mathbb{L}_l \quad \text{if for a vector function } f(t) : \mathbb{R} \mapsto \mathbb{R}^m, \left( \int_0^\infty \| f(\tau) \| d\tau \right)^{\frac{1}{2}} < \infty \]

\[ f(t) \in \mathbb{L}_\infty \quad \text{if for each element of } f(t), \text{noted as } f_i(t), \sup_{t \geq 0} |f_i(t)| < \infty, i = 1, \ldots, m \]

1.4.2 Lagrange Dynamics

Suppose that \( n \) followers are described by Lagrange equations of the form [40]

\[ M_i(q_i) \ddot{q}_i + C_i(q_i, \dot{q}_i) \dot{q}_i + g_i(q_i) = u_i, \quad i = 1, \ldots, n, \tag{1.1} \]

where \( q_i \in \mathbb{R}^p \) is the vector of generalized coordinates\(^1\), \( M_i(q_i) \) is the \( p \times p \) symmetric inertia matrix, \( C_i(q_i, \dot{q}_i) \dot{q}_i \) is the Coriolis and centrifugal force, \( g_i(q_i) \) is the vector of gravitational force, and \( u_i \) is the control input. The dynamics of the Lagrange systems satisfy the following properties:

(P1) There exist positive constants \( k_M, k_{\overline{M}}, k_C, k_G \) such that \( k_M I_p \leq M_i(q_i) \leq k_{\overline{M}} I_p, \| C_i(q_i, \dot{q}_i) \| \leq k_C \| \dot{q}_i \| \) and \( \| g_i(q_i) \| \leq k_G \).

(P2) \( \dot{M}_i(q_i) - 2C_i(q_i, \dot{q}_i) \) is skew symmetric.

(P3) The left-hand side of the Lagrange dynamics can be parameterized, i.e., \( M_i(q_i)x + C_i(q_i, \dot{q}_i)y + g_i(q_i) = Y_i(q_i, \dot{q}_i, x, y)\theta_i, \forall x, y \in \mathbb{R}^p \), where \( Y_i \in \mathbb{R}^{p \times p_\theta} \) is the regression matrix and \( \theta_i \in \mathbb{R}^{p_\theta} \) is the unknown but constant parameter vector.

In this dissertation, there exist \( n + 1 \) agents (e.g., autonomous vehicles) consisting of one leader and \( n \) followers. The leader is labeled as agent 0 and the followers are labeled as agent 1 to

\(^{1}\)In the context of autonomous vehicles, \( q_i \) denotes the position of agent \( i \).
Note that the leader can be a physical or virtual vehicle, which encapsulates the group trajectory. The leader’s position and velocity are denoted by, respectively, \( q_0 \in \mathbb{R}^p \) and \( \dot{q}_0 \in \mathbb{R}^p \).

### 1.4.3 Graph Theory

With \( k \) agents in a team, a graph is used to characterize the interaction topology among the agents. A graph is a pair \( G = (V, E) \), where \( V = \{1, \ldots, k\} \) is the node set and \( E \subseteq V \times V \) is the edge set. In a directed graph, an edge \( (j, i) \in E \) means that node \( i \) can obtain information from node \( j \) but not necessarily vice versa. Here node \( j \) is a neighbor of node \( i \). In an undirected graph \( (i, j) \in E \Leftrightarrow (j, i) \in E \). A directed path in a directed graph is an ordered sequence of edges of the form \( (i_1, i_2), (i_2, i_3), \ldots \), where \( i_j \in V \). A subgraph of \( G \) is a graph whose node set and edge set are subsets of those of \( G \). \( N_i = \{j|(j, i) \in E\} \) denotes the set of neighbors of agent \( i \).

The adjacency matrix \( A = [a_{ij}] \in \mathbb{R}^{k \times k} \) of the graph \( G \) is defined such that the edge weight \( a_{ij} = 1 \) if \( (j, i) \in E \) and \( a_{ij} = 0 \) otherwise. For an undirected graph, \( a_{ij} = a_{ji} \). The Laplacian matrix \( L = [l_{ij}] \in \mathbb{R}^{k \times k} \) associated with \( A \) is defined as \( l_{ii} = \sum_{j \neq i} a_{ij} \) and \( l_{ij} = -a_{ij} \), where \( i \neq j \). For an undirected graph, \( L \) is symmetric positive semi-definite [18]. To simplify our analysis, we assign an orientation to an edge by considering one node the positive end of the edge and the other node the negative end of the edge. We recall that the \( n \times N \) incidence matrix \( D = [d_{ik}] \in \mathbb{R}^{n \times N} \) of a graph is defined as [67]

\[
    d_{ik} = \begin{cases} 
    +1 & \text{if node } i \text{ is the positive end of the edge } E_k, \\
    -1 & \text{if node } i \text{ is the negative end of the edge } E_k, \\
    0 & \text{otherwise,}
    \end{cases}
\]

where \( E_k \in E \) is the \( k \)th edge. Then the Laplacian matrix of the graph can be denoted by \( L = DD^T \).
1.4.4 Nonsmooth Analysis

Consider a vector-valued differential equation

\[ \dot{x}(t) = f(t, x(t)), \] (1.2)

where \( t \in \mathbb{R} \) and \( x(t) \in \mathbb{R}^n \).

**Definition 1** For the differential equation (1.2), define the Filippov set-valued map

\[ K[f](t, x(t)) \triangleq \bigcap_{\delta > 0} \bigcap_{u(N) = 0} \text{co} \left( f(t, B(x(t), \delta) - N) \right), \]

where co denotes the convex closure, \( \bigcap_{u(N) = 0} \) denotes the intersection over all sets of Lebesgue measure zeros and \( B(x, \delta) \) is the open ball of radius \( \delta \) centered at \( x \) [27].

**Definition 2** Replace the differential equation (1.2) by the differential inclusion

\[ \dot{x}(t) \in K[f](t, x(t)). \] (1.3)

A vector function \( x(\cdot) : \mathbb{R} \to \mathbb{R}^n \) is called a Filippov solution of (1.2) on \([t_0, t_1], t_1 \leq \infty\), if \( x(\cdot) \) is absolutely continuous and satisfies (1.3) for almost all \( t \in [t_0, t_1] \).

**Lemma 3** Suppose that \( f(t, x(t)) \) in (1.2) is measurable and locally essentially bounded, that is, bounded on a bounded neighborhood of every point excluding sets of measure zero. Then, for any \( x(0) \in \mathbb{R}^n \), there exists a Filippov solution of (1.2) with initial condition \( x(0) = x_0 \) [27].

Let \( W[\cdot] : \mathbb{R}^n \to \mathbb{R} \) be a locally Lipschitz function of \( x(t) \). The generalized gradient of \( W[\cdot] \) is defined

\[ \partial \tilde{W} \triangleq \text{co} \left\{ \lim_{w \to x} \nabla W(w) : w \in \mathbb{R}^n, w \notin \Omega_W \cup S \right\}, \]
where \( co(\cdot) \) is the convex hull, \( \Omega_W \) is the set of points in which \( W[x(t)] \) is not differentiable and \( S \) is a set of measure zero that can be arbitrarily chosen so as to simplify the calculation. The set-valued Lie derivative of \( W[x(t)] \) with respect to \( x(t) \), the trajectory of (1.2), is defined as 
\[
\dot{\tilde{W}} \triangleq \cap_{\zeta \in \partial \tilde{W}} \zeta^T \mathcal{K}[f].
\]

Chapter 2

Distributed Flocking with a Moving Leader for Lagrange Networks with Parametric Uncertainties

In this chapter, the leader-follower flocking problem with a moving leader is investigated. The problem is investigated for networked Lagrange systems with parametric uncertainties under a proximity graph. Two cases are considered: i) the leader moves with a constant velocity, and ii) the leader moves with a varying velocity. In both cases, a distributed adaptive control algorithm accounting for unknown parameters is proposed in combination with a distributed estimator for each follower.
2.1 Problem Statement

In this section, the goal is to design $u_i$ for each follower to achieve the leader-follower flocking. That is, the followers move cohesively with the leader (connectivity maintenance) and avoid collisions for all time and eventually achieve velocity matching with the leader ($\|\dot{q}_i(t) - \dot{q}_0(t)\| \to 0$) in the presence of unknown parameters under only local interaction defined by the proximity graph. Before moving on, the following auxiliary variables are defined:

$$s_i = \dot{q}_i - v_i, \quad \tilde{q}_i = q_i - q_0, \quad \tilde{v}_i = v_i - \dot{q}_0,$$

(2.1)

where $v_i$ is agent $i$’s estimate of the leader’s velocity to be designed later. Note that

$$s_i = \dot{\tilde{q}}_i - \tilde{v}_i.$$

(2.2)

In this chapter, we assume that the neighbor relationship among the leader and the followers is based on their relative distance and hence the graph characterizing the interaction topology is a proximity graph. We also assume that the leader has no neighbor and its motion is not necessarily dependent on the followers. In particular, followers $i$ and $j$ are neighbors of each other if $\|q_i - q_j\| < R$ and the leader is a neighbor of follower $i$ if $\|q_i - q_0\| < R$, where $R$ denotes the sensing radius of the agents. Let $G_F$ be the proximity graph characterizing the interaction among the $n$ followers with the associated Laplacian matrix $L_F$. Note that by definition $G_F$ is undirected and hence $L_F$ is symmetric positive semi-definite. Then the Laplacian matrix of the graph can be denoted by $L_F = D_F D_F^T$.

Let $G$ be the directed graph characterizing the interaction among the leader and the $n$ followers corresponding to $G_F$. Also let the edge weight $a_{i0} = 1$ if the leader is a neighbor of follower $i$ and $a_{i0} = 0$ otherwise. Define $\Lambda \triangleq \text{diag}(a_{10}, \ldots, a_{n0})$. Note that $\Lambda^2 = \Lambda$ because $a_{i0}$
is either 1 or 0. Also define the leader-follower topology matrix associated with the graph $G$ as $H = L_F + \Lambda$. It is obvious that $H$ is symmetric positive semi-definite. Before moving on, we need the following lemmas.

**Lemma 4** [64] If the leader has directed paths to all followers, the matrix $H$ is symmetric positive definite.

**Lemma 5** Let $H^a$ and $H^b$ be the leader-follower topology matrix associated with, respectively the graph $G^a$ and $G^b$. If $G^a$ is a subgraph of $G^b$, then $H^a \leq H^b$.

*Proof:* When $G^a$ is a subgraph of $G^b$, $H^b$ can be written as $H^b = H^a + P$, where $P$ is a positive semi-definite matrix. Therefore, it can be concluded that $H^a \leq H^b$.

### 2.1.1 Flocking of Lagrange Networks, Where the Leader Moves with a Constant Velocity

In this subsection, we consider the case, where the leader has a constant velocity. We propose the following distributed control algorithm

\[
\begin{align*}
    u_i &= \hat{u}_i + Y_i(q_i, \dot{q}_i, \dot{v}_i, v_i)\hat{\theta}_i, \quad (2.3) \\
    \dot{\hat{u}}_i &= -\sum_{j=0}^{n} \frac{\partial V_{ij}}{\partial q_i} - \gamma \sum_{j=0}^{n} a_{ij}(t)(\dot{q}_i - \dot{q}_j), \quad (2.4) \\
    \dot{\hat{v}}_i &= -\sum_{j=0}^{n} \frac{\partial V_{ij}}{\partial q_i} - \gamma \sum_{j=0}^{n} a_{ij}(t)(\dot{q}_i - \dot{q}_j), \quad (2.5) \\
    \dot{\hat{\theta}}_i &= -\Gamma_i Y_i^T(q_i, \dot{q}_i, \dot{v}_i, v_i)s_i, \quad (2.6)
\end{align*}
\]

where $V_{ij}$ is the potential function between agents $i$ and $j$ to be designed, $\hat{\theta}_i$ is the estimate of the unknown but constant parameter $\theta_i$, $s_i$ is defined in (2.1), $\gamma$ is a positive constant, and $\Gamma_i$ is a symmetric positive-definite matrix representing the adaptation gain.
Remark 6 Here $v_i$ is the reference velocity, which introduces the partial derivatives of the potential functions in the estimators and it is a key to our problem. It is worthy mentioning that (2.5) has a similar form of the reference velocity derivative proposed in [81], where the partial derivatives are replaced by the position synchronization term. Compared to the position and velocity synchronization problem considered in [81], here we study the flocking problem (connectivity maintenance, collision avoidance, and velocity matching with a moving leader whose information is available to only the followers in its proximity).

The potential function $V_{ij}$ is defined as follows (see [5])

1. When $\|q_i(0) - q_j(0)\| \geq R$, $V_{ij}$ is a differentiable nonnegative function of $\|q_i - q_j\|$ satisfying the conditions (Fig. 2.1.a)

   i) $V_{ij} = V_{ji}$ achieves its unique minimum when $\|q_i - q_j\|$ is equal to the value $d_{ij}$, where $\overline{d}_{ij} < R$.

   ii) $V_{ij} \to \infty$ as $\|q_i - q_j\| \to 0$.

   iii) $\frac{\partial V_{ij}}{\partial \|q_i - q_j\|} = 0$ if $\|q_i - q_j\| \geq R$.

   iv) $V_{ii} = c, i = 1, \ldots, n$, where $c$ is a positive constant.

2. When $\|q_i(0) - q_j(0)\| < R$, $V_{ij}$ is defined as above except that condition iii) is replaced with the condition that $V_{ij} \to \infty$ as $\|q_i - q_j\| \to R$ (Fig. 2.1.b).

The motivation of $V_{ij}$ is to maintain the initial connectivity pattern and to avoid collision.

In the control algorithm (2.3)-(2.6), the term $- \sum_{j=0}^{n} \frac{\partial V_{ij}}{\partial q_i}$ is used for collision avoidance and connectivity maintenance while the term $- \sum_{j=0}^{n} a_{ij}(t)(\dot{q}_i - \dot{q}_j)$ is used for velocity matching.
The control algorithm (2.3)-(2.6) is distributed in the sense that each agent uses only its own position and velocity and the relative position and relative velocity between itself and its neighbors.

**Theorem 7** Suppose that at the initial time $t = 0$, the leader has directed paths to all followers and there is no collision among the agents. Using (2.3)-(2.6) for (1.1), the leader-follower flocking is achieved.

**Proof**: By using the property (P3) of the Lagrange dynamics (1.1), it follows that $M_i(q_i) \dot{v}_i + C_i(q_i, \dot{q}_i)v_i + g_i(q_i) = Y_i(q_i, \dot{q}_i, \dot{v}_i, v_i)\theta_i$. Then using (1.1), (2.1) and (2.3), we have the following closed-loop dynamics

$$M_i(q_i) \dot{s}_i + C_i(q_i, \dot{q}_i)s_i = \ddot{u}_i - Y_i(q_i, \dot{q}_i, \dot{v}_i, v_i)\tilde{\theta}_i,$$

(2.7)
where \(\tilde{\theta}_i = \theta_i - \hat{\theta}_i\). We first define the following non-negative function, which is a common Lyapunov function candidate used in the literature [48, 54, 80, 82] with different definition of \(s_i\)

\[
V_1 = \frac{1}{2} \sum_{i=1}^{n} s_i^T M_i(q_i) s_i + \frac{1}{2} \sum_{i=1}^{n} \tilde{\theta}_i^T \Gamma_i^{-1} \tilde{\theta}_i. \tag{2.8}
\]

The derivative of \(V_1\) is given as

\[
\dot{V}_1 = \sum_{i=1}^{n} [s_i^T M_i(q_i) \dot{s}_i + \frac{1}{2} s_i^T M_i(q_i) s_i - \tilde{\theta}_i^T \Gamma_i^{-1} \dot{\tilde{\theta}}_i] = \sum_{i=1}^{n} s_i^T \dot{u}_i, \tag{2.9}
\]

where we have used the property (P2) and (2.6) to obtain the second equality. To maintain the initial connectivity pattern and to avoid collision, we then define the following non-negative function by the combination of the potential functions

\[
V_2 = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} V_{ij} + \sum_{i=1}^{n} V_{i0}. \tag{2.10}
\]

The derivative of \(V_2\) can be written as

\[
\dot{V}_2 = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (\dot{q}_i^T \frac{\partial V_{ij}}{\partial q_i} + \dot{q}_j^T \frac{\partial V_{ij}}{\partial q_j}) + \sum_{i=1}^{n} (\dot{q}_i^T \frac{\partial V_{i0}}{\partial q_i} + \dot{q}_0^T \frac{\partial V_{i0}}{\partial q_i})
\]

\[
= \sum_{i=1}^{n} \sum_{j=1}^{n} (\dot{q}_i - \dot{q}_0)^T \frac{\partial V_{ij}}{\partial q_i} + \sum_{i=1}^{n} \dot{q}_i^T \frac{\partial V_{i0}}{\partial q_i} = \sum_{i=1}^{n} \dot{q}_i^T \frac{\partial V_{i}}{\partial q_i},
\]

where we have used Lemma 3.1 in [5] and the fact that \(\frac{\partial V_{ij}}{\partial q_i} = -\frac{\partial V_{ij}}{\partial q_i}\) to obtain the second equality, and have used the fact that \(\sum_{i=1}^{n} \sum_{j=1}^{n} \dot{q}_0^T \frac{\partial V_{ij}}{\partial q_i} = \dot{q}_0^T \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial V_{ij}}{\partial q_i} = 0\) to obtain the third equality.

