2181 Glenn L. Martin Hall
College Park, MD 20742-3035
TEL: 301-405-2410
http://www.enme.umd.edu

04/07/2025

Re: Previously Published Materials Appearing in Thesis or Dissertation

Dean of the Graduate School,

Sara Honarvar has:

☐ NOT INCLUDED any previously published works within their thesis or dissertation.

☒ INCLUDED one or more previously published works within their thesis or dissertation. This letter certifies that the examining committee for the student has determined that the student made a substantial contribution to the previously published work. The inclusion of the previously published work has the approval of the thesis or dissertation advisor and the Graduate Director.

Sincerely,

Peter Sandborn

Peter Sandborn
Director of Graduate Studies
Department of Mechanical Engineering
University of Maryland

Sincerely,

Dr. Yancy Diaz-Mercado
Advisor for Sara Honarvar

ABSTRACT

Title of Dissertation: LEARNING INTERACTION BEHAVIOR IN DISTRIBUTED INTELLIGENT SYSTEMS

Sara Honarvar
Doctor of Philosophy, 2025

Dissertation Directed by: Yancy Diaz-Mercado
Department of Mechanical Engineering

This dissertation develops a formal framework for learning robust, state-dependent interaction behaviors in distributed intelligent systems. While learning-based approaches are increasingly used in multi-agent systems applications such as robotics and crowd-aware navigation, they often lack theoretical guarantees of stability and robustness, particularly under state-dependent or asymmetric interaction topologies. Inspired by the adaptability seen in human collaboration, this work aims to bridge data-driven methods and control-theoretic foundations to address key challenges in systematic modeling and learning effective interaction strategies. We begin by drawing an analogy between human motor synergies and consensus algorithms, showing that human collaboration in motor tasks can be modeled via decentralized control laws. Empirical validation using static finger force matching data demonstrates that consensus algorithms over fixed undirected graphs capture key features of human coordination. Next, we introduce a geometric Graph Neural Network (GNN) model for complex human interactions in crowds. By connecting the structure of GNNs to weighted consensus dynamics and designing edge weights inspired by

psychological studies on pedestrian behavior, we show how human-inspired topological priors improve predictive performance and interpretability. The simulation results show the sensitivity of these models to the interaction topology, highlighting the need for principled edge weight design in learned models. To generalize beyond static and handcrafted graph structures, we develop a consensus framework over graphs that support directed, signed, and state-dependent edge weights. Through Lyapunov-based edge agreement analysis, we derive sufficient conditions for stability and robustness—even under nonlinear and antagonistic interactions. Building on these results, we formulate an optimal control framework to learn optimal, state-dependent interaction strategies from observations. We derive optimality conditions for recovering interaction weights, investigate data richness requirements for successful learning, and validate the effectiveness of the framework through numerical simulation. Overall, this work integrates theoretical insights with data-driven solutions, aiming to provide useful perspectives toward safer, more intuitive, and reliable cooperative distributed systems.

LEARNING INTERACTION BEHAVIOR IN DISTRIBUTED INTELLIGENT SYSTEMS

by

Sara Honarvar

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2025

Advisory Committee:
Professor Yancy Diaz-Mercado, Chair/Advisor
Professor Nikhil Chopra
Professor Hosam K. Fathy
Professor Jin-Oh Hahn
Professor Dinesh Manocha, Dean's Representative

© Copyright by
Sara Honarvar
2025

Dedication

To my beloved parents, Mehri and Abbas. “It’s your light that lights the world.”

Acknowledgments

As I look back on my PhD journey, I am filled with a deep sense of gratitude. It has been a challenging and an incredibly rewarding path—one that has shaped me in ways I could have never imagined. Along the way, I have been fortunate to meet so many brilliant, kind-hearted people who have left a lasting impact on my life.