Now consider the following Lyapunov function candidate

\[
V = V_1 + \frac{1}{2} \sum_{i=1}^{n} \dot{v}_i^T \dot{v}_i + V_2. \tag{2.11}
\]
Then the derivative of $V$ is given as

$$
\dot{V} = \sum_{i=1}^{n} s_i^T \dot{u}_i + \sum_{i=1}^{n} \dot{v}_i^T \ddot{v}_i + \sum_{i=1}^{n} \sum_{j=0}^{n} \dot{q}_i^T V_{ij} \frac{\partial V_{ij}}{\partial q_i}.
$$

Since the leader’s velocity $\dot{q}_0$ is constant, we have $\dot{\tilde{v}}_i = \tilde{v}_i = \hat{u}_i$ according to (2.4) and (2.5). It follows that

$$
\dot{V} = \sum_{i=1}^{n} \dot{q}_i^T \hat{u}_i + \sum_{i=1}^{n} \sum_{j=0}^{n} \dot{q}_i^T \frac{\partial V_{ij}}{\partial q_i} = - \sum_{i=1}^{n} \sum_{j=0}^{n} \gamma a_{ij}(t) \dot{q}_i^T (\hat{q}_i - \hat{q}_j),
$$

(2.12)

where we have used (2.2) to obtain the first equality and have used (2.4) and $\dot{q}_i - \dot{q}_j = \dot{\tilde{q}}_i - \dot{\tilde{q}}_j$ to obtain the second equality. Eq. (2.12) can be written in a compact form as

$$
\dot{V} = -\gamma \tilde{q}^T [H(t) \otimes I_p] \tilde{q},
$$

(2.13)

where $\tilde{q}$ is a column stack vector of $\tilde{q}_i$, $i = 1, \ldots, n$, and $H(t)$ is the leader-follower topology matrix at time $t$ defined in Sub-section 1.4.3. Note that $H(t)$ is symmetric positive semi-definite. It follows that $\dot{V}$ is negative semi-definite. Therefore, from $V \geq 0$ and $\dot{V} \leq 0$, it can be concluded that $V$ is bounded and thus $s_i, \tilde{\theta}_i, \tilde{v}_i, V_{ij} \in \mathbb{L}_\infty$. Since $V_{ij}$ is bounded, it is guaranteed that there is no collision and no edge in the graph $\bar{G}(0)$ will be lost. In other words, for any pair of agents $i, j$, there exist positive constants $0 < R_{\min} \leq R_{\max} < R$, such that

$$
\|q_i(t) - q_j(t)\| \in [R_{\min}, R_{\max}], \quad \text{if } \|q_i(0) - q_j(0)\| < R,
$$

$$
\|q_i(t) - q_j(t)\| \in [R_{\min}, (n-1)R_{\max}], \quad \text{otherwise.}
$$

(2.14)

Hence, we can conclude that the graph $\bar{G}(0)$ is a subgraph of the graph $\bar{G}(t)$ for all $t \geq 0$. It follows from Lemma 5 that $H(0) \leq H(t)$. Therefore, we can get from (2.13) that

$$
\dot{V} \leq -\gamma \tilde{q}^T [H(0) \otimes I_p] \tilde{q}.
$$

(2.15)
Since in $G(0)$ the leader has directed paths to all followers, it follows from Lemma 4 that $H(0)$ is symmetric positive definite. Integrating both sides of (2.15), we can obtain that $\dot{q} \in L_2$. Note that $\dot{q}_0$ is constant and hence bounded. Combining the above boundedness arguments we can get from (2.1) that $\dot{q}_i, \ddot{q}_i, v_i \in L_\infty$. Since $V_{ij}$ is continuously differentiable, we can get from (2.14) that $\frac{\partial V_{ij}}{\partial q_i} \in L_\infty$. From (2.4) and (2.5), we have $\dot{u}_i, \dot{v}_i \in L_\infty$. Then from (2.7) and the property (P1), it can be concluded that $\dot{s}_i \in L_\infty$. By noting that $\dot{s}_i = \dot{q}_i - \dot{v}_i$, it follows that $\ddot{q}_i \in L_\infty$. Overall, we have $\dot{q}_i \in L_\infty \cap L_2$ and $\ddot{q}_i \in L_\infty$. From Barbalat’s lemma [69], we can conclude that $\dot{q}_i \to 0$, that is, $\|\dot{q}_i - \dot{q}_0\| \to 0$ asymptotically. 

**Remark 8** As it can be seen, by using the control law (2.3)-(2.6) for (1.1), the followers can track the leader with the same velocity while avoiding collision and maintaining the initial connectivity. Note that with our algorithm design, as long as at the initial time the connectivity is maintained and there is no collision, the connectivity maintenance and collision avoidance are ensured for all time. The proposed algorithm is continuous and accounts for unknown parameters of the agents’ dynamics.

### 2.1.2 Flocking of Lagrange Networks, Where the Leader Moves with a Varying Velocity

In this subsection, we consider the case where the leader moves with a varying velocity. In this case, the problem is more difficult to tackle since all followers must track the leader while the leader’s velocity changes over time and the leader is a neighbor of only a subset of the followers. In the remainder of this chapter, we have the following assumption on the leader.

**Assumption 9** The leader’s velocity $\dot{q}_0$ and acceleration $\ddot{q}_0$ are both bounded. It is assumed that
\[ \|1_n \otimes \dot{q}_0\| \leq \sigma_l, \text{ where } \sigma_l \in \mathbb{R}^+ \text{ is constant.} \]

We propose the following distributed control algorithm

\[ u_i = \hat{u}_i + Y_i(q_i, \dot{q}_i, \dot{v}_i, v_i)\dot{\theta}_i, \]  \hspace{1cm} (2.16)

\[ \hat{u}_i = -n \sum_{j=0}^{n} \frac{\partial V_{ij}}{\partial q_i} - \gamma n \sum_{j=0}^{n} a_{ij}(t)(\dot{q}_i - \dot{q}_j) - \beta \chi_i - \alpha \text{sgn}(s_i), \]  \hspace{1cm} (2.17)

\[ \dot{v}_i = -n \sum_{j=0}^{n} \frac{\partial V_{ij}}{\partial q_i} - \gamma n \sum_{j=0}^{n} a_{ij}(t)(\dot{q}_i - \dot{q}_j) - \beta \chi_i, \]  \hspace{1cm} (2.18)

\[ \dot{\theta}_i = -\Gamma_i Y_i^T(q_i, \dot{q}_i, \dot{v}_i, v_i)s_i, \]  \hspace{1cm} (2.19)

where \( \gamma, \alpha, \beta \in \mathbb{R}^+ \) are control gains to be designed, \( a_{ij}(t), V_{ij}, s_i \) and \( \Gamma_i \) are defined in Subsection 2.1.1 with the additional assumptions that \( a_{0i} = 0 \), \( i = 1, \ldots, n \) and

\[ \chi_i = \sum_{j=0}^{n} a_{ij}(t)\{\text{sgn}\left[\sum_{k=0}^{n} a_{ik}(t)(\dot{q}_i - \dot{q}_k)\right] - \text{sgn}\left[\sum_{k=0}^{n} a_{jk}(t)(\dot{q}_j - \dot{q}_k)\right]\}. \]

**Theorem 10** Suppose that at the initial time \( t = 0 \), the leader has directed paths to all followers and there is no collision among the agents. Using the control law (2.16)-(2.19) for (1.1), if \( \alpha \geq \sigma_l \) and \( \beta \geq \frac{\sigma_l}{\lambda_{\min}[H(0)]} \), then the leader-follower flocking is achieved asymptotically.

**Proof:** Using (2.16) for (1.1), we will have

\[ M_i(q_i)s_i + C_i(q_i, \dot{q}_i)s_i = \hat{u}_i - Y_i(q_i, \dot{q}_i, \dot{v}_i, v_i)\dot{\theta}_i. \]  \hspace{1cm} (2.20)

Define the same function \( V_1 \) as in (2.8). The derivative of \( V_1 \) can be written as (2.9). By substituting (2.17) to (2.9), \( \dot{V}_1 \) can be rewritten as

\[ \dot{V}_1 = -\sum_{i=1}^{n} s_i^T \sum_{j=0}^{n} \frac{\partial V_{ij}}{\partial q_i} - \gamma \sum_{i=1}^{n} \sum_{j=0}^{n} a_{ij}(t)s_i^T(q_i - q_j) - \beta \sum_{i=1}^{n} s_i^T \chi_i - \alpha \|s\|_1. \]  \hspace{1cm} (2.21)

\footnote{Since at the initial time \( t = 0 \), the leader has directed paths to all followers, we can get from Lemma 4 that \( \lambda_{\min}[H(0)] > 0 \), and thus the term \( \frac{\sigma_l}{\lambda_{\min}[H(0)]} \) is well defined.}
Now we consider the following Lyapunov function candidate

\[ V = V_1 + \frac{1}{2} \sum_{i=1}^{n} \ddot{v}_i^T \ddot{v}_i + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} V_{ij} + \sum_{i=1}^{n} V_{i0}, \tag{2.22} \]

where \( \ddot{v}_i \) is defined in (2.1). The derivative of \( V \) is given as

\[
\dot{V} = \dot{V}_1 + \sum_{i=1}^{n} \ddot{v}_i^T \dddot{v}_i + \sum_{i=1}^{n} \sum_{j=1}^{n} \dddot{q}_i^T \frac{\partial V_{ij}}{\partial q_i} + \sum_{i=1}^{n} \dddot{q}_i^T \frac{\partial V_{i0}}{\partial q_i} \\
= -\sum_{i=1}^{n} \dddot{q}_i^T \sum_{j=0}^{n} \frac{\partial V_{ij}}{\partial q_i} + \sum_{i=1}^{n} \dddot{v}_i^T \sum_{j=0}^{n} \frac{\partial V_{ij}}{\partial q_i} - \gamma \sum_{i=1}^{n} \sum_{j=0}^{n} a_{ij}(t) \dddot{q}_i^T (q_i - q_j) \\
+ \gamma \sum_{i=1}^{n} \sum_{j=0}^{n} a_{ij}(t) \dddot{q}_i^T (q_i - q_j) - \beta \sum_{i=1}^{n} \dddot{q}_i^T \chi_i + \beta \sum_{i=1}^{n} \dddot{v}_i^T \chi_i - \alpha \|s\|_1 \\
+ \gamma \sum_{i=1}^{n} \sum_{j=0}^{n} \frac{\partial V_{ij}}{\partial q_i} \\
= \sum_{i=1}^{n} \dddot{v}_i^T \sum_{j=0}^{n} \frac{\partial V_{ij}}{\partial q_i} - \gamma \dddot{q}^T [H(t) \otimes I_p] \ddot{q} + \gamma \dddot{v}^T [H(t) \otimes I_p] \ddot{q} - \beta \dddot{q}^T \{H(t) \otimes \text{sgn}([H(t) \otimes I_p] \ddot{q})\} \\
+ \beta \dddot{v}^T \{H(t) \otimes \text{sgn}([H(t) \otimes I_p] \ddot{q})\} - \alpha \|s\|_1 + \dddot{v}^T \dddot{v}, \tag{2.23} \]

where \( \dddot{v} \) is a column stack vector of all \( \dddot{v}_i \)'s, \( i = 1, \ldots, n \). According to (2.18), we can rewrite (2.23) as

\[
\dot{V} = \sum_{i=1}^{n} \dddot{v}_i^T \sum_{j=0}^{n} \frac{\partial V_{ij}}{\partial q_i} - \gamma \dddot{q}^T [H(t) \otimes I_p] \ddot{q} + \gamma \dddot{v}^T [H(t) \otimes I_p] \ddot{q} - \beta \dddot{q}^T \{H(t) \otimes \text{sgn}([H(t) \otimes I_p] \ddot{q})\} \\
+ \beta \dddot{v}^T \{H(t) \otimes \text{sgn}([H(t) \otimes I_p] \ddot{q})\} - \alpha \|s\|_1 - \sum_{i=1}^{n} \dddot{v}_i \sum_{j=0}^{n} \frac{\partial V_{ij}}{\partial q_i} - \gamma \dddot{v}^T [H(t) \otimes I_p] \ddot{q} \\
- \beta \dddot{v}^T \{H(t) \otimes \text{sgn}([H(t) \otimes I_p] \ddot{q})\} - \dddot{v} (1_n \otimes \dddot{q}_0) \\
= -\gamma \dddot{q}^T [H(t) \otimes I_p] \ddot{q} - \beta \| [H(t) \otimes I_p] \ddot{q} \|_1 - \alpha \|s\|_1 - \dddot{q}^T (1_n \otimes \dddot{q}_0) + \dddot{q}^T (1_n \otimes \dddot{q}_0) \\
\leq -\gamma \dddot{q}^T [H(t) \otimes I_p] \ddot{q} - \beta \| [H(t) \otimes I_p] \ddot{q} \|_1 - \alpha \|s\|_1 + \| 1_n \otimes \dddot{q}_0 \| \cdot \| \dddot{q} \| + \| 1_n \otimes \dddot{q}_0 \| \cdot \| s \|_1, \tag{2.24} \]
where we have used the Cauchy-Schwartz inequality and $\|\cdot\| \leq \|\cdot\|_1$ to obtain the inequality. Since $\|1_n \otimes \hat{q}_0\| \leq \sigma_l$, we have

$$
\dot{V} \leq -\gamma \hat{q}^T [H(t) \otimes I_p] \hat{q} - \beta \|H(t) \otimes I_p\|_1 + (\sigma_l - \alpha)\|s\|_1 + \sigma_l\|\hat{q}\|
$$

$$
\leq -\gamma \hat{q}^T [H(t) \otimes I_p] \hat{q} - \beta \|H(t) \otimes I_p\|_1 + (\sigma_l - \alpha)\|s\|_1 + \sigma_l\|\hat{q}\|
$$

$$
\leq -\gamma \hat{q}^T [H(t) \otimes I_p] \hat{q} - \beta \lambda_{\min}[H(t)]\|\hat{q}\| + (\sigma_l - \alpha)\|s\|_1 + \sigma_l\|\hat{q}\|.
$$

(2.25)

where we have used $\|\cdot\| \leq \|\cdot\|_1$ and the fact that $H(t)$ is positive semidefinite to obtain, respectively, the second and the third inequalities. Since at the initial time, $t = 0$, the leader has directed paths to all followers, we can get from Lemma 4 that $H(0)$ is symmetric positive definite and hence $\lambda_{\min}[H(0)] > 0$. Thus, by using the two conditions $\alpha \geq \sigma_l$ and $\beta \geq \frac{\sigma_l}{\lambda_{\min}[H(0)]}$, we have at $t = 0$ that,

$$
\dot{V}(t) \leq -\gamma \hat{q}^T [H(0) \otimes I_p] \hat{q} \leq 0.
$$

(2.26)

Although the control input $u_i$ is discontinuous, the positions of the agents are continuous and $H(t)$ changes according to the relative positions among the agents. Suppose that $H(t)$ changes at some times, then there exists a time $t_1 > 0$ such that, $H(t) = H(0)$ for $t \in [0, t_1)$ and $H(t_1) \neq H(0)$. Therefore, we have

$$
\dot{V}(t) \leq -\gamma \hat{q}^T [H(0) \otimes I_p] \hat{q} \leq 0, \quad t \in [0, t_1),
$$

(2.27)

which implies that $V_{ij} \in L_\infty$ for all pairs of $q_i(t), q_j(t)$, when $t \in [0, t_1)$. Since $V_{ij}$ is continuous, it can be concluded that $V_{ij} \in L_\infty$ at $t = t_1$. Thus, by using the definition of $V_{ij}$ for $t \in [0, t_1]$, there is no collision and no edge in the graph $\overline{G}(0)$ is lost. Therefore, the only possibility that $H(t)$ changes at $t = t_1$ is that, some edges are added to the graph. It implies that $\overline{G}(0)$ is a subgraph.
of $\overline{G}(t_1)$ which means from Lemma 5 that $H(0) \leq H(t_1)$ and hence $\lambda_{\min}[H(0)] \leq \lambda_{\min}[H(t_1)]$.

Therefore, at time $t = t_1$,

$$\dot{V}(t) \leq -\gamma \hat{q}^T[H(t_1) \otimes I_p] \hat{q} - \beta \lambda_{\min}[H(t_1)] \| \hat{q} \| + \sigma_1 \| \hat{q} \|$$

$$\leq -\gamma \hat{q}^T[H(0) \otimes I_p] \hat{q} - \beta \lambda_{\min}[H(0)] \| \hat{q} \| + \sigma_1 \| \hat{q} \|$$

$$\leq -\gamma \hat{q}^T[H(0) \otimes I_p] \hat{q} \leq 0.$$ (2.28)

From the same statements, if $H(t)$ changes at $t = t_i > t_1$, $i = 2, \ldots$, we can get that $V_{ij}$ will always be bounded. Hence there is no collision and no edge in the graph $\overline{G}(0)$ will be lost. This in turn implies that for all $t \in [t_i, t_{i+1})$, $\overline{G}(0)$ is a subgraph of $\overline{G}_{i}(t)$. It thus follows that for all $t \in [t_i, t_{i+1})$, $H(0) \leq H_i(t)$ and $\lambda_{\min}[H(0)] \leq \lambda_{\min}[H_i(t)]$. That is, $\beta \geq \frac{\sigma_1}{\lambda_{\min}[H(0)]} \geq \frac{\sigma_1}{\lambda_{\min}[H_i(t)]}$ for all $t \geq 0$. Hence (2.28) holds for all $t \geq 0$. We then can get that $s_i, \tilde{\theta}_i, \tilde{v}_i \in \mathbb{L}_\infty$. Since $V(t) \geq 0$ and $\dot{V}(t) \leq 0$, it is concluded that $V_{\infty, i} \triangleq \lim_{t \to \infty} V(t) \in [0, V(0)]$ exists. Thus, integrating both sides of (2.28), we can obtain that $\dot{\hat{q}} \in \mathbb{L}_1$. Note from Assumption 9 that both $\dot{\hat{q}}_0$ and $\ddot{\hat{q}}_0$ are bounded.

Combining the above boundedness arguments, we can get from (2.1) that $\dot{\hat{q}}_i, \ddot{\hat{q}}_i, v_i \in \mathbb{L}_\infty$. Following the same statements from the proof of Theorem 7, we can conclude that $\frac{\partial V_{ij}}{\partial q_i} \in \mathbb{L}_\infty$. From (2.17) and (2.18), we have $\hat{u}_i, \hat{v}_i \in \mathbb{L}_\infty$. Then from the closed-loop dynamics for each follower and (P1), we have $\dot{s}_i \in \mathbb{L}_\infty$. By noting that $\dot{s}_i = \ddot{q}_i - \dot{v}_i$, it follows that $\ddot{q}_i \in \mathbb{L}_\infty$ and thus $\ddot{q}_i \in \mathbb{L}_\infty$. Overall, we have $\ddot{q}_i \in \mathbb{L}_\infty \cap \mathbb{L}_1$ and $\dddot{q}_i \in \mathbb{L}_\infty$. From Barbalat’s lemma [69], we can conclude that $\dddot{q}_i \to 0$. That is, $\| \dddot{q}_i - \dddot{q}_0 \| \to 0$ asymptotically. ■

**Remark 11** As it can be seen, the proposed algorithm (2.16)-(2.19) guarantees that leader-follower flocking is achieved, where the leader has a varying velocity in the presence of unknown parameters. Therefore, despite the hard restrictions such as nonlinear Lagrange dynamics, unknown models’
parameters, and the existence of a moving leader with a varying velocity, the control input (2.16)-(2.19) solves the flocking problem.

**Remark 12** Due to the existence of the signum function, the closed-loop dynamics of (1.1) using (2.16) is discontinuous. The solution should be investigated in terms of differential inclusions. Note that the signum function is measurable and locally essentially bounded. Therefore, from the nonsmooth analysis in [27], the Filippov solutions for the closed-loop dynamics always exist. Because the Lyapunov function candidate in the proof of Theorem 10 is continuously differentiable and the set-valued Lie derivative of the Lyapunov function is a singleton at the discontinuous point, the proof of Theorem 10 still holds. To avoid symbol redundancy, we do not use the differential inclusions in the proof. It is worthy mentioning that the drawback of the signum function is the potential chattering behavior. In practice, a simple and useful way to avoid the discontinuous control action is to replace the signum function with a smooth function such as $\tanh(\cdot)$, with which satisfactory performance can still be achieved, as confirmed in our later simulation.

**Remark 13** The case of a leader with a constant velocity is a special case of a leader with a varying velocity. Hence we can also use the algorithm (2.16)-(2.19) for the leader-follower flocking problem, where the leader has a constant velocity. However, the algorithm (2.3)-(2.6) is continuous. In contrast, the algorithm (2.16)-(2.19) is discontinuous and may cause the chattering issues. Therefore, when the leader has a constant velocity, the algorithm (2.3)-(2.6) is more favorable than the algorithm (2.16)-(2.19).
2.1.3 Fully Distributed Flocking of Lagrange Networks with One-hop Neighbors’ Information, Where the Leader Has a Varying velocity

In Subsection 2.1.2, each follower needs its two-hop neighbors’ information to track the leader. Besides, all followers use common gains in their control inputs and the gains should be above certain bounds which are actually determined by the global information \( \lambda_{\min}[H(0)], \sigma_l \). Therefore, the algorithm (2.16)-(2.19) is not fully distributed. In this subsection, the previous algorithm is improved to be fully distributed and use one-hop neighbors’ information.

The following control algorithm and gain adaptation law for each follower is designed as

\[
\begin{align*}
    u_i &= \hat{u}_i + Y_i(q_i, \dot{q}_i, \dot{v}_i, v_i)\hat{\theta}_i, \\
    \dot{\hat{u}}_i &= -\sum_{j=0}^{n} \frac{\partial V_{ij}}{\partial q_i} - \sum_{j=0}^{n} \alpha_{ij}a_{ij}(t)\text{sgn}(\dot{q}_i - \dot{q}_j) - \beta_i\text{sgn}(s_i), \\
    \dot{\hat{v}}_i &= -\sum_{j=0}^{n} \frac{\partial V_{ij}}{\partial q_i} - \sum_{j=0}^{n} \alpha_{ij}a_{ij}(t)\text{sgn}(\dot{q}_i - \dot{q}_j), \\
    \dot{\alpha}_{ij} &= \gamma_{1ij}a_{ij}(t)\|\dot{q}_i - \dot{q}_j\|_1, \\
    \dot{\beta}_i &= \gamma_{2i}\|s_i\|_1, \\
    \dot{\hat{\theta}}_i &= -\Gamma_iY_i^T(q_i, \dot{q}_i, \hat{v}_i, v_i)s_i,
\end{align*}
\]

where \( a_{ij}(t) \), \( V_{ij} \), \( s_i \) and \( \Gamma_i \) are defined in Subsection 2.1.1, \( \gamma_{1ij}, \gamma_{2i} \) are positive constants with \( \gamma_{1ij} = \gamma_{1ji} \), and \( \alpha_{ij}(t), \beta_i(t) \) are varying gains with initial values \( \alpha_{ij}(0) = \alpha_{ji}(0) \geq 0 \) and \( \beta_i(0) \geq 0 \).

**Remark 14** The gain adaptation laws (2.32) and (2.33) are inspired by recent results on adaptive gain design for multi-agent systems [45, 46, 84]. The intuition behind (2.32) and (2.33) is that the control gains in Theorem 10 must be above certain lower bounds. Under (2.32) and (2.33), as
long as the velocity matching is not achieved, the gains will always increase, eventually rendering the agents to achieve velocity matching. The drawback of (2.32) and (2.33) is that the 1-norm of the signals will result in the non-stop increase of the gains in the presence of disturbances or measurement errors. Here we just show the theoretical analysis in the ideal situation. In practice, one alteration is to introduce a small bound on the right hand sides (RHSs) of (2.32) and (2.33). When the RHSs of (2.32) and (2.33) are within some given bound, \( \alpha_{ij} \) and \( \beta_i \) stop increasing.

**Theorem 15** Suppose that at the initial time \( t = 0 \), the leader has directed paths to all followers and there is no collision among the agents. Using (2.29)-(2.34) for (1.1), the leader-follower flocking is achieved.

**Proof**: Define \( V_3 = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{1}{4\gamma_{ij}} (\alpha_{ij} - \bar{\alpha})^2 + \sum_{i=1}^{n} \frac{1}{2\gamma_{i0}} (\alpha_{i0} - \bar{\alpha})^2 + \sum_{i=1}^{n} \frac{1}{2\gamma_{i}} (\beta_i - \bar{\beta})^2 \), where \( \bar{\alpha} \) and \( \bar{\beta} \) are chosen such that \( \bar{\alpha} > \frac{\sigma_l}{\sqrt{\lambda_{\text{min}}[H(0)]}} \) and \( \bar{\beta} > \sigma_l \). The derivative of \( V_3 \) is given as

\[
\dot{V}_3 = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{1}{2} a_{ij}(t)(\alpha_{ij} - \bar{\alpha})(\dot{q}_i - \dot{q}_j) + \sum_{i=1}^{n} a_{i0}(t)(\alpha_{i0} - \bar{\alpha})(\dot{q}_i - \dot{q}_0)^T \text{sgn}(\dot{q}_i - \dot{q}_0) + \sum_{i=1}^{n} (\beta_i - \bar{\beta})||s_i||_1
\]

\[
= \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{1}{2} a_{ij}(t)(\alpha_{ij} - \bar{\alpha})\dot{\hat{q}}_i^T \text{sgn}(\hat{q}_i - \hat{q}_j) + \sum_{i=1}^{n} a_{i0}(t)(\alpha_{i0} - \bar{\alpha})\dot{\hat{q}}_i^T \text{sgn}(\hat{q}_i - \hat{q}_0) + \sum_{i=1}^{n} (\beta_i - \bar{\beta})||s_i||_1
\]

\[
= \sum_{i=1}^{n} \sum_{j=0}^{n} a_{ij}(t)(\alpha_{ij} - \bar{\alpha})\dot{\hat{q}}_i^T \text{sgn}(\hat{q}_i - \hat{q}_j) + (\beta_i - \bar{\beta})||s_i||_1. 
\]  

Since at the initial time \( t = 0 \), the leader has directed paths to all followers, we can get from Lemma 4 that \( \lambda_{\text{min}}[H(0)] > 0 \), and thus the term \( \frac{\sigma_l}{\sqrt{\lambda_{\text{min}}[H(0)]}} \) is well defined.
Now we introduce the following Lyapunov function candidate

\[ V = V_1 + \frac{1}{2} \sum_{i=1}^{n} \tilde{\eta}_i^T \tilde{\eta}_i + V_2 + V_3, \]

where \( V_1 \) is defined in (2.8) and \( V_2 \) is defined in (2.10). Note that using (2.29) for (1.1), where \( \tilde{\eta}_i \) is given by (2.30), both (2.7) and (2.9) still hold. The derivative of \( V \) is given as

\[ \dot{V} = \sum_{i=1}^{n} s_i^T \dot{u}_i + \sum_{i=1}^{n} \dot{\eta}_i^T \dot{\eta}_i + \sum_{i=1}^{n} \sum_{j=0}^{n} \dot{\eta}_i^T \frac{\partial V_{ij}}{\partial \eta_i} + \dot{V}_3. \]  
(2.36)

Note from (2.30) and (2.31) that

\[ \dot{v}_i = \dot{\eta}_i + \beta_i \text{sgn}(s_i). \]

Also note from (2.1) that \( \dot{v}_i = \dot{\eta}_i - \bar{q}_0 \). Thus, it follows from (2.36) that

\[ \dot{V} = \sum_{i=1}^{n} s_i^T \dot{u}_i + \sum_{i=1}^{n} \dot{\eta}_i^T [\dot{\eta}_i + \beta_i \text{sgn}(s_i) - \bar{q}_0] + \sum_{i=1}^{n} \sum_{j=0}^{n} \dot{\eta}_i^T \frac{\partial V_{ij}}{\partial \eta_i} + \dot{V}_3 
= \sum_{i=1}^{n} \dot{\eta}_i^T [\dot{\eta}_i + \beta_i \text{sgn}(s_i) - \bar{q}_0] - \sum_{i=1}^{n} s_i^T [\beta_i \text{sgn}(s_i) - \bar{q}_0] + \sum_{i=1}^{n} \sum_{j=0}^{n} \dot{\eta}_i^T \frac{\partial V_{ij}}{\partial \eta_i} + \dot{V}_3, \]  
(2.37)

where we have used \( \dot{v}_i = \dot{\eta}_i - s_i \) to obtain the second equality. Substituting (2.30) and (2.35) to (3.13) and by doing some mathematics manipulation, we can get

\[ \dot{V} = -\sum_{i=1}^{n} \sum_{j=0}^{n} \dot{\eta}_i \frac{\partial V_{ij}}{\partial \eta_i} - \sum_{i=1}^{n} \sum_{j=0}^{n} \alpha_{ij} a_{ij}(t) \dot{\eta}_i^T \text{sgn}(\dot{\eta}_i - \hat{q}_j) - \sum_{i=1}^{n} \dot{q}_i \bar{q}_0 - \sum_{i=1}^{n} \beta_i \|s_i\|_1 + \sum_{i=1}^{n} s_i^T \bar{q}_0 
+ \sum_{i=1}^{n} \sum_{j=0}^{n} \dot{\eta}_i \frac{\partial V_{ij}}{\partial \eta_i} + \sum_{i=1}^{n} \sum_{j=0}^{n} \alpha_{ij}(t)(\alpha_{ij} - \bar{q}_i) \dot{\eta}_i^T \text{sgn}(\dot{\eta}_i - \hat{q}_j) + \sum_{i=1}^{n} (\beta_i - \bar{q}_i) \|s_i\|_1 
= -\alpha \dot{q}_F^T [D_F(t) \otimes I_p] \text{sgn}([D_F^T(t) \otimes I_p] \dot{\hat{q}}) - \alpha \dot{q}_F^T [\Lambda(t) \otimes I_p] \text{sgn}([\Lambda(t) \otimes I_p] \dot{\hat{q}}) - \dot{q}_F^T (1_n \otimes \bar{q}_0) 
- \beta \|s\|_1 + s^T (1_n \otimes \bar{q}_0) 
= -\alpha \|D_F^T(t) \otimes I_p\|_1 - \alpha \|\Lambda(t) \otimes I_p\|_1 - \dot{q}_F^T (1_n \otimes \bar{q}_0) - \beta \|s\|_1 + s^T (1_n \otimes \bar{q}_0) \]
\[
\begin{align*}
\dot{V} &\leq -\tilde{\alpha}\|D_F^T(t) \otimes I_p\tilde{\hat{q}}\| - \tilde{\alpha}\|\Lambda(t) \otimes I_p\tilde{\hat{q}}\| + \|1_n \otimes \tilde{\hat{q}}_0\| \cdot \|\tilde{\hat{q}}\| - \tilde{\beta}\|s\| + \|1_n \otimes \tilde{\hat{q}}_0\| \cdot \|s\|,
\end{align*}
\] (2.38)

where \(\tilde{q}\) and \(s\) are, respectively, column stack vectors of \(\tilde{q}_i\) and \(s_i, i = 1, \ldots, n\), and we have used the Cauchy-Swartz inequality to obtain the inequality. Since \(\|1_n \otimes \tilde{\hat{q}}_0\| \leq \sigma_l\), we will have

\[
\dot{V} \leq -\tilde{\alpha}\|D_F^T(t) \otimes I_p\tilde{\hat{q}}\| - \tilde{\alpha}\|\Lambda(t) \otimes I_p\tilde{\hat{q}}\| + \|1_n \otimes \tilde{\hat{q}}_0\| \cdot \|\tilde{\hat{q}}\| - \tilde{\beta}\|s\| + \|1_n \otimes \tilde{\hat{q}}_0\| \cdot \|s\|
\]

thus, \(\dot{V} \leq 0\) for \(t \geq 0\) and \(v(t) = 0\) for \(t = 0\). Therefore, \(V(t)\) is positive definite. Similarly, we can show that \(\dot{V} = -\alpha\|\tilde{\hat{q}}\| - \alpha\|s\|\) for \(\alpha > 0\). Therefore, \(V(t)\) is negative definite and \(v(t) = 0\) for \(t = 0\). Therefore, \(V(t)\) is positive definite. Similarly, we can show that \(\dot{V} = -\alpha\|\tilde{\hat{q}}\| - \alpha\|s\|\) for \(\alpha > 0\). Therefore, \(V(t)\) is negative definite and \(v(t) = 0\) for \(t = 0\). Therefore, \(V(t)\) is positive definite. Similarly, we can show that \(\dot{V} = -\alpha\|\tilde{\hat{q}}\| - \alpha\|s\|\) for \(\alpha > 0\). Therefore, \(V(t)\) is negative definite and \(v(t) = 0\) for \(t = 0\). Therefore, \(V(t)\) is positive definite. Similarly, we can show that \(\dot{V} = -\alpha\|\tilde{\hat{q}}\| - \alpha\|s\|\) for \(\alpha > 0\). Therefore, \(V(t)\) is negative definite and \(v(t) = 0\) for \(t = 0\). Therefore, \(V(t)\) is positive definite. Similarly, we can show that \(\dot{V} = -\alpha\|\tilde{\hat{q}}\| - \alpha\|s\|\) for \(\alpha > 0\). Therefore, \(V(t)\) is negative definite and \(v(t) = 0\) for \(t = 0\). Therefore, \(V(t)\) is positive definite. Similarly, we can show that \(\dot{V} = -\alpha\|\tilde{\hat{q}}\| - \alpha\|s\|\) for \(\alpha > 0\). Therefore, \(V(t)\) is negative definite and \(v(t) = 0\) for \(t = 0\). Therefore, \(V(t)\) is positive definite.
Thus, it is concluded that
\[\bar{\alpha} > \frac{\sigma_1}{\sqrt{\lambda_{\min}(H(0))}} \geq \frac{\sigma_1}{\sqrt{\lambda_{\min}(H_i(t))}}\]
and \(\dot{V}(t) \leq 0\) for all \(t \geq 0\). Again by using the same analyses as the proof of Theorem 10, it is proved that
\[\dot{q}_i \in L_\infty \cap L_1\] and \(\ddot{q}_i \in L_\infty\). Hence from Barbalat’s lemma, we conclude that \(\|\dot{q}_i - \dot{q}_0\| \to 0\) asymptotically.

**Remark 16** In the fully distributed algorithm (2.29)-(2.32) adaptive gain schemes are introduced. This algorithm guarantees that the leader-follower flocking is achieved, where the leader has a varying velocity in the presence of unknown parameters and there is no requirement of any global information. Therefore, despite the hard restrictions due to uncertainty in followers’ dynamics and the existence of a moving leader with a varying velocity, the fully distributed control input (2.29)-(2.32) solves the flocking problem.

**Remark 17** In [5], the distributed flocking problem with a moving leader has been solved for multi-agent systems with single- or double-integrators. Here in the proposed algorithm (2.29)-(2.32), we address the problem for networked nonlinear Lagrange systems with parametric uncertainties, which is more challenging. The algorithms in [5] cannot deal with nonlinear Lagrange dynamics and account for fully distributed gain design. Besides, the algorithms in [5] rely on both one-hop and two-hop neighbors’ information, while only one-hop neighbors’ information is required in (2.29)-(2.32). In [47] a distributed coordinated tracking problem is studied for Lagrange systems with parametric uncertainties. However, the algorithms in [47] cannot deal with the nonlinear flocking behavior or account for fully distributed gain design and still requires the two-hop neighbors’ information.
2.2 Simulation

In this section, numerical simulation results are given to illustrate the effectiveness of the theoretical results obtained in Section 2.1. We consider the formation flying of four spacecraft, where the formation control is based on the relative translation with respect to a virtual point or chief spacecraft following a circular reference orbit [1]. The relative dynamics of the \( i \)th spacecraft is considered in a chief-fixed, LVLH rotating frame, which can be written as

\[
\begin{align*}
    m_i \ddot{x}_i &- 2m_i n_0 \dot{y}_i - m_i n_0^2 x_i + \frac{m_i \mu_e (r_0 + x_i)}{r_i^3} - \frac{m_i \mu_e}{r_0^2} = u_{ix}, \\
    m_i \ddot{y}_i &+ 2m_i n_0 \dot{x}_i - m_i n_0^2 y_i + \frac{m_i \mu_e y_i}{r_i^3} = u_{iy}, \\
    m_i \ddot{z}_i + \frac{m_i \mu_e z_i}{r_i^3} = u_{iz},
\end{align*}
\]

where \( m_i \) is the unknown but constant mass of the \( i \)th spacecraft, \( \mu_e \) is the gravitational constant of Earth, \( r_0 \) is the radius of the chief, \( n_0 = \sqrt{\mu_e / r_0^3} \) is the angular velocity of the reference orbit, \( q_i = [x_i, y_i, z_i]^T \) is the position of the \( i \)th spacecraft in the LVLH frame, and \( u_i = [u_{ix}, u_{iy}, u_{iz}]^T \) is the control input. Let \( M_i = m_i I_3 \), \( C_i = m_i \begin{pmatrix} 0 & -2n_0 & 0 \\ 2n_0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \), \( g_i = m_i \begin{pmatrix} -n_0^2 x_i + \frac{\mu_e (r_0 + x_i)}{r_i^3} - \frac{\mu_e}{r_0^2} \\ -n_0^2 y_i + \frac{\mu_e y_i}{r_i^3} \\ \frac{\mu_e z_i}{r_i^3} \end{pmatrix} \), then the relative translation dynamics can be written in the form of (1.1), with the unknown parameter \( \theta_i = m_i \).