First, I want to express my deepest gratitude to my advisor, Dr. Yancy Diaz-Mercado. Pursuing my PhD under your supervision has been one of the most meaningful experiences of my life. Your intellectual curiosity, commitment to research and teaching, and the genuine care you show for your students have continually inspired me. Thank you for supporting me through every challenge, for your patience and encouragement, and for pushing me to step out of my comfort zone. To my committee members, Dr. Dinesh Manocha, Dr. Jin-Oh Hahn, Dr. Nikhil Chopra and Dr. Hosam Fathy, thank you for your thoughtful feedback, your valuable time, and your encouragement throughout this process. Your insights have not only strengthened my work but also broadened my perspective. I am sincerely thankful to the organizations that supported my research. The Maryland Robotics Center (MRC) provided a collaborative environment and facilitated meaningful industry partnerships. I was fortunate to receive fellowships from Amazon Lab126, Microsoft Diversity in Research and AI, and MRC, each of which enriches my doctoral journey in unique ways. A heartfelt thank you to Dr. Dinesh Manocha, who supported my work since my time as an Amazon fellow and continually connected me with valuable resources to expand the impact of my research. I also would like to thank Ania Picard for her support during my Ph.D. I am especially grateful to the School of Engineering at UMD, and in particular, the

Department of Mechanical Engineering for creating a space where I felt I belonged. I'm also deeply thankful to Dr. Balakumar Balachandran for his constant support and for always listening to the students' voices, and to Dr. Samuel Graham for his commitment to building an inclusive and welcoming environment. To my labmates in CCRL, Xiaotian, Suraj, Sarjana, Andrew, and Kimia, thank you for the laughter, support, and collaboration. It's been a joy to work alongside you. And to my incredible friends, thank you for warming my heart. Your unwavering love and presence, near or far, has meant everything to me.

To my incredible family, Madar and Baba, thank you for your unconditional love and the many sacrifices you've made for me. To my siblings, Simin, Kianoush and Kamal, you have each helped shape who I am and I feel truly blessed to share this life with you. To Alireza and Samira, thank you for the joy and warmth you bring into our family. And to my sweet nephew and niece, Hirad and Rira, you light up my world in ways I never knew possible. Being your aunt is one of my greatest joys. And last but certainly not least, to my best friend, my love, and my husband, Farshad, thank you for being my home. Through every challenge and every little victory, your love and constant encouragement have been my greatest source of strength. Thank you for always making me feel special, I'm endlessly grateful to walk this journey with you.

Table of Contents

Dedication ii

Acknowledgements iii

Table of Contents v

List of Tables viii

List of Figures ix

List of Abbreviations x

Chapter 1: Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Research Questions and Objectives . . . 2

Chapter 2: Literature Review 5
- 2.1 Human Coordination as Inspiration for Multi-Agent Systems . . . 5
- 2.2 Consensus Algorithms in Multi-Agent Systems . . . 7
- 2.3 Edge Weight Design in Weighted Consensus . . . 8
- 2.4 Stability and Robustness Analysis of Weighted Consensus . . . 12
- 2.5 Learning Interaction Rules from Observation . . . 13
- 2.6 Summary . . . 14

Chapter 3: Modeling Human-Human Collaboration: A Connection Between Inter-Personal Motor Synergy and Consensus Algorithms 16
- 3.1 Overview . . . 16
- 3.2 Introduction . . . 16
- 3.3 Preliminaries . . . 19
  - 3.3.1 Inter-personal Interaction Experiment . . . 19
  - 3.3.2 Notation . . . 20
  - 3.3.3 Topology of Interaction . . . 21
  - 3.3.4 Level of Error Compensation . . . 22
  - 3.3.5 Agent Dynamics . . . 23
  - 3.3.6 State Estimation Using Kalman Filter . . . 24
  - 3.3.7 Desired Averaged Dynamics . . . 24
- 3.4 Problem Formulation . . . 26

3.5 Control Law . . . 26
3.5.1 Ensemble-Level Dynamics . . . 27
3.5.2 Task-Level Convergence Analysis . . . 28
3.5.3 UCM Convergence Analysis . . . 29
3.6 Simulation Results . . . 30
3.6.1 Choice of Parameters . . . 30
3.6.2 Model Validation . . . 32
3.7 Summary . . . 35

Chapter 4: Complex Network Interactions: Geometric Graph Neural Network Modeling of Human Interactions in Crowded Environments 36
4.1 Overview . . . 36
4.2 Introduction . . . 37
4.3 Preliminaries . . . 39
4.3.1 Notation . . . 39
4.3.2 Agent Dynamics: . . . 39
4.3.3 Graph Representation of Crowd Interaction: . . . 39
4.3.4 From State-Dependent Weighted Consensus To GNN Structure . . . 40
4.4 Problem Formulation . . . 42
4.4.1 Topology of Interaction . . . 42
4.4.2 Model Architecture: . . . 45
4.4.3 Datasets and Evaluation Metrics . . . 47
4.4.4 Training Setup: . . . 48
4.5 Results . . . 48
4.6 Summary . . . 52

Chapter 5: Stability and Robustness Conditions for Nonlinear Interaction Rules 53
5.1 Overview . . . 53
5.2 Introduction . . . 53
5.3 Preliminaries . . . 54
5.3.1 Notation . . . 55
5.3.2 Graph Theory . . . 55
5.3.3 Multi-Agent Dynamics and Interaction Representation . . . 59
5.4 Problem Formulation . . . 66
5.5 Stability and Robustness Analysis . . . 66
5.5.1 Lyapunov Analysis via Edge-Based Potential Energy . . . 67
5.5.2 Local Exponential Convergence to Desired Behavior . . . 69
5.5.3 Robustness to Bounded Disturbances . . . 73
5.6 Summary . . . 76

Chapter 6: Optimality Conditions for the Interaction Weights 77
6.1 Overview . . . 77
6.2 Introduction . . . 77
6.3 Notation . . . 79
6.4 Problem Formulation: Spatio-Temporal Optimal Control Problem . . . 79

6.5 Optimality Condition 81
6.5.1 Learning Interaction Weight Algorithm 84
6.6 Richness of data 84
6.7 Simulation Example 87
6.7.1 Simulation Setup 88
6.7.2 Simulation Results 90
6.8 Summary 90

Chapter 7: Conclusion 94

Chapter 8: Supplementary Material 96

Bibliography 109

List of Tables

3.1 Summary of RMSE Results 34
3.2 Summary of PCA results. 34
3.3 Summary of UCM results 34

4.1 Summary of ADE/FDE Results 49

6.1 Summary of Edge parameters for the simulation. 88

List of Figures

3.1 Human Interaction Experiment 20
3.2 Demonstration of experimental co-working data 21
3.3 Comparison between simulation and experiment for a representative trial. 31

4.1 A visual representation of the neighborhood of interaction 40
4.2 Illustration of our geometric GNN model 46
4.3 Visualizing the predicted distribution of pedestrian trajectories for a representative scene 50

6.1 Visualization of Simulation Network 88
6.2 Evaluation of richness of data. 89
6.3 Cost evolution during the OC learning process. 91
6.4 Comparison of learned and true control policies. 91
6.5 Comparison of learned and true Potential policies. 91
6.6 Representative trials: evolution of edge weights and corresponding agent trajectories. 92

List of Abbreviations

GNN	Graph Neural Network
IRL	Inverse Reinforcement Learning
IOC	Inverse Optimal Control
MAS	Multi-Agent Systems
ADE	Average Displacement Error
FDE	Final Displacement Error
APF	Artificial Potential Function
CNS	Central Nervous System
DOF	Degrees-of-Freedom
ORT	Orthogonal
PC	Principal Component
PCA	Principal Component Analysis
UCM	Uncontrolled Manifold
IPMS	Inter-Personal Motor Synergy
HHC	Human-Human Collaboration
STGCNN	Spatio-Temporal GCNN
AI	Artificial Intelligence
IL	imitation learning