In the simulations, we let \( m_i = 30 + 5i \text{ kg} \), \( r_0 = 7000 \text{ km} \), and \( R = 200 \text{ m} \). The initial positions of the leader and the four spacecraft are, respectively, \( q_0(0) = [-80, 200, 0]^T \text{ m} \), \( q_1(0) = [-80, 90, 0]^T \text{ m} \), \( q_2(0) = [100, 90, 0]^T \text{ m} \), \( q_3(0) = [100, -100, 0]^T \text{ m} \), and \( q_4(0) = [-80, -100, 0]^T \text{ m} \).
The initial velocities are assumed to be zero. The unique minimums of $V_{ij}$ are assumed to be 80 m.

Following [5], when $\|q_i(0) - q_j(0)\| \geq 200$, the potential functions are defined whose partial derivatives satisfy

$$\frac{\partial V_{ij}}{\partial q_i} = \begin{cases} 
0, & \|q_i - q_j\| > 200; \\
\frac{(q_i - q_j)\cos(0,1\pi(\|q_i - q_j\|-80))}{250\|q_i - q_j\|}, & 80 < \|q_i - q_j\| \leq 200; \\
\frac{(q_i - q_j)(\|q_i - q_j\|-80)}{250\|q_i - q_j\|^2}, & \|q_i - q_j\| \leq 80.
\end{cases}$$

When $\|q_i(0) - q_j(0)\| < 200$, the potential functions are defined whose partial derivatives satisfy

$$\frac{\partial V_{ij}}{\partial q_i} = \begin{cases} 
\frac{(q_i - q_j)(\|q_i - q_j\|-80)}{250\|q_i - q_j\|}, & 80 < \|q_i - q_j\| \leq 200; \\
\frac{(q_i - q_j)(\|q_i - q_j\|-80)}{250\|q_i - q_j\|^2}, & \|q_i - q_j\| \leq 80.
\end{cases}$$

In the first case, we simulate the case where the leader has a constant velocity under the control algorithm (2.3)-(2.6). The constant velocity of the leader is assumed to be $\dot{q}_0 = [0, 0.1, 0.2]^T$. The initial values for the estimates of the leader’s velocity are all zero. The control parameter is chosen as $\gamma = 0.04$ and $\Gamma_i = 5I_3$, $i = 1, \ldots, 4$. Fig. 2.2 shows the trajectories of the leader and the followers. Clearly, all followers move cohesively with the leader without colliding with each other. Fig. 2.3 shows the velocity of the followers and the leader. It can be seen that the velocities of the followers converge to that of the leader and all agents move with the same velocity. There are two new edges added to the graph and no edge is lost.

In the second case, we simulate the case where the leader has a varying velocity under the control algorithm (2.16)-(2.19). The initial states of the followers are chosen as above and the leader’s velocity is chosen as $\dot{q}_0(t) = [0.1\sin(\frac{2\pi}{60}t), 0.1\cos(\frac{2\pi}{60}t), 0.2]^T$. The initial position of the leader is chosen as $q_0(0) = [-80, 200, 0]^T$. The control parameters are chosen as $\alpha = 0.04$, and $\Gamma_i = 5I_3$, $i = 1, \ldots, 4$. We use $\tanh(1000\cdot)$ to replace the function $\text{sgn}(\cdot)$. Fig. 2.4 shows
the trajectories of the followers and the leader. The agents maintain the initial connectivity while avoiding collisions. Fig. 2.5 shows that each follower eventually moves with the same velocity as the leader. Similarly, there are two new edges added to the graph and no edge is lost.

In the third case, we simulate the case where the leader has a varying velocity under the fully distributed control algorithm (2.29)-(2.34). Here the initial states and the leader’s trajectory are chosen as the second case. The control parameter is chosen as \( \Gamma_i = 5I_3, \gamma_{1ij} = \gamma_{2i} = 0.003, i = 1, \ldots, 4 \), where agent \( j \) is the neighbor of agent \( i \). Fig. 2.8 shows the trajectories while Fig. 2.9 shows the velocities of the followers and the leader. It can be seen that the leader-following flocking is achieved and there is no edge added or lost.
Figure 2.2: The trajectories of the followers and the leader in the first case. The leader is represented as a square while the followers are represented as circles. An edge between two followers denotes that the two are neighbors, and an arrow from the leader to a follower denotes that the leader is a neighbor of the follower.
Figure 2.3: The velocity errors between the followers and the leader using (2.3)-(2.6).
Figure 2.4: The trajectories of the followers and the leader in the second case.
Figure 2.5: The velocity errors between the followers and the leader using (2.16)-(2.19).
Figure 2.6: The trajectories of the followers and the leader in the second case
Figure 2.7: The velocities of the followers and the leader in the second case
Figure 2.8: The trajectories of the followers and the leader in the third case.
Figure 2.9: The velocity errors between the followers and the leader using (2.29)-(2.34).
Chapter 3

Distributed Average Tracking for

Physical Second-Order Linear and

Nonlinear Systems

In this chapter, we study the distributed average tracking of physical double-integrator agents. We propose distributed algorithms with reduced requirement on velocity measurements to drive the agents to track the average of these input signals. Two tracking algorithms are introduced, where each algorithm has its own advantages.
3.1 Distributed Average Tracking for Double-Integrator Multi-Agent Systems with Reduced Requirement on Velocity Measurements

3.1.1 Problem statement

Here we consider \( n \) physical agents described by double-integrator dynamics

\[
\dot{x}_i(t) = v_i(t), \quad \dot{v}_i(t) = u_i(t), \quad i = 1, \ldots, n,
\]

where \( x_i(t) \in \mathbb{R}^p \) and \( v_i(t) \in \mathbb{R}^p \) are, respectively, agent \( i \)'s position and velocity, and \( u_i(t) \) is its control input. Let \( x(t) = [x^T_1(t), \ldots, x^T_n(t)]^T \) and \( v(t) = [v^T_1(t), \ldots, v^T_n(t)]^T \).

\[\text{Assumption 18} \quad \text{The undirected graph } G \text{ is connected.}\]

\[\text{Lemma 19} \quad [67] \text{ Under Assumption 18, the Laplacian matrix } L \text{ has a simple zero eigenvalue such that } 0 = \lambda_1(L) < \lambda_2(L) \leq \ldots \leq \lambda_n(L), \text{ where } \lambda_i(\cdot) \text{ denotes the } i\text{th eigenvalue. Furthermore, for any vector } y \in \mathbb{R}^n \text{ satisfying } 1_n^T y = 0, \lambda_2(L)y^T y \leq y^T Ly \leq \lambda_n(L)y^T y.\]

Suppose that each agent has a time-varying input signal \( r_i(t) \in \mathbb{R}^p, i = 1, \ldots, n, \) satisfying

\[
\dot{r}_i(t) = v^r_i(t), \quad \dot{v}^r_i(t) = a^r_i(t),
\]

where \( v^r_i(t) \in \mathbb{R}^p \) and \( a^r_i(t) \in \mathbb{R}^p \) are, respectively, \( i \)th agent’s input velocity and input acceleration.

Here, we assume that the input signals are generated internally by the agents and each agent has access to its own input signal, input velocity, and input acceleration. Define \( x(t) = [x_1^T, \ldots, x_n^T]^T, v(t) = [v_1^T, \ldots, v_n^T]^T, r(t) = [r_1^T, \ldots, r_n^T]^T, v^r(t) = [v_1^{rT}, \ldots, v_n^{rT}]^T, \) and \( a^r(t) = [a_1^{rT}, \ldots, a_n^{rT}]^T.\)

We study the distributed average tracking problem for the double-integrator agents. The goal is to design \( u_i(t) \) for agent \( i, i = 1, \ldots, n, \) to track the average of the input signals and input
velocities, i.e.,

\[
\lim_{t \to \infty} \| x_i(t) - \frac{1}{n} \sum_{j=1}^{n} r_j(t) \|_2 = 0_p,
\]

\[
\lim_{t \to \infty} \| v_i(t) - \frac{1}{n} \sum_{j=1}^{n} v_j^r(t) \|_2 = 0_p,
\]

(3.3)

where each agent has access to its own input information and has only local interaction with its neighbors. We are interested in controller design with reduced requirement on velocity measurements. First, we consider the scenario that each agent has communication capabilities but without the need for either absolute or relative velocity measurements. The agents achieve distributed average tracking for a wide class of input signals with bounded deviations among the input signals, among the input velocities, and among the input accelerations in the absence of velocity measurements and correct position and velocity initialization. Second, we consider the scenario that each agent has sensing but not necessarily communication capabilities without the need for relative velocity measurements. The agents achieve distributed average tracking in the absence of relative velocity measurements and correct position and velocity initialization.

### 3.1.2 Velocity Free Distributed Average Tracking in the Absence of Correct Initialization

In this subsection, we consider the case where the agents can communicate with each other. To remove the requirement on velocity measurements, we introduce the following filter for agent \( i \)

\[
\dot{\psi}_i(t) = \kappa \left( x_i(t) - r_i(t) \right) - 2\kappa \left( \omega_i(t) + v_i^r(t) \right) + \alpha \sum_{j \in N_i} \left[ x_i(t) - x_j(t) \right] - \alpha^2 \sum_{j \in N_i} \left[ \omega_i(t) - \omega_j(t) \right]
- \gamma \sum_{j \in N_i} \text{sgn} \left[ \omega_i(t) - \omega_j(t) \right] - a_i^r(t),
\]

(3.4)
\[ \omega_i(t) = \psi_i(t) - \kappa (x_i(t) - r_i(t)) - \alpha \sum_{j \in N_i} [x_i(t) - x_j(t)], \]  
\hspace{1cm} (3.5) 

where \( \psi_i \in \mathbb{R}^p \) is an auxiliary filter variable, \( \omega_i \in \mathbb{R}^p \) is the filter output, and \( \kappa, \alpha, \text{ and } \gamma \in \mathbb{R}^+ \) are control gains to be designed.

We propose the following distributed control law for agent \( i \)

\[ u_i(t) = -\kappa (x_i(t) - r_i(t)) + \kappa (\omega_i(t) + v^r_i(t)) - \alpha \beta \sum_{j \in N_i} [x_i(t) - x_j(t)] + \alpha^2 \sum_{j \in N_i} [\omega_i(t) - \omega_j(t)] \]
\[ + \gamma \sum_{j \in N_i} \text{sgn} [\omega_i(t) - \omega_j(t)] + a^r_i(t), \]  
\hspace{1cm} (3.6) 

where \( \beta \in \mathbb{R}^+ \) is another control gain to be designed. For notational simplicity, we will remove the index \( t \) from variables in the reminder of the paper. Here, \( x_i - r_i \) and \( \sum_{j \in N_i} (x_i - x_j) \) are filtered through the first-order filter (3.5). The the relative filter output \( \omega_i - \omega_j \) with respect to neighbors is employed to substitute the absolute and relative velocity measurements, \( v_i \) and \( \sum_{j \in N_i} (v_i - v_j) \). The term \( \sum_{j \in N_i} (x_i - x_j) \) is employed in (3.6) to drive the agents’ positions to consensus. The terms \( x_i - r_i, \omega_i + v^r_i, \) and \( a^r_i \) in (3.6) are used to guarantee that \( \sum_{j=1}^n x_j \to \sum_{j=1}^n r_j \) and \( \sum_{j=1}^n v_j \to \sum_{j=1}^n v^r_j \) in the absence of correct position and velocity initialization. The term \( \sum_{j \in N_i} \text{sgn} (\omega_i - \omega_j) \) is employed in (3.6) to determine how much effort and in which direction agent \( i \) should adopt to guarantee consensus in the presence of \( r_i, v^r_i, \) and \( a^r_i \). The terms in (3.4) are designed accordingly to carry out the proof as shown later. To implement (3.4)-(3.6), each agent needs the relative position with respect to its neighbors, its own and neighbors’ filter outputs as well as its own input signal, input velocity, and input acceleration. There is no need for either absolute or relative velocity measurements. Because each agent needs its neighbors’ filter outputs, which cannot be measured by the agent itself, communication is necessary between neighbors.
**Assumption 20** The deviations among input signals, among input velocities, and among input accelerations are bounded, i.e., \( \sup_{t \in [0, \infty)} \| r_i(t) - r_j(t) \|_2 \leq \bar{r}_d \), \( \sup_{t \in [0, \infty)} \| v^r_i(t) - v^r_j(t) \|_2 \leq \bar{v}^r_d \), \( \sup_{t \in [0, \infty)} \| a^r_i(t) - a^r_j(t) \|_2 \leq \bar{a}^r_d \), \( i, j = 1, \ldots, n \), where \( \bar{r}_d, \bar{v}^r_d, \bar{a}^r_d \in \mathbb{R}^+ \).

**Theorem 21** Using the control law given by (3.4)-(3.6) for system (3.1), distributed average tracking is achieved asymptotically, provided that Assumptions 18 and 20 hold and the gains \( \kappa, \beta, \alpha, \) and \( \gamma \) are chosen such that \( \kappa > 0, \beta > 1, \alpha > \frac{1}{\lambda_2(L)} + \max\left\{ \frac{\lambda_{\max}(L)}{\lambda_2(L)}, \frac{3\lambda_{\max}(L) - 1}{2\kappa \lambda_2(L)} \right\} \), and \( \gamma > n(n - 1)(\kappa \bar{r}_d + \kappa \bar{v}^r_d + \bar{a}^r_d) \), where \( \lambda_2(L) \) and \( \lambda_n(L) \) are defined in Lemma 19.

**Proof:** The proof contains two steps. First, we prove for each agent \( x_i \to \frac{1}{n} \sum_{j=1}^{n} x_j \) and \( v_i \to \frac{1}{n} \sum_{j=1}^{n} v_j \). Then by proving that \( \sum_{j=1}^{n} x_j \to \sum_{j=1}^{n} r_j \) and \( \sum_{j=1}^{n} v_j \to \sum_{j=1}^{n} v^r_j \), it can be concluded that \( x_i \to \frac{1}{n} \sum_{j=1}^{n} r_j \) and \( v_i \to \frac{1}{n} \sum_{j=1}^{n} v^r_j \), \( i = 1, \ldots, n \), and hence distributed average tracking is achieved.

Now, we prove the first step. Using the control law (3.6) for (3.1), we can get

\[
\dot{x}_i = v_i, \\
\dot{v}_i = -\kappa (x_i - r_i) + \kappa (\omega_i + v^r_i) - \alpha \beta \sum_{j \in N_i} [x_i - x_j] + \alpha^2 \sum_{j \in N_i} [\omega_i - \omega_j] + \gamma \sum_{j \in N_i} \text{sgn}(\omega_i - \omega_j) + a^r_i. \tag{3.7}
\]

Due to the existence of the signum function in the algorithm, the closed-loop dynamics is discontinuous. Thus, the solutions should be investigated in terms of differential inclusions by using nonsmooth analysis [21, 26]. Since the signum function is measurable and locally essentially bounded, it follows from Lemma 3 that the Filippov solutions for the closed-loop dynamics (3.7) always exist and are absolutely continuous.
Define \( \xi_x = [M \otimes I_p]x, \xi_v = [M \otimes I_p]v, \xi_\omega = [M \otimes I_p]\omega \) and \( \xi = \begin{bmatrix} \xi_x^T & \xi_v^T & \xi_\omega^T \end{bmatrix}^T \), where \( M = I_n - \frac{1}{n}1_n1_n^T \) and \( \omega = [\omega_1^T, \ldots, \omega_n^T]^T \). It follows from (3.4), (3.5) and (3.7) that

\[
\dot{\xi} \in a.e. \mathcal{K}[f](\xi) \tag{3.8}
\]

where \( a.e. \) stands for “almost everywhere”, \( f = \begin{bmatrix} f_x^T & f_v^T & f_\omega^T \end{bmatrix}^T \) and

\[
f_x(\xi) = \xi_v, \quad \quad f_v(\xi) = -\kappa \xi_x + \kappa (M \otimes I_p)r + \kappa \xi_\omega + \kappa (M \otimes I_p)v^r - \alpha \beta (L \otimes I_p)\xi_x + \alpha^2 (L \otimes I_p)\xi_\omega \\
+ \gamma (D \otimes I_p) \text{sgn} [(D^T \otimes I_p)\xi_\omega] + (M \otimes I_p) a^r, \quad \quad f_\omega(\xi) = \kappa \xi_x - \kappa (M \otimes I_p)r - 2\kappa \xi_\omega - \kappa (M \otimes I_p)v^r + \alpha (L \otimes I_p)\xi_x - \alpha^2 (L \otimes I_p)\xi_\omega \\
- \kappa (D \otimes I_p) \text{sgn} [(D^T \otimes I_p)\xi_\omega] - (M \otimes I_p) a^r - \kappa \xi_v - \alpha (L \otimes I_p)\xi_\omega,
\]

where we have used \( M \times M = M, L \times M = L \) and \( D \times M = D \) to obtain \( f_x(\cdot), f_v(\cdot) \) and \( f_\omega(\cdot) \).

Consider the function \( V = \frac{1}{2} \xi^T \begin{bmatrix} \mu [L \otimes I_p] + \kappa I_{np} & I_{np} & I_{np} \\ I_{np} & I_{np} & I_{np} \\ I_{np} & I_{np} & \beta I_{np} \end{bmatrix} \xi, \) where \( \mu \in \mathbb{R}^+ \).