Chapter 1: Introduction

1.1 Motivation

In recent years, multi-agent systems (MAS) have played a crucial role in shaping intelligent, distributed decision-making systems in domains including robotics [1], autonomous driving [2], human-robot (swarm) interaction [3], and crowd modeling [4]. These systems require agents to collaborate and coordinate effectively, often with limited access to global information, making coordination a nontrivial challenge. Consensus algorithms have emerged as a fundamental tool for analyzing cooperative behaviors in MAS, offering mathematical frameworks for modeling agent interactions [5]. However, achieving coordination using only local interactions presents significant challenges, especially when accounting for uncertainties in the environment, nonlinear agent dynamics, and evolving interaction patterns. [6].

A promising approach to addressing these challenges is to draw inspiration from nature. Nature provides remarkable examples of coordination from flocks of birds effortlessly adapting to environmental changes [7] to humans seamlessly navigating crowded environments [8] or performing cooperative motor tasks [9]. Biological agents demonstrate the ability to coordinate efficiently by continuously adapting their actions based on local perception and implicit communication. For instance, when two people carry a heavy object together, they instinctively balance forces without explicit dialogue—revealing a form of implicit coordination known as motor syn-

ergy [10]. Integrating such principles into MAS coordination design can substantially enhance their adaptability, efficiency, and robustness in practical, real-world scenarios. To replicate such capabilities in MAS, researchers have developed both rule-based and learning-based approaches to model interactions [11]. Rule-based models, while intuitive and grounded in theory, often fail to fully capture the complex aspects of behavior. In contrast, learning-based methods such as inverse reinforcement learning (IRL), inverse optimal control (IOC), imitation learning (IL) [12], and neural networks (GNN) [13] have proven effective at predicting cooperative behaviors, but they often lack formal guarantees of stability, robustness, or generalizability.

These limitations are especially concerning in safety-critical applications such as autonomous driving, assistive robots, and emergency response. In such settings, it is essential to not only predict agent behaviors, but also ensure that learned interaction policies lead to safe, stable, and interpretable system dynamics. Further, learning these behaviors opens new opportunities for analyzing coordination strategies—whether to improve team performance in robotics, design more efficient human-robot systems, or even analyze collaborative human activities such as sports. For instance, learning the interactions among soccer players can provide deeper insights into coaching strategies and impact team performance.

1.2 Research Questions and Objectives

This dissertation investigates the following central question: ***Under what conditions can distributed agents learn and maintain robust interaction behaviors while ensuring convergence to the desired cooperative dynamics?*** In answering this primary question, we also address several specific sub-questions: (1) How effectively can consensus-based models represent human collab-

orative behaviors? (2) How can we connect consensus models with learning-based methods such as GNNs, and how does graph topology influence both model performance and interpretability? (3) What explicit mathematical conditions ensure stability and robustness of consensus under generalized interactions—including directed, state-dependent, and mixed cooperative-repulsive interactions? (4) Given these conditions, how can we learn interaction behaviors from observations, and what are the data richness requirements to ensure identifiability?

To address these questions, this dissertation proposes a framework integrating theoretical insights from control theory, graph theory, human motor control, and data-driven learning methods. Specifically, we demonstrate that human interaction behaviors emerging in physical coordination tasks can be robustly modeled and learned through nonlinear agreement dynamics. The goal is to develop distributed systems that can extract interpretable interaction rules from spatio-temporal observations while guaranteeing robust and reliable coordination—particularly in dynamic and uncertain environments.