Since \( \xi_x^T 1_{np} = 0 \), using Lemma 19, we have \( V \geq \frac{1}{2} \xi^T (\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 \end{bmatrix} \otimes I_{np}) \xi \). It can be proved that if \( \mu > \frac{1}{\lambda_{2}(L)}, \kappa > 0, \) and \( \beta > 1, \) then

\[
\begin{bmatrix} \mu \lambda_2(L) + \kappa & 1 & 1 \\ 1 & 1 & \beta \end{bmatrix} - \frac{1}{\beta} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix} > 0. \text{ Thus, using Shur complement, it is concluded that } V \text{ is positive definite. Since } V \text{ is a continuous function, its set-valued Lie derivative, } \dot{V}, \text{ along (3.8) is given as}
\]

\[
\dot{V} = \mathcal{K} \left[ \mu \xi_x^T (L \otimes I_p) \xi_v^T - \kappa \xi_x^T \xi_\omega - \alpha \beta \xi_x^T (L \otimes I_p) \xi_x + \alpha \xi_x^T (L \otimes I_p) \xi_\omega + \xi_v^T \xi_v - \kappa \xi_v^T \xi_\omega \right]
\]
where \((M \otimes I_p)\xi_x = (M^2 \otimes I_p)x = \xi_x\) and using the same analysis \((M \otimes I_p)\xi_v = \xi_v\) and \((M \otimes I_p)\xi_\omega = \xi_\omega\). Also we have used the fact that \(\mathcal{K}\left[\xi_\omega^T(D \otimes I_p)\text{sgn}\left[(D^T \otimes I_p)\xi_\omega\right]\right] = \mathcal{K}\left[\|(D^T \otimes I_p)\xi_\omega\|_1\right]\) = \(\left\{\|(D^T \otimes I_p)\xi_\omega\|_1\right\}\), since \(\|(D^T \otimes I_p)\xi_\omega\|_1\) is continuous during the whole time. We can analyze the term \(\xi_\omega^T(M \otimes I_p)(\kappa r + \kappa v^r + a^r)\) as

\[
\xi_\omega^T(M \otimes I_p)(\kappa r + \kappa v^r + a^r) = \xi_\omega^T(M^2 \otimes I_p)(\kappa r + \kappa v^r + a^r)
\]

\[
\leq \|(M \otimes I_p)\xi_v\|_2\|(M \otimes I_p)(\kappa r + \kappa v^r + a^r)\|_2
\]

\[
\leq (\kappa \bar{r}_d + \kappa \bar{v}_d + \bar{a}_d) \sum_{i=1}^{n} \sum_{j=1,j \neq i}^{n} \|\xi_{wi} - \xi_{wj}\|_2
\]

\[
\leq (\kappa \bar{r}_d + \kappa \bar{v}_d + \bar{a}_d) \sum_{i=1}^{n} \max_{j=1,j \neq i} \{ \sum_{j=1,j \neq i}^{n} \|\xi_{wi} - \xi_{wj}\|_2 \}
\]

\[
\leq n(\kappa \bar{r}_d + \kappa \bar{v}_d + \bar{a}_d) \max_{j=1,j \neq i} \{ \sum_{j=1,j \neq i}^{n} \|\xi_{wi} - \xi_{wj}\|_2 \}
\]
\[ \leq \frac{n(n-1)}{2} (\kappa \bar{v}_d^T + \kappa \tilde{v}_d^T + \bar{a}_d^T) \sum_{i=1}^{n} \sum_{j \in N_i} \| \xi_{wi} - \xi_{wj} \|_2 \]

\[ \leq \frac{n(n-1)}{2} (\kappa \bar{v}_d^T + \kappa \tilde{v}_d^T + \bar{a}_d^T) \sum_{i=1}^{n} \sum_{j \in N_i} \| \xi_{wi} - \xi_{wj} \|_1, \quad (3.10) \]

where we have used Assumption 20 and \( \| \cdot \|_2 \leq \| \cdot \|_1 \) to obtain the second and the last inequalities, respectively. Let \( \mu = \alpha \beta \). Therefore, \( \alpha > \frac{1}{\lambda_2(L)} \) and \( \beta > 1 \) ensures that \( \mu > \frac{1}{\lambda_2(L)} \). Note that \( \dot{V} \) is a singleton and the function \( V \) is continuously differentiable.

Combining (3.9) and (3.10), it follows from \( \dot{V} \leq \bar{V} \), where \( \bar{V} \) denotes the derivative of \( V \), that

\[
\begin{align*}
\dot{V} & \leq \kappa (\beta - 2) \xi^T \xi_{\omega} - \alpha (\beta - 1) \lambda_2(L) \xi^T \xi_x + \left( 1 - \kappa - \alpha \lambda_2(L) \right) \xi^T \xi_{\mu} + \xi^T \left( 1 - \kappa - \kappa \beta \right) I_{np} \\
& \quad - \alpha \beta (L \otimes I_p) \xi_{\omega} + \frac{n(n-1)(\beta - 1)(\kappa \bar{v}_d^T + \kappa \tilde{v}_d^T + \bar{a}_d^T)}{2} \sum_{i=1}^{n} \sum_{j \in N_i} \| \xi_{wi} - \xi_{wj} \|_1 \\
& \quad + \left( \kappa - 2 \kappa \beta + \alpha^2 (1 - \beta) \lambda_2(L) \right) \xi^T \xi_{\omega} - \gamma (\beta - 1) \sum_{i=1}^{n} \sum_{j \in N_i} \xi^T \xi_{\omega} \text{sgn}(\xi_{wi} - \xi_{wj}) \\
& = \kappa (\beta - 2) \xi^T \xi_{\omega} - \alpha (\beta - 1) \lambda_2(L) \xi^T \xi_x + \left( 1 - \kappa - \alpha \lambda_2(L) \right) \xi^T \xi_{\mu} + \xi^T \left( 1 - \kappa - \kappa \beta \right) I_{np} \\
& \quad - \alpha \beta (L \otimes I_p) \xi_{\omega} + \frac{n(n-1)(\beta - 1)(\kappa \bar{v}_d^T + \kappa \tilde{v}_d^T + \bar{a}_d^T)}{2} \sum_{i=1}^{n} \sum_{j \in N_i} \| \xi_{wi} - \xi_{wj} \|_1 \\
& \quad + \left( \kappa - 2 \kappa \beta + \alpha^2 (1 - \beta) \lambda_2(L) \right) \xi^T \xi_{\omega} - \frac{\gamma (\beta - 1)}{2} \sum_{i=1}^{n} \sum_{j \in N_i} (\xi_{wi} - \xi_{wj})^T \text{sgn}(\xi_{wi} - \xi_{wj}),
\end{align*}
\]

where we have used \( \xi^T (D \otimes I_p) \text{sgn}[(D^T \otimes I_p) \xi_{\omega}] = \sum_{i=1}^{n} \sum_{j \in N_i} \xi^T \xi_{\omega} \text{sgn}(\xi_{wi} - \xi_{wj}) \) and Lemma 19 to obtain the first inequality. The gain \( \beta \) should satisfy the inequality \( \beta > 1 \); however, to make the proof easier, here we choose \( \beta = 2 \) without loss of generality. Thus, if \( \gamma > n(n-1)(\kappa \bar{v}_d + \kappa \tilde{v}_d + \bar{a}_d) \), we will have

\[ \dot{V} \leq \xi^T Q \xi, \quad (3.11) \]

where \( Q \triangleq [Q_{ij}], i, j = 1, \ldots, 3, Q_{ij} \in \mathbb{R}^{np \times np}, Q_{11} = -\alpha \lambda_2(L) I_{np}, Q_{22} = [1 - \kappa - \alpha \lambda_2(L)] I_{np}, \]

55
\[ Q_{23} = Q_{32} = \frac{1-3\kappa}{2}I_{np} - \alpha[L \otimes I_p], \quad Q_{33} = -[3\kappa + \alpha^2\lambda_2(L)]I_{np} \] and \( Q_{12} = Q_{13} = Q_{21} = Q_{31} = 0_{np}. \) Therefore, if \( \alpha \) satisfies the condition \( \alpha > \frac{1}{\lambda_2(L)} + \max\{\frac{\lambda^2_{\text{max}}(L)}{\lambda_2(L)}, \frac{3\lambda_{\text{max}}(L)-1}{\lambda_2(L)}, \frac{1}{12\kappa \lambda_2(L)}\}, \) it can be seen that \( \dot{V} \) is negative definite. Using Theorem 3.1 in [68], it is concluded that \( \xi = 0_{3np} \) is globally asymptotically stable, which means \( x_i \to \frac{1}{n} \sum_{j=1}^{n} x_j, \ v_i \to \frac{1}{n} \sum_{j=1}^{n} v_j, \) and \( \omega_i \to \frac{1}{n} \sum_{j=1}^{n} \omega_j, \) \( i = 1, \ldots, n. \)

Now, it is proved that \( \sum_{j=1}^{n} x_j \to \sum_{j=1}^{n} r_j \) and \( \sum_{j=1}^{n} v_j \to \sum_{j=1}^{n} v^r_j. \) Define the variables \( S_x = \sum_{j=1}^{n} (x_j - r_j), \ S_v = \sum_{j=1}^{n} (v_j - v^r_j), \) and \( S_\omega = \sum_{j=1}^{n} (\omega_j + v^r_j), \) we can get from (3.4), (3.5), and (3.7) that

\[
\dot{S} = \begin{bmatrix}
0 & 1 & 0 \\
\kappa & -\kappa & \kappa \\
-\kappa & 0 & \kappa
\end{bmatrix} \otimes I_p) S = (A \otimes I_p) S,
\]

where \( S = [S^T_x, S^T_v, S^T_\omega]^T. \) If \( \kappa > 0, \) the matrix \( A \) is Hurwitz. Therefore, \( \lim_{t \to \infty} S = 0_p, \) which means \( \lim_{t \to \infty} \sum_{j=1}^{n} x_j = \sum_{j=1}^{n} r_j \) and \( \lim_{t \to \infty} \sum_{j=1}^{n} v_j = \sum_{j=1}^{n} v^r_j. \) Combining the two parts shows that \( x_i \to \frac{1}{n} \sum_{j=1}^{n} r_j \) and \( v_i \to \frac{1}{n} \sum_{j=1}^{n} v^r_j \) asymptotically. Therefore, distributed average tracking is achieved asymptotically.

**Remark 22** Compared with [14], using the algorithm defined by (3.4)-(3.6), the requirement for both absolute and relative velocity measurements is removed, which could lower the cost of the system and save space by completely removing the velocity sensors. In addition, the algorithm in [14] relies on correct initialization. While, the introduced algorithm (3.4)-(3.6) is robust to initialization errors.

**Assumption 23** The deviations among the input accelerations are bounded, i.e., \( \sup_{t \in [0, \infty)} \|a^r_i(t) - a^r_j(t)\|_2 \leq \bar{a}^r_d, \ i, j = 1, \ldots, n, \) where \( \bar{a}^r_d \in \mathbb{R}^+. \)
Assumption 24 $\sum_{j=1}^{n} x_j(0) = \sum_{i=1}^{n} r_j(0)$ and $\sum_{j=1}^{n} v_j(0) = \sum_{j=1}^{n} v_r^j(0)$.

Corollary 25 Suppose that each agent’s position and velocity are initialized such that Assumption 24 holds. If the filter dynamics (3.4)-(3.5) and the control input (3.6) are modified as

$$
\dot{\psi}_i = \alpha \sum_{j \in N_i} [x_i - x_j] - \alpha^2 \sum_{j \in N_i} [\omega_i - \omega_j] - \gamma \sum_{j \in N_i} \text{sgn} [\omega_i - \omega_j] - a^*_i,
$$

$$\omega_i = \psi_i - \alpha \sum_{j \in N_i} [x_i - x_j],
$$

$$u_i = -\alpha \beta \sum_{j \in N_i} [x_i - x_j] + \alpha^2 \sum_{j \in N_i} [\omega_i - \omega_j] + \gamma \sum_{j \in N_i} \text{sgn} [\omega_i - \omega_j] + a^*_i,$$

(3.12)

the distributed average tracking can be achieved for a larger group of input signals that satisfy Assumption 23 provided Assumption 18 holds and the gains $\alpha$, $\gamma$ and $\beta$ are chosen such that $\beta > 1$, $\alpha > \max\{1, \frac{1}{\lambda_2(L)} + \frac{\beta^2 \lambda_n(L) + 1}{\lambda_2(L) \beta - 1}\}$, and $\gamma > n^2(n - 1) \bar{a}_r^i$, where $\lambda_2(L)$ and $\lambda_n(L)$ are defined in Lemma 19.

Proof: Similar to the proof of Theorem 21, the proof contains two steps. In the first step, the Lypanov function is replaced with

$$V = \frac{1}{2} \xi^T \begin{bmatrix} \mu [L \otimes I_p] & I_{np} & I_{np} \\ I_{np} & I_{np} & I_{np} \\ I_{np} & I_{np} & \beta I_{np} \end{bmatrix} \xi,$$

where $\mu > \frac{1}{\lambda_2(L)}$.

In the second step, by using the control law (3.12) for (3.1), the derivative of $\sum_{j=1}^{n} x_j$ and $\sum_{j=1}^{n} v_j$ can be calculated as $\sum_{j=1}^{n} \dot{x}_j = \sum_{j=1}^{n} v_j$ and $\sum_{j=1}^{n} \dot{v}_j = \sum_{j=1}^{n} a^*_j$, where we have used the fact that the graph $G$ is undirected. Therefore, it can be proved that $\sum_{j=1}^{n} x_j = \sum_{j=1}^{n} r_j$ and $\sum_{j=1}^{n} v_j = \sum_{j=1}^{n} v_r^j$ provided Assumption 24 holds. Distributed average tracking can be concluded by combining the two parts.

Remark 26 Here, the discontinuous signum function, which switches instantaneously at 0, is used to compensate for input signals, input velocities and input accelerations for each agent so that the

---

1A special choice is $x_j(0) = r_j(0)$ and $v_j(0) = v_r^j(0), j = 1, \ldots, n$.

2For example, each agent first achieves its own initial input signal and input velocity before the mission starts.
agents can reach consensus. However, by employing the signum function in the control law, there
might be chattering in each agent’s states in practice. To reduce the effects of chattering caused
by the discontinuous control actions, many analytical design methods have been proposed. The
boundary layer approach is one of the most cited approaches among the existing methods [23, 79],
which approximates the switching function by a continuous (or smooth) function such as $\|z\|_2 + \epsilon e^{-\phi t}$,
$\text{sat}(\frac{z}{\epsilon})$ or $\tanh(\frac{z}{\epsilon})$, where $z \in \mathbb{R}^n$, $\epsilon$ and $\phi \in \mathbb{R}^+$. In our simulation, we have used the function
$h(z) = \frac{z}{\|z\|_2 + \epsilon e^{-\phi t}}$ as a smooth approximation of the signum function. The size of the boundary
layer, $\epsilon e^{-\phi t}$ is time-varying and as $t \to \infty$ the continuous function $h(z)$ approaches the signum
function. In the following corollary, it is proved that by replacing the signum function with the
continuous function $h(\cdot)$ the distributed average tracking can still be achieved with zero error.

**Corollary 27** With $\text{sgn}(\cdot)$ in (3.4)-(3.6) replaced with $h(\cdot)$ defined in Remark 26, by employing
(3.4)-(3.6) for system (3.1), distributed average tracking is achieved asymptotically, provided that
Assumptions 18 and 20 hold and the gains $\kappa$, $\beta$, $\alpha$, and $\gamma$ are chosen as in Theorem 21.

Proof: If the signum function is replaced with the function $h(\cdot)$, the Lie derivative of
the Lyapunov function is replaced with it derivative and the proof steps before equation (3.9) for
Theorem 21 remain unchanged. However, since then we will have

$$
\dot{V} \leq \kappa(\beta - 2)\xi^T_\omega \xi_\omega - \alpha(\beta - 1)\lambda_2(L)\xi^T_\omega \xi_x + \left(1 - \kappa - \alpha\lambda_2(L)\right)\xi^T_\omega \xi^T_\nu + \xi^T_\nu \left(1 - \kappa - \kappa\beta\right)I_{np}
$$

$$
- \alpha\beta(L \otimes I_p)\xi_\omega + \frac{n(n - 1)(\beta - 1)(\kappa\bar{r}_d + \kappa\bar{u}_d + \bar{a}_r)}{2} \sum_{i=1}^{n} \sum_{j \in N_i} \|\xi_{\omega i} - \xi_{\omega j}\|_2
$$

$$
+ \left(\kappa - 2\kappa\beta + \alpha^2(1 - \beta)\lambda_2(L)\right)\xi^T_\omega \xi_\omega - \gamma(\beta - 1) \sum_{i=1}^{n} \sum_{j \in N_i} \xi^T_{\omega i} h(\xi_{\omega i} - \xi_{\omega j})
$$

$$
= \kappa(\beta - 2)\xi^T_\omega \xi_\omega - \alpha(\beta - 1)\lambda_2(L)\xi^T_\omega \xi_x + \left(1 - \kappa - \alpha\lambda_2(L)\right)\xi^T_\omega \xi^T_\nu + \xi^T_\nu \left(1 - \kappa - \kappa\beta\right)I_{np}
$$

$$
- \alpha\beta(L \otimes I_p)\xi_\omega + \frac{n(n - 1)(\beta - 1)(\kappa\bar{r}_d + \kappa\bar{u}_d + \bar{a}_r)}{2} \sum_{i=1}^{n} \sum_{j \in N_i} \|\xi_{\omega i} - \xi_{\omega j}\|_2
$$

58
\[ + \left( \kappa - 2\kappa\beta + \alpha^2 (1 - \beta)\lambda_2(L) \right) \xi_\omega^T \xi_\omega - \frac{\gamma(\beta - 1)}{2} \sum_{i=1}^{n} \sum_{j \in N_i} (\xi_{\omega i} - \xi_{\omega j})^T h(\xi_{\omega i} - \xi_{\omega j}), \]

(3.13)

where we have used the property that \( h(p - q) = -h(q - p) \). By replacing the definition of the function \( h(\cdot) \) in (3.13) and doing some mathematical manipulation, we can get that

\[ \dot{V} \leq \xi^T Q \xi + \frac{n(n - 1)(\beta - 1)(\kappa\bar{r}_d + \kappa\bar{v}_d + \bar{a}_d)}{2} \sum_{i=1}^{n} \sum_{j \in N_i} \epsilon e^{-\phi t} \left\| \xi_{\omega i} - \xi_{\omega j} \right\|_2 + \epsilon e^{-\phi t} \]

\[ \leq \xi^T Q \xi + \frac{n(n - 1)(\beta - 1)(\kappa\bar{r}_d + \kappa\bar{v}_d + \bar{a}_d)}{2} \sum_{i=1}^{n} \sum_{j \in N_i} \epsilon e^{-\phi t}, \]

where \( Q \) is defined in the proof of Theorem 21. Define \( Q' = -Q \) and \( \lambda_M \) as the maximum eigenvalue of the matrix

\[ \frac{1}{2} \begin{bmatrix} \mu [L \otimes I_p] + \kappa I_{np} & I_{np} & I_{np} \\ I_{np} & I_{np} & I_{np} \\ I_{np} & I_{np} & \beta I_{np} \end{bmatrix}. \]

Therefore, \( V \leq \lambda_M \xi^T Q \xi \). Since \( Q \) is a negative definite matrix, it is concluded that \( \xi^T Q \xi = -\xi^T Q' \xi \leq -\lambda_{\min}(Q') \xi^T \xi \leq -\frac{\lambda_{\min}(Q')}{\lambda_M} V. \)

Thus, by knowing that \( e^{-\phi t} \rightarrow 0 \) as \( t \rightarrow \infty \), Lemma 2.19 in [59] implies that the closed-loop dynamics (8) is exponentially stable. The rest of the proof is the same as Theorem 21.

3.1.3 Distributed Average Tracking in the Absence of Relative Velocity Information and Communication

In the previous section, the proposed algorithm solves the distributed average tracking problem in the presence of communication without velocity measurements. In this section, we deal with the distributed average tracking problem without communication in the absence of the relative velocity information and correct initialization constraint.

**Assumption 28** Each input signal, its velocity and acceleration are bounded, i.e., \( \sup_{t \in [0, \infty]} \| r_i(t) \|_2 \leq \bar{r}, \) \( \sup_{t \in [0, \infty]} \| v_i^r(t) \|_2 \leq \bar{v}^r \) and \( \sup_{t \in [0, \infty]} \| a_i^r(t) \|_2 \leq \bar{a}^r, \) \( i = 1, \ldots, n, \) where \( \bar{r}, \bar{v}^r, \bar{a}^r \in \mathbb{R}^+. \)
We introduce the following filter for agent $i$

$$
\dot{\psi}_i = \alpha \sum_{j \in N_i} (x_i - x_j) - \alpha^2 \omega_i - \kappa r_i - \kappa v^r_i - a^r_i - \gamma \sgn(\omega_i), \quad (3.14)
$$

$$
\omega_i = \psi_i - \alpha \sum_{j \in N_i} (x_i - x_j), \quad (3.15)
$$

where $\psi_i \in \mathbb{R}^p$ is an auxiliary filter variable, $\omega_i \in \mathbb{R}^p$ is the filter output, and $\kappa, \alpha, \gamma \in \mathbb{R}^+$ are control gains.

We propose the following control input for agent $i$

$$
u_i = -\kappa (x_i - r_i) - \kappa (v_i - v^r_i) - \alpha \beta \sum_{j \in N_i} (x_i - x_j) + \alpha^2 \omega_i + \gamma \sgn(\omega_i) + a^r_i, \quad (3.16)$$

where $\beta \in \mathbb{R}^+$ is another control gain to be designed. In (3.16), the terms $\kappa (x_i - r_i)$, $\kappa (v_i - v^r_i)$ and $a^r_i$ are used to drive the two errors $\sum_{j=1}^n (x_j - r_j)$ and $\sum_{j=1}^n (v_j - v^r_j)$ to zero because the correct initialization condition does not hold. The term $\sgn(\omega_i)$ is employed to guarantee consensus in the presence of $r_i$, $v^r_i$ and $a^r_i$ in each agent’s closed-loop dynamics. To implement (3.14)-(3.16), each agent needs its own position, velocity, and relative position between itself and neighbors, and its own input signal, input velocity and input acceleration. There is no need for relative velocity measurements between neighbors. Note that here the relative positions are the only information related to neighbors, which can be obtained by using only local sensing, and hence communication between neighbors is not necessary. The following theorem presents sufficient conditions to solve distributed average tracking without neighbors’ velocity and communication.

**Theorem 29** Distributed average tracking is achieved for system (3.1) using (3.14)-(3.16) asymptotically, under Assumptions 18 and 28 and provided that the gains $\kappa$, $\beta$, $\alpha$, and $\gamma$ are chosen such that $\kappa > 0$, $\beta > \sqrt{n}$, $\gamma > \frac{1+\beta}{\beta-\sqrt{n}} (\kappa \bar{r} + \kappa \bar{v}^p + \bar{a}^r)$, $\alpha > \frac{1}{\lambda_2(L)} + \max \left\{ \frac{\beta^2 \lambda_{\max}^2(L)}{4(\beta-1)\lambda_2(L)}, \frac{\beta \lambda_{\max}(L)(\kappa-1)}{2\kappa}, \frac{2\kappa^2}{\lambda_2(L)} \right\}$, where $\lambda_2(L)$ and $\lambda_n(L)$ are defined in Lemma 19.
Proof: Similar to Section 3.1.2, the proof contains two steps. In the first step, it is proved that 
\[ x_i \to \frac{1}{n} \sum_{j=1}^{n} x_j, \quad v_i \to \frac{1}{n} \sum_{j=1}^{n} v_j, \quad \text{and} \quad \omega_i \to 0 \] asymptotically. Again due to the existence of the signum function in the algorithm, the closed-loop dynamics is discontinuous. Thus, the solutions should be investigated in terms of differential inclusions by using nonsmooth analysis.

Define \( \xi, \xi_v, M, \) and \( \omega \) as in the proof of Theorem 21. Let \( \xi' = \begin{bmatrix} \xi_x^T & \xi_v^T & \omega^T \end{bmatrix}^T \) and \( f' = \begin{bmatrix} f_x^T & f_v^T & f_\omega^T \end{bmatrix}^T \).

By rewriting the dynamics (3.1) using (3.16) in vector form, it follows from (3.14) and (3.15) that

\[ \dot{\xi}' \in \mathcal{K}[f'](\xi') \]  

(3.17)

where

\[
\begin{align*}
  f_x' &= \xi_v, \\
  f_v' &= -\kappa \xi_x + \kappa (M \otimes I_p)r - \kappa \xi_v + \kappa (M \otimes I_p)v^r - \alpha \beta [L \otimes I_p] \xi_x + \alpha^2 (M \otimes I_p)\omega \\
  &\quad + \gamma (M \otimes I_p) \text{sgn}(\omega) + (M \otimes I_p) a^r, \\
  f_\omega' &= \alpha [L \otimes I_p] \xi_x - \alpha^2 \omega - \kappa r - \kappa v^r - a^r - \gamma \text{sgn}(\omega) - \alpha [L \otimes I_p] \xi_v.
\end{align*}
\]

Consider the following function

\[
V = \frac{1}{2} \xi'^T \begin{bmatrix} \mu [L \otimes I_p] + 2 \kappa I_{np} & I_{np} & I_{np} \\
I_{np} & I_{np} & I_{np} \\
I_{np} & I_{np} & \beta I_{np} \end{bmatrix} \xi',
\]

where \( \mu \in \mathbb{R}^+ \). Using the same analysis as in Section 3.1.2, if \( \mu > \frac{1}{\chi_2(L)}, \kappa > 0, \) and \( \beta > 1 \), the function \( V \) is positive definite. Since \( V \) is a continuous function, its set-valued Lie derivative along (3.17) is given as

\[
\dot{V} = \mathcal{K}[\mu \xi_x^T [L \otimes I_p] \xi_v - \kappa \xi_x^T \xi_x - \alpha \beta \xi_v^T [L \otimes I_p] \xi_x + \alpha \xi_v^T [L \otimes I_p] \xi_x + \xi_v^T \xi_v - \kappa \xi_v^T \xi_v,
\]

61
where we used the fact that $K = \beta > 1$ Combining (3.18) and (3.19), if $V \leq -\varepsilon V_\dot{} = \dot{\beta}_{\dot{\gamma}} = T\left\{ (1 + \omega + \) \right. \\
\beta \gamma T r - \beta \gamma T v^r - \beta\omega T a^\gamma - $ $\left. \beta \omega T [L \otimes I_p]k_x \right\},$ (3.18)
\begin{align*}
\kappa \omega T(M \otimes I_p)r + \kappa \omega T(M \otimes I_p)v^r + \omega T(M \otimes I_p)a^\gamma + \gamma \omega T(M \otimes I_p)sgn(\omega) - \beta \kappa \omega T r - \beta \kappa \omega T v^r \\
- \beta \omega T a^\gamma - \beta \gamma T sgn(\omega) \\
= \omega T \left( (M - \beta I_n) \otimes I_p \right) \left[ k_r + \kappa v^r + a^\gamma + \gamma sgn(\omega) \right] \\
\leq \left[ (1 + \beta)(k_r + \kappa v^r + a^\gamma) + \gamma \sqrt{n} \right] \cdot \|\omega\|_2 - \beta \gamma \|\omega\|_1 \\
\leq \left[ (1 + \beta)(k_r + \kappa v^r + a^\gamma) + \gamma \sqrt{n} \right] \cdot \|\omega\|_1 - \beta \gamma \|\omega\|_1. \\
(3.19)
\end{align*}
Combining (3.18) and (3.19), if $\beta > \sqrt{n}$, $\gamma > \frac{1 + \beta}{\beta - \sqrt{n}}(k_r + \kappa v^r + a^\gamma)$ and $\mu = \alpha \beta$, it follows from $\dot{V} \in \dot{V}$ that
\begin{align*}
\dot{V} \leq & -\kappa \xi_x^T \xi_x - \alpha(\beta - 1)\lambda_2(L)\xi_x^T \xi_x + \xi_v^T \xi_v - \alpha \lambda_2(L)\xi_v^T \xi_v + \omega^T \xi_v + \kappa \omega^T \xi_v - \kappa \omega^T \xi_x - \kappa \omega^T \xi_v \\
+ & \alpha^2 \omega^T(M \otimes I_p)\omega - \alpha^2 \beta \omega^T \omega - \alpha \beta \omega^T [L \otimes I_p]k_v \\
= & \left[ \begin{array}{ccc}
\xi_x^T & \xi_v^T & \omega^T
\end{array} \right] P \left[ \begin{array}{c}
\xi_x \\
\xi_v \\
\omega
\end{array} \right], \\
(3.20)
\end{align*}
where \( P \triangleq [P_{ij}], \ i,j = 1, \ldots, 3, \ P_{ij} \in \mathbb{R}^{np \times np}, \ P_{11} = -[\kappa + \alpha(\beta - 1)\lambda_2(L)]I_{np}, \ P_{22} = [1 - \kappa - \alpha\lambda_2(L)]I_{np}, \ P_{12} = P_{21} = 0_{np}, P_{13} = P_{31} = -\frac{1}{2}\kappa I_{np}, \ P_{23} = P_{32} = \frac{1}{2}[(1 - \kappa)I_{np} - \alpha\beta L], \) and \( P_{33} = -\alpha^2(\beta - 1)I_{np} \) and we have used the fact that \( M < I_n \) and Lemma 19 to obtain the inequality. If the control gains \( \kappa, \alpha, \) and \( \beta \) satisfy the constraints mentioned in Theorem 29, the matrix \( P \) and thus \( \dot{V} \) are negative definite. Similar to the proof of Theorem 21, by using Theorem 3.1 in [68], it is concluded that \( S_x, S_v \) is globally asymptotically stable, which implies that \( x_i - \frac{1}{n} \sum_{j=1}^{n} x_j \to 0, v_i - \frac{1}{n} \sum_{j=1}^{n} v_j \to 0 \) and \( \omega_i \to 0 \) as \( t \to \infty. \)

In the second step, we prove that \( \sum_{j=1}^{n} x_j \to \sum_{j=1}^{n} r_j \) and \( \sum_{j=1}^{n} v_j \to \sum_{j=1}^{n} v^r_j \) asymptotically. Define \( S_x, S_v \) as in the proof of Theorem 21. Then, we can get from (3.2) and (3.16) that

\[
\begin{bmatrix}
\xi_x \\
\xi_v \\
\omega
\end{bmatrix} = 0_{3np},
\]

(3.21)

We then use input-to-state stability to analyze the system (3.21) by treating the term \( \alpha^2 \sum_{i=1}^{n} \omega_i + \gamma \sum_{i=1}^{n} \text{sgn}(\omega_i) \) as the input and \( S_x \) and \( S_v \) as the states. If \( \kappa > 0 \), the matrix

\[
\begin{bmatrix}
0_p & I_p \\
-\kappa I_p & -\kappa I_p
\end{bmatrix}
\]

is Hurwitz. The system (3.21) with zero input is exponentially stable and hence input-to-state stable. Since \( \omega_i \to 0 \) as \( t \to \infty \) for each agent, it follows that \( S_x \to 0 \) and \( S_v \to 0 \), which implies that \( \sum_{j=1}^{n} x_j \to \sum_{j=1}^{n} r_j \) and \( \sum_{j=1}^{n} v_j \to \sum_{j=1}^{n} v^r_j \), respectively. Employing the result of first step, it is concluded that \( x_i \to \frac{1}{n} \sum_{j=1}^{n} r_j \) and \( v_i \to \frac{1}{n} \sum_{j=1}^{n} v^r_j \) and the distributed average tracking is achieved.
**Remark 30** Using the algorithm defined by (3.14)-(3.16), there is no need for relative velocity measurements, communication and correct initialization. Therefore, the algorithm (3.14)-(3.16) can rely on only local sensing that lowers the cost of the system by removing the communication devices and the system is robust to initialization errors.

**Remark 31** Both algorithms in Sections 3.1.2 and 3.1.3 have their unique features and advantages while with trade-offs. Depending on the application scenarios, one might be more appropriate than the other. The advantages of the algorithm (3.4)-(3.6) are that it can deal with more general input signals, and both absolute and relative velocity measurements are removed. However, the trade-off of the algorithm is that it necessitates communication between neighbors. Here, if the agents’ positions and velocities satisfy an initialization condition, the algorithm was modified to deal with a even larger group of input signals. It should be noted that, this assumption is not very restrictive for some applications. In fact, the setup of the distributed average tracking problem is that each agent knows its own input signal and input velocity accurately. Even for the simpler static distributed averaging problem (computing the average of constant values), each agent still needs to know its own quantity to be averaged accurately. Otherwise, it is hard to imagine how the problem can even be solved. Here, correct initialization means that before a mission starts, each agent adjusts its initial position and velocity to achieve the initial conditions of its input signal and input velocity. The advantages of the algorithm (3.14)-(3.16) are that the needs for relative velocity measurements and communication are removed. However, the trade-off of this algorithm is that it still requires absolute velocity measurements and is limited to bounded input signals with bounded velocities and accelerations.
Remark 32 While the lower bounds of the control gains in both algorithms in Sections 3.1.2 and 3.1.3 depend on global information, the bounds are used to show the existence of such control gains. If the control gains are chosen large enough, the distributed average tracking can be achieved by both algorithms. Despite the dependence of the lower bounds on some global information, the gains themselves are constant and can be determined off-line before a mission starts. Hence both algorithms are implementable. One conservative approach is to select large enough gains based on the worst case scenario of the graphs (e.g., considering the finite number of all possible connected graphs and checking the eigenvalues of the Laplacian matrices in all cases). Also to obtain a better estimate of the lower bounds of the gains, it is possible to run an average consensus algorithm to estimate $n$ and a max consensus algorithm to figure out the largest bounds of $r_i$, $v_i^r$ and $a_i^r$, respectively.

Corollary 33 With $\text{sgn}(\cdot)$ in (3.14)-(3.16) replaced with $h(\cdot)$ defined in Remark 26, by employing (3.14)-(3.16) for system (3.1), distributed average tracking is achieved asymptotically, provided that Assumptions 18 and 28 hold and the gains $\kappa$, $\beta$, $\alpha$, and $\gamma$ are chosen as in Theorem 29. Proof: The proof is very similar to Corollary 27 and is omitted here.

Remark 34 The introduced algorithm in Section 3.1.3 is still valid for the case of a strongly connected weight balanced directed graph. However, in order for the introduced algorithm in Section 3.1.2 to be valid for a strongly connected weight balanced directed graph, the term $\sum_{j \in N_i} \text{sgn}(\omega_i - \omega_j)$ in (3.4)-(3.6) should be replaced with $\text{sgn}(\sum_{j \in N_i} (\omega_i - \omega_j))$. In the proof, $L$ can be replaced with the symmetric matrix $L + L^T$ as $x^T L x = \frac{1}{2} x^T (L + L^T) x$. Since the graph $G$ is strongly connected weight balanced, $L + L^T$ is positive semidefinite with a simple zero eigenvalue. Note that applying the introduced algorithms for directed graphs, we need to redefine $\lambda_2$ as the smallest...
nonzero eigenvalue of $L + L^T$.

### 3.1.4 Simulation

In this section, simulation results are given to illustrate the effectiveness of the theoretical results obtained in Sections 3.1.2 and 3.1.3. Assume that there are ten agents ($n = 10$), where the network topology is an undirected ring. In the first case, the input acceleration for agent $i$ is given by $0.1 \times [i \sin(5t) + \text{mod}(t, 2), i \cos(5t) + \text{mod}(t, 2)]^T$, and the initial position and velocity of the agents are chosen as $x(0) = [0.1 \times b, -0.15 \times b]^T$ and $v(0) = [0.15 \times b, 0.1 \times b]^T$, $r_i(0) = [0.2 \times b, -0.4 \times b]^T$ and $v^r_i(0) = [0.2 \times i, 0.3 \times i]^T$, $i = 1, \ldots, 10$, where $b = [-4, -3, -2, -1, 0, 1, 2, 3, 4, 5]^T$. Denote the $j$th component of $x_i$ as $x_{ij}$. Similar notations are used for $v_i$, $r_i$, and $v_i^r$. The control parameters for all agents are chosen as $\kappa = 1$, $\beta = 2$, $\alpha = 10$, and $\gamma = 5$. We simulate the algorithm defined by (3.4)-(3.6). Fig. 3.1 shows the positions of the agents and the average of the input signals. Fig. 3.2 shows the velocities of the agents and the average of the input velocities. Clearly, all agents’ positions and velocities track, respectively, the average of the input signals and input velocities in the absence of velocity measurements.
Figure 3.1: The positions of 10 agents and the average of the input signals, where the states are initialized correctly. The initial average of input signals is represented as a dot while the initial positions of the agents are represented as squares.
Figure 3.2: The velocities of 10 agents and the average of input velocities, where the agents’ velocity are initialized correctly.

In the second case, we simulate distributed average tracking when the agents’ positions and velocities are not initialized correctly. The input acceleration for each agent is described by $0.1i \times [\sin(t), \cos(t)]^T$. We simulate the algorithm defined by (3.14)-(3.16). The initial values are set the same as previous algorithm and the control parameters are chosen for all agents as $\kappa = 2$, $\alpha = 80$, $\gamma = 1$ and $\beta = 10$. Fig. 3.3 and Fig. 3.4 show that the distributed average tracking is achieved for both agents’ positions and velocities in the absence of neighbors’ velocity information and accurate initialization.
Figure 3.3: The positions of 10 agents and the average of input signals in the absence of correct initialization. The initial average of input signals is represented as a dot while the initial positions of the agents are represented as squares.
3.2 Distributed Average Tracking of Physical Second-order Agents

Without Constraint on Input Signals

In this subsection, I study distributed average tracking of physical second-order agents with linear and heterogeneous nonlinear dynamics, where there is no constraint on input signals.