Dissertation Roadmap

This dissertation follows a structured approach to progressively build towards a comprehensive framework for learning robust interaction behaviors in MAS. The chapters are organized as follows: **Chapter 2** reviews foundational theories and recent advances in MAS, consensus algorithms, human-inspired collaboration, edge-weight design, and learning-based approaches. This chapter identifies significant research gaps, particularly emphasizing the underexplored research on dynamic, state-dependent, and signed weighted consensus and their implications for stability and robustness. **Chapter 3** demonstrates the application of consensus algorithms to model hu-

man collaborative behaviors. Through theoretical analysis and empirical validation using a human finger-force matching task, we validate that simple, fixed-topology consensus models (undirected, unweighted graphs) capture essential features of human collaboration, providing evidence for the broader application of consensus frameworks to human interaction modeling. **Chapter 4** extends these insights by introducing geometric GNNs, which incorporate weighted graphs to capture more complex interactions in crowded environments. This chapter illustrates how the design of interaction topologies and edge weights, informed by psychological studies, critically affects both model performance and interpretability, emphasizing the sensitivity of these methods to the structure and weighting of interaction graphs. **Chapter 5** generalizes our consensus model to capture directed, state-dependent, and potentially signed (both cooperative and repulsive) interactions. Using Lyapunov theory, it derives mathematical conditions that ensure stability and robustness in the face of nonlinear and dynamic inter-agent interactions. Building on these, **Chapter 6** develops a spatio-temporal optimal control framework for learning interactions from observed data. We present rigorous derivations of necessary and sufficient optimality conditions for learning robust, state-dependent interaction weights from observations. Additionally, we propose an algorithmic solution for recovering interaction behavior from data. This chapter focuses on uncovering interpretable interaction rules and analyzes the richness conditions required for robust model identification. A numerical example is provided to evaluate the effectiveness of the proposed method. **Chapter 7** summarizes the key contributions of this dissertation, and outline potential directions for future research. **Chapter 8** provides supplementary material, including detailed proofs of selected lemmas and theorems presented throughout the work.

Chapter 2: Literature Review

This chapter reviews relevant literature on multi-agent systems (MAS), focusing specifically on human-inspired interaction models, consensus algorithms, edge weight design, stability and robustness analysis. The chapter identifies key challenges and research gaps that motivate the contributions of this dissertation.

2.1 Human Coordination as Inspiration for Multi-Agent Systems

Human behaviors in collaborative and crowd scenarios serve as powerful inspiration for designing decentralized coordination strategies in MAS. Human motor coordination, shaped by motor synergies, offers rich examples of adaptability, error compensation, and distributed control. The concept of motor synergy describes the structured co-variation among multiple degrees of freedom to stabilize task performance [10, 14]. Analysis tools such as the Uncontrolled Manifold (UCM) method and Principal Component Analysis (PCA) have demonstrated the underlying dimensionality reduction and compensation mechanisms in joint motor actions [9, 15].

Interestingly, these behavioral dynamics align conceptually with consensus (agreement) algorithms [5] in MAS. In collaborative tasks, humans implicitly coordinate their effort to maintain balance or complete a shared objective—similar to agents minimizing disagreement in consensus frameworks [16]. Chapter 3 explores this connection by presenting a decentralized model that

captures the essence of interpersonal motor synergy, validated against experimental data.

In more complex environments, such as crowded scenes, human interactions are dynamic, context-dependent, and implicitly coordinated. Predicting such behavior is vital for autonomous systems operating in human environments. While traditional approaches like the Social Force Model [17] and velocity-based models [18] provide interpretable physical models, learning-based approaches—especially using Graph Neural Networks (GNNs)—have emerged as effective tools for modeling human trajectories in crowds [19, 20]. A key limitation of many deep learning methods is the lack of interpretability and sensitivity to interaction topology. Recent advances such as Social-STGCNN [19] use kernel-based edge weighting to address these issues, but often rely on handcrafted proximity functions. Chapter 4 builds on this line of work, incorporating geometric kernel functions derived from psychological insights into pedestrian behavior [21], showing how principled edge design can significantly improve prediction accuracy and interpretability.