Figure 3.4: The velocities of 10 agents and the average of input velocities in the absence of correct initialization.
3.2.1 Problem Statement

Consider a multi-agent system consisting of \( n \) physical agents described by the following heterogeneous nonlinear second-order dynamics

\[
\dot{x}_i(t) = v_i(t),
\]
\[
\dot{v}_i(t) = f_i(x_i(t), v_i(t), t) + u_i(t), \quad i = 1, \ldots, n,
\]

(3.22)

where \( x_i(t) \in \mathbb{R}^p \), \( v_i(t) \in \mathbb{R}^p \) and \( u_i(t) \in \mathbb{R}^p \) are \( i \)th agent’s position, velocity and control input, respectively and \( f_i : \mathbb{R}^p \times \mathbb{R}^p \times \mathbb{R}^+ \rightarrow \mathbb{R}^p \) is a vector-valued nonlinear function which will be defined later. As it was mentioned, there are applications, where the physical agents should track the average of a group of time varying signals. While, the physical agents and time varying signals might be described by more complicated dynamics rather than the single- or double-integrator dynamics. Here, we investigate the distributed average tracking for a more general group of agents and input signals.

**Lemma 35** For any vector \( x \in \mathbb{R}^n \), we have

\[
x^T LDW \text{sgn}(D^T x) \geq \lambda_2(L)x^T DW \text{sgn}(D^T x),
\]

(3.23)

where \( W \) is a positive definite diagonal matrix.

*Proof*: If \( D^T x = 0_n \), the equality holds. However, if \( D^T x \neq 0_n \), then we replace \( L \) by \( DD^T \) in (3.23). Thus, we will have

\[
x^T LDW \text{sgn}(D^T x) = x^T DD^T DW \text{sgn}(D^T x) = (D^T x)^T D^T DW \text{sgn}(D^T x).
\]

Note that both \( D^T D \) and \( DD^T \) have the same set of nonzero eigenvalues \( s = \{\lambda_2, \ldots, \lambda_n\} \) [85]. Suppose that \( S \) is the space spanned by the eigenvectors belong to the nonzero eigenvalues of \( D^T D \).
If $D^T x \in S$, there is a $\lambda_i(D^T D) \in \mathbb{R}$ such that $(D^T x)^T D^T D = \lambda_i(D^T D)(D^T x)^T$. Thus, we get that

$$(D^T x)^T D^T D W \text{sgn}(D^T x) = \lambda_i(D^T D)(D^T x)^T W \text{sgn}(D^T x) \geq \lambda_2(D^T D)(D^T x)^T W \text{sgn}(D^T x) = \lambda_2(D^T D)x^T DW \text{sgn}(D^T x).$$

If $D^T x \notin S$, it follows that $D^T x$ belongs to the null space of $D^T D$. Based on Lemma 3 in [13], the null space of the incidence matrix $D$ coincides with null space of $D^T D$. Thus, $D(D^T x) = 0_n$ which means $Lx = 0_n$. It follows that $x$ belongs to the space spanned by vector $1_n$ and hence $D^T x = 0_n$. This contradicts with $D^T x \neq 0_n$ and hence $D^T x \in S$.

3.2.2 Distributed Average Tracking for Physical Heterogeneous Nonlinear Second-order Agents

In this subsection, I study the distributed average tracking problem for a group of heterogeneous nonlinear second-order agents, where the nonlinear term $f_i(\cdot, \cdot, t)$ satisfies the Lipschitz-like condition and there is no constraint on the input signal.

**Assumption 36** The vector-valued function $f_i(\cdot, \cdot, t)$ is continuous in $t$ and satisfies the following Lipschitz-like condition $\forall t \geq 0$

$$
\begin{align*}
\|f_i(x, v, t) - f_i(y, z, t)\|_1 & \leq \rho_1\|x - y\|_1 + \rho_2\|v - z\|_1 + \rho_3, \\
\|f_i(0_p, 0_p, t)\|_1 & \leq \rho_4,
\end{align*}
$$

where $x, v, y, z \in \mathbb{R}^p$, and $\rho_1, \rho_2, \rho_3, \rho_4 \in \mathbb{R}^+$.  

**Remark 37** Note that Assumption 36 is more general than the Lipschitz-type condition, satisfied by many well-known systems such as the pendulum system with a control torque, car-like robots, the
Chua’s circuit, the Lorenz system, and the Chen system [46]. In fact, the term \( f_i(\cdot, \cdot, t) \) is general enough to represent both the nonlinear dynamics and possible bounded disturbances.

It should be noted that the nonlinear term \( f_i(\cdot, \cdot, t) \) is unknown and the input acceleration is arbitrary and can be unbounded. Therefore, a novel filter is introduced with state-dependent time varying gains to estimate the average of the input signals and input velocities in the presence of these challenges. Consider the following local filter for agent \( i \)

\[
p_i = z_i + r_i,
\]

\[
\ddot{z}_i = -\kappa(p_i - r_i) - \kappa(q_i - v_i^r) - \beta \sum_{j=1}^{n} a_{ij}(\psi_i + \psi_j) \text{sgn}((p_i + q_i) - (p_j + q_j)), \tag{3.24}
\]

where \( p_i, q_i \in \mathbb{R}^p \) are the filter outputs, \( q_i = \dot{p}_i, z_i \in \mathbb{R}^p \) is an auxiliary filter variable, \( \psi_i = \|r_i\|_1 + \|v_i^r\|_1 + \|a_i^r\|_1 + \gamma \), and \( \beta, \gamma \in \mathbb{R}^+ \) will be designed later.

The control input \( u_i \) is designed as

\[
u_i = -\eta(\|x_i\|_1 + \|v_i\|_1 + \|a_i^r\|_1 + \gamma)\{\ddot{x}_i + \ddot{v}_i + \text{sgn}(\ddot{x}_i + \ddot{v}_i)\} + \ddot{z}_i, \tag{3.25}
\]

where \( \ddot{x}_i = x_i - p_i, \ddot{v}_i = v_i - q_i \), and \( \eta_i \) is a varying gain with \( \eta_i(0) \geq 0 \). Let \( \psi_i' = \|x_i\|_1 + \|v_i\|_1 + \|a_i^r\|_1 + \gamma \). It should be noted that under Assumption 36, the unknown term \( f_i(\cdot, \cdot, t) \) might be unbounded. Further, the input acceleration \( a_i^r \) is arbitrary and hence might be unbounded. Thus, the state-dependent time varying gains \( \psi_i \) and \( \psi_i' \) are employed in the filter’s dynamics and the control input to overcome these unboundedness challenges.

**Theorem 38** Under the control algorithm given by (3.24)-(3.25) for system (3.22), the distributed average tracking goal (3.3) is achieved asymptotically, provided that Assumptions 18 and 36 hold and the control gains \( \beta > \kappa, \kappa > \max\left\{1, \frac{\lambda^2_{\max}(L)}{2\lambda^2(L)}\right\}, \gamma > \rho_3 + \rho_4, \eta > \max\{1, \rho_1, \rho_2\} \) \( i = 1, \ldots, n \).
Proof: The proof contains two steps. First, it is proved that \( \lim_{t \to \infty} p_i = \frac{1}{n} \sum_{j=1}^{n} r_j \) and \( \lim_{t \to \infty} q_i = \frac{1}{n} \sum_{j=1}^{n} v_j \).

Using \( q_i = \dot{p}_i \), the local filter’s dynamics (3.24) can be rewritten as

\[
\begin{align*}
\dot{p}_i &= q_i, \\
\dot{q}_i &= - \kappa (p_i - r_i) - \kappa (q_i - v^r_i) - \beta \sum_{j=1}^{n} a_{ij} (\psi_i + \psi_j) \text{sgn} [(p_i + q_i) - (p_j + q_j)] + a^r_i,
\end{align*}
\]

(3.26)

Due to the existence of the signum function in the algorithm, the closed-loop dynamics is discontinuous. Thus, the solutions should be investigated in terms of differential inclusions by using non-smooth analysis [21, 26]. Since the signum function is measurable and locally essentially bounded, the Filippov solutions for the closed-loop dynamics (3.26) always exist and is absolutely continuous, Lemma 3. Let \( r(t) = [r_1^T, \ldots, r_n^T]^T \), \( v^r(t) = [v_1^r T, \ldots, v_n^r T]^T \), \( a^r(t) = [a_1^r T, \ldots, a_n^r T]^T \), \( p = [p_1^T, \ldots, p_n^T]^T \), \( q = [q_1^T, \ldots, q_n^T]^T \), \( \bar{p}_i = p_i - \sum_{j=1}^{n} p_j \) and \( \bar{q}_i = q_i - \sum_{j=1}^{n} q_j \). Defining \( M = I_n - \frac{1}{n} 1_n^T 1_n \), we get that

\[
\begin{bmatrix}
\dot{\bar{p}} \\
\dot{\bar{q}}
\end{bmatrix} \in^{a.e.} K[f](\bar{p}, \bar{q})
\]

(3.27)

where \( a.e. \) stands for “almost everywhere”, \( \bar{p} = (M \otimes I_p)p \), \( \bar{q} = (M \otimes I_p)q \) and \( f = \begin{bmatrix} f^T_p & f^T_q \end{bmatrix}^T \)

and

\[
\begin{align*}
\tilde{f}_p &= \bar{q}, \\
\tilde{f}_q &= - \kappa \bar{p} + \kappa (M \otimes I_p)v - \kappa \bar{q} + \kappa (M \otimes I_p)v^r - \beta (DW \otimes I_p)\text{sgn}[(D^T \otimes I_p)(\bar{p} + \bar{q})] \\
&\quad + (M \otimes I_p)a^r,
\end{align*}
\]

where \( W \) is a diagonal matrix and \( MD = D \). The \( k \)th diagonal element of matrix \( W \) describes the \( k \)th edge that is between node \( i \) and node \( j \) and it equals to \( (\psi_i + \psi_j) \).
Consider the following Lyapunov function candidate

\[ V_1 = \frac{1}{2} \begin{bmatrix} \tilde{p}^T & \tilde{q}^T \end{bmatrix} \left( L \otimes \begin{bmatrix} 2\kappa I_p & I_p \\ I_p & I_p \end{bmatrix} \right) \begin{bmatrix} \tilde{p} \\ \tilde{q} \end{bmatrix}. \] (3.28)

Since \((1_n \otimes I_p)^T \tilde{p} = 0_{np}\) and \((1_n \otimes I_p)^T \tilde{q} = 0_{np}\), by using Lemma 19, we will have

\[ V_1 \geq \frac{1}{2} \begin{bmatrix} \tilde{p}^T & \tilde{q}^T \end{bmatrix} \begin{bmatrix} 2\kappa \lambda_2(L)I_{np} & L \otimes I_{np} \\ L \otimes I_{np} & \lambda_2(L)I_{np} \end{bmatrix} \begin{bmatrix} \tilde{p} \\ \tilde{q} \end{bmatrix}. \]

It can be proved that if \(\kappa > \frac{\lambda_{2,\text{max}}(L)}{2\lambda_2(L)}\), then \(\begin{bmatrix} 2\kappa \lambda_2(L)I_{np} & L \otimes I_{np} \\ L \otimes I_{np} & \lambda_2(L)I_{np} \end{bmatrix}\) is positive definite and hence \(V_1\) is positive definite. Since \(V_1\) is a continuous function, its set-valued Lie derivative along (3.27) is given as

\[ \dot{V}_1 = \mathcal{K} \left[ 2\kappa \tilde{p}^T (L \otimes I_p) \tilde{q} + \tilde{q}^T (L \otimes I_p) \tilde{q} - \kappa \tilde{p}^T (L \otimes I_p) \tilde{p} + \kappa \tilde{p}^T (L \otimes I_p) r - \kappa \tilde{p}^T (L \otimes I_p) \tilde{q} 
\] + \kappa \tilde{p}^T (L \otimes I_p) v^r - \beta \tilde{p}^T (LDW \otimes I_p) \text{sgn}[(D^T \otimes I_p)(\tilde{p} + \tilde{q})] + \tilde{p}^T (L \otimes I_p)a^r 
\] - \kappa \tilde{q}^T (L \otimes I_p) \tilde{p} + \kappa \tilde{q}^T (L \otimes I_p) r - \kappa \tilde{q}^T (L \otimes I_p) \tilde{q} + \kappa \tilde{q}^T (L \otimes I_p) v^r 
\] - \beta \tilde{q}^T (LDW \otimes I_p) \text{sgn}[(D^T \otimes I_p)(\tilde{p} + \tilde{q})] + \tilde{q}^T (L \otimes I_p)a^r \]

\[ = - \kappa \tilde{p}^T (L \otimes I_p) \tilde{p} - (\kappa - 1) \tilde{q}^T (L \otimes I_p) \tilde{q} - \beta (\tilde{p}^T + \tilde{q}^T)(LDW \otimes I_p) \mathcal{K} \left[ \text{sgn}[(D^T \otimes I_p)(\tilde{p} + \tilde{q})] \right] \]
\[ + (\tilde{p}^T + \tilde{q}^T)(L \otimes I_p)(kr + \kappa v^r + a^r). \] (3.29)

If \((D^T \otimes I_p)(\tilde{p} + \tilde{q}) = 0_{np}\), then \((L \otimes I_p)(\tilde{p} + \tilde{q}) = 0_{np}\). Therefore, it is concluded that \(-\beta (\tilde{p}^T + \tilde{q}^T)(LDW \otimes I_p) \mathcal{K} \left[ \text{sgn}[(D^T \otimes I_p)(\tilde{p} + \tilde{q})] \right] = 0\). If \((D^T \otimes I_p)(\tilde{p} + \tilde{q}) \neq 0_{np}\), then \(\mathcal{K} \left[ \text{sgn}[(D^T \otimes I_p)(\tilde{p} + \tilde{q})] \right] \) is a singleton and the function \(V_1\) is continuously differentiable. By using Lemma 35, it follows from
\( \dot{V}_1 \in \dot{V}_1 \), where \( \dot{V}_1 \) denotes the derivative of \( V_1 \), that

\[
\dot{V}_1 \leq - \kappa \ddot{p}^T (L \otimes I_p) \dot{p} - (\kappa - 1) \ddot{q}^T (L \otimes I_p) \dot{q} - \beta \lambda_2(L) (\ddot{p}^T + \ddot{q}^T) (DW \otimes I_p) \text{sgn}[(DT \otimes I_p)(\dot{p} + \dot{q})]
\]

\[
+ (\ddot{p}^T + \ddot{q}^T) (L \otimes I_p) (\kappa r + \kappa v^r + a^r)
\]

\[
= - \kappa \ddot{p}^T (L \otimes I_p) \dot{p} - (\kappa - 1) \ddot{q}^T (L \otimes I_p) \dot{q}
\]

\[
- \beta \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} (\psi_i + \psi_j) \left[ (\dot{p}_i + \dot{q}_i) - (\dot{p}_j + \dot{q}_j) \right]^T \text{sgn} \left[ (\dot{p}_i + \dot{q}_i) - (\dot{p}_j + \dot{q}_j) \right]
\]

\[
+ (\ddot{p}^T + \ddot{q}^T) (L \otimes I_p) (\kappa r + \kappa v^r + a^r)
\]

\[
= - \kappa \ddot{p}^T (L \otimes I_p) \dot{p} - (\kappa - 1) \ddot{q}^T (L \otimes I_p) \dot{q}
\]

\[
- \beta \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} \left\| \dot{p}_i + \dot{q}_i - (\dot{p}_j + \dot{q}_j) \right\|_1
\]

\[
+ \sum_{i=1}^{n} \left[ \sum_{j=1}^{n} a_{ij} \right] (\dot{p}_i + \dot{q}_i) - (\dot{p}_j + \dot{q}_j) \right\|_1 (\kappa \|r_i\|_1 + \kappa \|v_i^r\|_1 + \|a_i^r\|_1)
\]

\[
= - \kappa \ddot{p}^T (L \otimes I_p) \dot{p} - (\kappa - 1) \ddot{q}^T (L \otimes I_p) \dot{q}
\]

\[
+ \sum_{i=1}^{n} \left( \kappa \|r_i\|_1 + \kappa \|v_i^r\|_1 + \|a_i^r\|_1 - \beta \psi_i \right) \sum_{j=1}^{n} a_{ij} \left\| (\dot{p}_i + \dot{q}_i) - (\dot{p}_j + \dot{q}_j) \right\|_1
\]

\[
- \beta \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} \psi_j \left\| (\dot{p}_i + \dot{q}_i) - (\dot{p}_j + \dot{q}_j) \right\|_1.
\]

Since \( \psi_i = \|r_i\|_1 + \|v_i^r\|_1 + \|a_i^r\|_1 + \gamma \) and \( \beta > \kappa > 1 \), we will have

\[
\dot{V}_1 \leq - \kappa \ddot{p}^T (L \otimes I_p) \dot{p} - (\kappa - 1) \ddot{q}^T (L \otimes I_p) \dot{q}
\]

\[
\leq - \kappa \lambda_2(L) \ddot{p}^T \dot{p} - (\kappa - 1) \lambda_2(L) \ddot{q}^T \dot{q} < 0,
\]

(3.30)
where we have used $\kappa > 1$ and Lemma 19 and the fact that $(1_n \otimes I_p)^T \tilde{p} = 0_{np}$ and $(1_n \otimes I_p)^T \tilde{q} = 0_{np}$ to obtain the second inequality. Integrating both sides of (3.30), we can obtain $\tilde{p}, \tilde{q} \in L_2$. Therefore, by using Theorem 4.10 in [41], it is concluded that

$$
\begin{bmatrix}
\tilde{p} \\
\tilde{q}
\end{bmatrix}
= 0_{2np}
$$

is globally exponentially stable which means

$$
\lim_{t \to \infty} p_i = \frac{1}{n} \sum_{j=1}^{n} p_j \text{ and } \lim_{t \to \infty} q_i = \frac{1}{n} \sum_{j=1}^{n} q_j \text{ for } i = 1, \ldots, n.
$$

Now, it is proved that

$$
\sum_{j=1}^{n} (x_j - r_j) \text{ and } S_q = \sum_{j=1}^{n} (v_j - v_j^r),
$$

we can get from (3.26) that

$$
\begin{bmatrix}
S_p \\
S_q
\end{bmatrix}
= (A \otimes I_p)
\begin{bmatrix}
S_p \\
S_q
\end{bmatrix}.
$$

If $\kappa > 0$, the matrix $A$ is Hurwitz. Therefore,

$$
\lim_{t \to \infty}
\begin{bmatrix}
S_p \\
S_q
\end{bmatrix}
= 0_p,
$$

which means

$$
\lim_{t \to \infty} \sum_{j=1}^{n} p_j = \sum_{j=1}^{n} r_j
$$

and

$$
\lim_{t \to \infty} \sum_{j=1}^{n} q_j = \sum_{j=1}^{n} v_j^r.
$$

Combining the two parts shows that $p_i \to \frac{1}{n} \sum_{j=1}^{n} r_j$ and $q_i \to \frac{1}{n} \sum_{j=1}^{n} v_j^r$ asymptotically.

Second, it is proved that by using the control law (3.25) for (3.22), $\lim_{t \to \infty} x_i = p_i$ and

$$
\lim_{t \to \infty} v_i = q_i
$$

in parallel and hence it can be concluded that $\lim_{t \to \infty} x_i = \frac{1}{n} \sum_{j=1}^{n} r_j$ and $\lim_{t \to \infty} v_i = \frac{1}{n} \sum_{j=1}^{n} v_j^r$. Define $\bar{x} = [\bar{x}_1^T, \ldots, \bar{x}_n^T]^T$, $\bar{v} = [\bar{v}_1^T, \ldots, \bar{v}_n^T]^T$, and $f(x, v, t) = [f_1^T(x_1, v_1, t), \ldots, f_n^T(x_n, v_n, t)]^T$.

Using the control law (3.25) for (3.22), we get the closed-loop dynamics in vector form as

$$
\dot{\bar{x}} = \bar{v},
$$

(3.31)

$$
\dot{\bar{v}} = f(x, v, t) - \eta(\bar{x} + \bar{v}) - \eta(p') \otimes I_p \text{sgn}(\bar{x} + \bar{v}) - a^r,
$$

(3.32)
where $\psi' \triangleq \text{diag}(\psi'_1, \ldots, \psi'_n)$. Consider the Lyapunov function candidate

$$V_2 = \frac{1}{2} \begin{bmatrix} \bar{x}^T & \bar{v}^T \end{bmatrix} \begin{bmatrix} 2\eta & 1 \\ 1 & 1 \end{bmatrix} \otimes I_{np} \begin{bmatrix} \bar{x} \\ \bar{v} \end{bmatrix}. \quad (3.33)$$

It is easy to see that $V_2$ is positive definite. By taking the set-valued Lie derivative of $V_2$, $\dot{V}_2$, along the Filippov set-valued map of (3.31), we will have

$$\dot{V}_2 = \mathcal{K} \left[ 2\eta \bar{x}^T \bar{v} + \bar{v}^T \bar{v} + \bar{x}^T f(x, v, t) - \eta \bar{x}^T (\bar{x} + \bar{v}) - \eta \bar{v}^T (\psi' \otimes I_p) \text{sgn}(\bar{x} + \bar{v}) - \bar{x}^T a^r \right.$$

$$+ \bar{v}^T f(x, v, t) - \eta \bar{v}^T (\bar{x} + \bar{v}) - \eta \bar{v}^T (\psi' \otimes I_p) \text{sgn}(\bar{x} + \bar{v}) - \bar{v}^T a^r \bigg]$$

$$= \left\{ - \eta \bar{x}^T \bar{x} - (\eta - 1) \bar{v}^T \bar{v} + (\bar{x}^T + \bar{v}^T) f(x, v, t) - \eta (\bar{x}^T + \bar{v}^T) (\psi' \otimes I_p) \text{sgn}(\bar{x} + \bar{v}) \right.$$

$$- (\bar{x}^T + \bar{v}^T) a^r \right\},$$

where we have used the fact that $\mathcal{K} \left[ (\bar{x}^T + \bar{v}^T) (\psi' \otimes I_p) \text{sgn}(\bar{x} + \bar{v}) \right] = \mathcal{K} \left[ \sum_{i=1}^n \psi'_i (\bar{x}_i + \bar{v}_i)^T \text{sgn}(\bar{x}_i + \bar{v}_i) \right] = \mathcal{K} \left[ \sum_{i=1}^n \psi'_i \|\bar{x}_i + \bar{v}_i\|_1 \right] = \left\{ \sum_{i=1}^n \psi'_i \|\bar{x}_i + \bar{v}_i\|_1 \right\}$. Note that the set-valued Lie derivative of $V_2$ is a singleton and the function $V_2$ is continuously differentiable. It follows from $\dot{V}_2 \in \dot{V}_2$, where $\dot{V}_2$ denotes the derivative of $V_2$, that

$$\dot{V}_2 = - \eta \bar{x}^T \bar{x} - (\eta - 1) \bar{v}^T \bar{v} + \sum_{i=1}^n (\bar{x}_i + \bar{v}_i)^T f_i(x_i, v_i, t) - \eta \sum_{i=1}^n \psi'_i \|\bar{x}_i + \bar{v}_i\|_1 - \sum_{i=1}^n (\bar{x}_i + \bar{v}_i)^T a_i^r$$

$$= - \eta \bar{x}^T \bar{x} - (\eta - 1) \bar{v}^T \bar{v} + \sum_{i=1}^n (\bar{x}_i + \bar{v}_i)^T f_i(x_i, v_i, t) - f_i(0_p, 0_p, t)) + \sum_{i=1}^n (\bar{x}_i + \bar{v}_i)^T f_i(0_p, 0_p, t)$$

$$- \eta \sum_{i=1}^n (\|x_i\|_1 + \|v_i\|_1 + a_i^r_1 + \gamma) \|\bar{x}_i + \bar{v}_i\|_1 - \sum_{i=1}^n (\bar{x}_i + \bar{v}_i)^T a_i^r$$

$$\leq - \eta \bar{x}^T \bar{x} - (\eta - 1) \bar{v}^T \bar{v}$$

$$- \eta \sum_{i=1}^n (\|x_i\|_1 + \|v_i\|_1 + a_i^r_1 + \gamma) \|\bar{x}_i + \bar{v}_i\|_1$$

$$+ \sum_{i=1}^n (\rho_1 \|x_i\|_1 + \rho_2 \|v_i\|_1 + \rho_3 + \rho_4 + \|a_i^r_1\|_1) \|\bar{x}_i + \bar{v}_i\|_1$$

78
\[ -\eta \dot{x}^T \ddot{x} - (\eta - 1) \dot{v}^T \ddot{v} < 0, \]

where we have used Assumption 36 to obtain first equality and \( \eta > \max\{1, \rho_1, \rho_2\} \) and \( \gamma > \rho_3 + \rho_4 \) to obtain last inequality. Therefore, by using Theorem 4.10 in [41], it is concluded that \( \begin{bmatrix} \ddot{x} \\ \ddot{v} \end{bmatrix} = 0_{2np} \) is globally exponentially stable. Thus, by combining the two steps, it is concluded that for agent \( i \),

\[
\lim_{t \to \infty} x_i = \frac{1}{n} \sum_{j=1}^{n} r_j \quad \text{and} \quad \lim_{t \to \infty} v_i = \frac{1}{n} \sum_{j=1}^{n} v_j^r.
\]

**Remark 39** As it can be seen, by using algorithm (3.24)-(3.25), each agent can achieve the distributed average tracking, where there is no constraint on the input signals and the nonlinear terms in agents’ dynamics are unknown and heterogeneous. Due to the presence of the unknown term \( f(\cdot, \cdot, t) \) in the agents’ dynamics, the proposed algorithms for double-integrator agents are not applicable to achieve the distributed average tracking. For example, by employing the algorithm in [14] for (3.22), the two equalities \( \sum_{j=1}^{n} x_j = \sum_{j=1}^{n} r_j \) and \( \sum_{j=1}^{n} v_j = \sum_{j=1}^{n} v_j^r \) do not hold anymore. In fact, the unknown term \( f(\cdot, \cdot, t) \) functions as a disturbance and it will not allow the average of the positions and velocities to track the input signals and the reference velocities, respectively. This shows the essence of using the local filter (3.24) in our algorithm.

**Remark 40** In the proposed algorithm (3.24)-(3.25), correct position and velocity initialization is not required, where the initialization of physical variables might not be feasible for real applications.

### 3.2.3 Distributed Average Tracking for Physical Double-Integrator Agents

In the proposed algorithm in Subsection 3.2.2, the agents are described by heterogeneous nonlinear second-order dynamics, where the nonlinear term \( f_i(\cdot, \cdot, t) \) satisfies a Lipschitz-like con-
dition. However, in some applications the agents’ dynamics can be linearized as double-integrator dynamics. Therefore, I modify the proposed algorithm (3.24)-(3.25) for double-integrator systems. Since the nonlinear term \( f_i(\cdot, \cdot, t) \) does not exist in agents’ dynamics, the local filter is not required here. We can directly design \( u_i \) to drive the agents’ positions and velocities to track the average of the input signals and reference velocities, respectively. The control input for agent \( i, i = 1, \ldots, n \), is designed as

\[
\begin{align*}
  u_i &= -\kappa(x_i - r_i) - \kappa(v_i - v_i^r) - \beta \sum_{j=1}^n a_{ij}(\psi_i + \psi_j)\text{sgn}[(x_i + v_i) - (x_j + v_j)] + a_i^r, \\
  \quad \text{(3.34)}
\end{align*}
\]

where \( \psi_i \) is defined in Subsection 3.2.2 and \( \beta, \kappa, \gamma \in \mathbb{R}^+ \) will be designed later.

**Theorem 41** Under the control input given by (3.34) for system (3.1), the distributed average tracking goal (3.3) is achieved asymptotically, provided that Assumption 18 holds and \( \beta > \kappa, \kappa > \max \left\{ 1, \frac{\lambda_2^2(L)}{2\lambda_2(L)} \right\} \), \( \gamma > \rho_3 + \rho_4 \).

**Proof:** Here the proof contains two steps. First, it is proved that for \( i \)th agent, \( \lim_{t \to \infty} x_i = \frac{1}{n} \sum_{j=1}^n x_j \) and \( \lim_{t \to \infty} v_i = \frac{1}{n} \sum_{j=1}^n v_j \). Using the control input (3.34) for (3.1), we get the closed-loop dynamics as

\[
\begin{align*}
  \dot{x}_i &= v_i, \\
  \dot{v}_i &= -\kappa(x_i - r_i) - \kappa(v_i - v_i^r) - \beta \sum_{j=1}^n a_{ij}(\psi_i + \psi_j)\text{sgn}[(x_i + v_i) - (x_j + v_j)] + a_i^r, \\
  \quad \text{(3.35)}
\end{align*}
\]

Let \( e_x = (M \otimes I_p)x \) and \( e_v = (M \otimes I_p)v \), where \( x, v \) and \( M \) are defined in Subsection 3.2.2. Then, we can rewrite (3.35) in vector form as

\[
\begin{align*}
  \dot{e}_x &= e_v, \\
  \dot{e}_v &= -\kappa e_x + \kappa(M \otimes I_p)r - \kappa e_v + \kappa(M \otimes I_p)v^r - \beta(DW \otimes I_p)\text{sgn}[(D^T \otimes I_p)(e_x + e_v)] + (M \otimes I_p)a^r, \\
  \quad \text{(3.36)}
\end{align*}
\]
where $W$ is defined in Subsection 3.2.2 and we have used $MD = D$. Consider the Lyapunov function candidate

$$V = \frac{1}{2} \begin{bmatrix} e^T_x & e^T_v \end{bmatrix} \left( L \otimes \begin{bmatrix} 2\kappa I_p & I_p \\ I_p & I_p \end{bmatrix} \right) \begin{bmatrix} e_x \\ e_v \end{bmatrix}. $$

Using the same procedure as the first part of proof of Theorem 38, it is concluded that

$$\begin{bmatrix} e_x \\ e_v \end{bmatrix} = 0_{2np}$$

is globally exponentially stable which means $\lim_{t \to \infty} x_i = \frac{1}{n} \sum_{j=1}^{n} x_j$ and $\lim_{t \to \infty} v_i = \frac{1}{n} \sum_{j=1}^{n} v_j$ for $i = 1, \ldots, n$.

In the second step, we prove that $\lim_{t \to \infty} \sum_{j=1}^{n} x_j = \sum_{j=1}^{n} r_j$ and $\lim_{t \to \infty} \sum_{j=1}^{n} v_j = \sum_{j=1}^{n} v^r_j$ asymptotically. Defining the variables $S_x = \sum_{i=1}^{n} (x_i - r_i)$ and $S_v = \sum_{i=1}^{n} (v_i - v^r_i)$, then we can get from (3.2) and (3.35) that

$$\dot{S}_x = S_v,$$

$$\dot{S}_v = -\kappa S_x - \kappa S_v. \tag{3.37}$$

If $\kappa > 0$, the system (3.37) is exponentially stable. Therefore, $\lim_{t \to \infty} \begin{bmatrix} S_x \\ S_v \end{bmatrix} = 0_p$, which means

$$\lim_{t \to \infty} \sum_{j=1}^{n} x_j = \sum_{j=1}^{n} r_j \text{ and } \lim_{t \to \infty} \sum_{j=1}^{n} v_j = \sum_{j=1}^{n} v^r_j.$$ Combining the two parts shows that $x_i \to \frac{1}{n} \sum_{j=1}^{n} r_j$ and $v_i \to \frac{1}{n} \sum_{j=1}^{n} v^r_j$ asymptotically.

**Remark 42** The introduced algorithm in [14] can achieve the distributed average tracking provided that $a^r_i$ is bounded. However, the algorithm (3.34) is more general and solves the problem regardless of any constraint on $a^r_i$.

**Corollary 43** Suppose that each agent is described by the following heterogeneous double-integrator

81
dynamics

\[ \dot{x}_i = v_i, \]
\[ m_i \dot{v}_i = u_i, \quad 0 \leq m_i \leq \bar{m}_i, \quad i = 1, \ldots, n, \]

where \( \bar{m}_i \in \mathbb{R}^+ \). Let \( \bar{m} = \max_{i=1,\ldots,n} \bar{m}_i \). If the input acceleration \( a_i' \) in (3.2) is arbitrary, by employing the local filter and the control input defined in, respectively, (3.24) and (3.25), where \( \kappa > 1 \) and \( \gamma \in \mathbb{R}^+ \), the distributed average tracking is achieved.

Proof: The proof is similar to the proof of Theorem 38, where (3.33) is replaced with

\[
V_2 = \frac{1}{2} \sum_{i=1}^{n} \begin{bmatrix} \tilde{x}_i^T & \tilde{v}_i^T \end{bmatrix} \begin{bmatrix} \frac{2n}{m_i} I_{np} & I_{np} \\ I_{np} & I_{np} \end{bmatrix} \begin{bmatrix} \tilde{x}_i \\ \tilde{v}_i \end{bmatrix},
\]

and \( \eta > \bar{m} \).

3.2.4 Simulation

In this section, numerical simulation results are given to illustrate the effectiveness of the theoretical results obtained in Subsections 3.2.2. It is assumed that there are five agents \( n = 5 \), where the network topology is described by Fig. 1.1. The nonlinear term \( f_i(\cdot, \cdot, t) \) for each agent is chosen as [70]

\[
f(x, y, t) = \begin{bmatrix} 0 \\ 0 \\ -\beta \epsilon \sin(\omega x_1) \end{bmatrix} + \begin{bmatrix} \delta(y_2 - y_1 h(y_1)) \\ y_1 - y_2 + y_3 \\ -\beta y_2 - \mu v_3 \end{bmatrix},
\]
where $\delta = 10$, $\beta = 19.53$, $\mu = 0.1636$, $\epsilon = 0.2$, $\omega = 0.5$ and $h(y_1) = -0.7831y_1 - 0.324(|y_1 + 1| - |y_1 - 1|)$. The initial values are set as

$$
x(0) = \begin{bmatrix} 0.1b \\ -0.2b \\ 0.3b \end{bmatrix}, \quad v(0) = \begin{bmatrix} 0.15b \\ 0.35b \\ -0.4b \end{bmatrix}, \quad r(0) = \begin{bmatrix} 0.3b \\ 0.1b \\ -0.15b \end{bmatrix}, \quad v_r(0) = \begin{bmatrix} -0.25b \\ 0.45b \\ 0.1b \end{bmatrix},
$$

where $b = [-2; -1; 0.5; 1.5; 2.5]$. The control parameters for all agents are chosen as $\alpha = 10$ and $\eta = 1$. We simulate the algorithm defined by (3.24)-(3.25). Fig. 3.5 shows the positions of the agents and the average of the reference inputs. Clearly, all agents have tracked the average of the reference inputs in the presence nonlinear term $f_i(\cdot, \cdot, t)$ in their dynamics. Fig. 3.6 shows the velocities of the agents and the average of the reference velocities. We see that the distributed average tracking is achieved for the agents’ velocities too.

Figure 3.5: The positions of 5 agents and the average of the reference inputs. The solid lines and the dashed lines describe, respectively the average of reference inputs and the position of agents.
Figure 3.6: The velocities of 5 agents and the average of reference inputs. The solid lines and the dashed lines describe, respectively the average of reference velocities and the velocity of agents.
Chapter 4

Conclusions

In this work, the following two distributed tracking issues in multi-agent systems were investigated in detail:

- leader-follower flocking with a moving leader for Lagrange networks,
- distributed average tracking of physical double-integrator systems with linear and nonlinear dynamics.

To solve the first issue, three novel distributed tracking algorithms were proposed for two cases: i) the leader moves with a constant velocity, and ii) the leader moves with a varying velocity. In the first case, a distributed continuous adaptive control algorithm accounting for unknown parameters was proposed in combination with a distributed continuous estimator for each follower. In the second case, two distributed discontinuous adaptive control laws and estimators were proposed to track the varying leader, where only a group of followers have access to the leader. In the first algorithm the agents used the two-hope neighbors’ information and needed some global information to determine the control gains. While, in the second algorithm, by introducing gain adaptation laws a fully
distributed algorithm was introduced for each agent, where only one-hop neighbors’ information was employed in the algorithm.

In the second part of the dissertation, two groups of distributed average tracking algorithms were introduced for, respectively, second-order linear and nonlinear multi-agent systems. First, two distributed algorithms (controller design combined with filter design) are introduced to achieve distributed average tracking with reduced requirement on velocity measurements and in the absence of correct position and velocity initialization. Second, a distributed algorithm was introduced to achieve distributed average tracking for physical second-order agents with heterogeneous nonlinear dynamics, where there was no constraint on the input signals. Due to unknown terms in agents’ dynamics and unbounded input signals, a local filter was introduced for each agent to estimate the average of the input signals and reference velocities. The novelty of the local filters was that by employing the time varying state-dependent gains, the agents can track the average of a group of input signals with no constraint on their dynamics.
Bibliography