Psychological studies have provided empirical support for consensus-based modeling of crowds through an experiment-driven approach [21]. These studies demonstrate that collective behavior in human crowds emerges from local interactions, with individuals linearly combining the influence of multiple neighbors within their vicinity. Their model incorporates a weighted agreement protocol, where the influence of neighbors decays with distance. This approach aligns with the concept of weighted consensus in MAS, indicating that individual motion is influenced by the aggregate behavior of neighbors, adjusted by the distance between them, which collectively results in coherent motion patterns. They validated their model with experimental human crowd data and emphasized the importance of understanding local interactions when modeling collective crowd dynamics. However, the design of such interaction rules—particularly for moving crowds in dynamic environments—remains an open challenge.

2.2 Consensus Algorithms in Multi-Agent Systems

Consensus algorithms form the backbone of many distributed systems, enabling agents to converge to a common state through local interactions [6]. In such systems, the communication topology is typically modeled as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node represents an agent and each edge $(j, i) \in \mathcal{E}$ denotes that agent i receives information from agent j. An edge weight w_{ij} quantifies the strength or influence of that interaction. The properties of the graph—such as whether it is directed or undirected, weighted or unweighted, signed or unsigned—have a significant impact on the convergence behavior of consensus algorithms. Classical consensus theory often assumes that the graph is undirected (i.e., symmetric weights) and positively weighted, which simplifies the analysis considerably. In an undirected graph with strictly positive weights, the Laplacian matrix L is positive semidefinite and has a single zero eigenvalue (corresponding to the all-ones vector) if and only if the graph is connected. This spectral property ensures convergence to a common value, as all nonzero Laplacian eigenvalues are positive, resulting in a stable consensus manifold [5].

Directed graphs, however, introduce complexity due to asymmetric interactions. Conditions such as strong connectivity, or the existence of a rooted out-branching (a directed spanning tree) are essential to ensure consensus [22]. In general, ensuring consensus on directed graphs requires careful graph conditions and weight design compared to the undirected case.

Consensus algorithms have evolved significantly, expanding from simple single-integrator rendezvous algorithms to more complex higher-order integrator dynamics [23], dynamic communication topologies [24], robustness to measurement noise [25], and tolerance to communication delays [26]. Variants such as flocking behavior [27], formation control [28], and leader-follower

configurations [29] illustrate the adaptability of consensus methods to diverse real-world scenarios. Real-world interactions often involve not only cooperative behaviors but also antagonistic or competitive dynamics. Signed graphs extend consensus algorithms to networks with both positive (attractive or cooperative) and negative (repulsive or antagonistic) interactions by allowing edge weights of either sign [30, 31]. This extension enables the modeling of phenomena such as opinion dynamics with disagreement or teams with adversarial relationships. For more detailed discussions of consensus variants, see [32].

In summary, classical consensus theory provides a foundational framework for decentralized agreement under fixed, homogeneous interaction rules. It guarantees stability and convergence in static, positively weighted graphs under mild connectivity conditions. However, the assumptions of fixed topology and uniformly cooperative interactions often do not hold in complex environments—for example, when modeling pedestrian interactions in crowded settings. This motivates a closer examination of how edge weights can be designed or adapted to improve performance and robustness, as well as how to handle more general weight patterns, including asymmetry, negativity, and state dependence. The remainder of this chapter reviews key developments in edge weight design, stability and robustness analysis, and emerging approaches for learning interaction weights from data, highlighting open challenges that set the stage for our contributions in Chapters 3–6.

2.3 Edge Weight Design in Weighted Consensus

Edge weights are critical parameters in multi-agent networks, as they encode how strongly agents influence each other. The choice of weights impacts convergence speed, steady-state bias,

and overall robustness of the group behavior. Traditional consensus models typically assume nonnegative weights to enforce cooperation. However, many real-world systems—including social and biological networks—exhibit both positive and negative interactions, which motivates the use of signed graphs to represent attraction vs. repulsion [31].

Historically, the design or tuning of edge weights has often been treated as an optimization problem, typically for undirected networks to improve network performance. Many approaches formulate weight selection as a convex optimization problem incorporating constraints that reflect the physical characteristics of the network or desired performance outcomes [33]. One common objective is to shape the spectrum of the Laplacian matrix, due to its strong influence on convergence and stability of consensus dynamics [5]. For example, in undirected graphs a typical goal is to maximize the algebraic connectivity (the second-smallest Laplacian eigenvalue) to speed up consensus, while keeping weights nonnegative and bounded. A larger algebraic connectivity implies the network can reach agreement more rapidly and is more robust to perturbations. Other spectral approaches include identifying node and edge weights by minimizing the gap between the largest and smallest eigenvalues of the Laplacian to tighten the convergence bounds [34]. Note that the largest eigenvalue of the graph Laplacian is connected to the stability analysis of discrete-time consensus and several continuous-time formation control. Beyond spectral metrics, researchers have explored alternative design criteria. Effective resistance is a graph metric that correlates with overall network performance (e.g. robustness to node or link failures). Efforts such as [35] focus on selecting edge weights that minimize the total effective resistance of a graph, thereby improving connectivity and fault tolerance. For example, Sun *et al.* [36], jointly optimize the design of the network topology and edge weights under cardinality constraints on number of edges. This is relevant in scenarios like ad-hoc wireless networks, where adding communication

links incurs cost and one seeks an optimal sparse graph with weighted edges. More recent work has extended these concepts to directed graphs [37]. In [37], the authors design edge weights so that the Laplacian matrix exhibits a prescribed spectrum while also meeting secondary objectives, like excluding certain undesirable edges and promoting sparsity in the network. Such approaches acknowledge the added complexity of directed interactions, though they still typically assume time-invariant weights determined a priori.

In parallel, some model-based control strategies implicitly define edge weights through physical rules. Distance-based artificial potential functions (APFs) are commonly used in formation control and flocking: the edge weight between two agents may be derived from a potential function of their relative distance [5]. The minimum of this potential corresponds to a desired formation spacing, and the interaction force (gradient of the potential) acts like a state-dependent edge weight that pushes agents into formation while repelling them if they get too close. Reynolds' seminal flocking rules [7] applied such ideas to simulate swarms of birds, and later work in multi-robot systems incorporated similar terms to maintain formations and avoid collisions [38]. In these approaches, the edge weighting scheme is often handcrafted (e.g., a function of distance or relative velocity) rather than optimized, yet they provide intuition for designing adaptive interactions.

Despite the variety of methods for edge weight design, most studies have imposed restrictive assumptions. In particular, the majority of prior works consider undirected graphs with fixed, nonnegative weights. Recent work has addressed potential-based control in directed networks by employing the *edge Laplacian*—a matrix that encodes flows of information among agents and elucidates how spanning trees and cycles influence convergence [39]. In [22, 40], the authors extend potential-based methods to directed networks via a structural Lyapunov matrix, whose positive semidefiniteness characterizes stable directed topologies. While the theory of signed

graphs allows for both positive and negative weights, it generally deals with static signed networks (fixed positive or negative ties) and rarely with weights that might change over time or depend on the state of the agents [31]. Early consensus and formation control research also assumed static or predetermined interaction topologies. Real-world MAS, by contrast, often exhibit time-varying or state-dependent interactions – for example, communication links appearing or disappearing or interaction strength changing with distance or other context. Recent advances such as self-attention encoders [41], aim to recover edge weights from data, yet typically assume time-invariant structures. While graph neural networks (GNNs) [19,20,42] offer increased flexibility in learning interaction patterns, they can be difficult to interpret for performance analysis. There is a clear research gap in developing systematic methods to design or learn edge weights for directed networks that may include both cooperative and antagonistic interactions, without assuming the weights remain constant or exclusively positive. This gap limits our ability to deploy consensus-type algorithms in complex scenarios like competitive teams, social networks with trust or distrust relations, or adaptive multi-robot teams.

Addressing this gap is a primary focus of our work. In Chapter 5, we will introduce a generalized consensus model with state-dependent edge weights that can take on both positive and negative values. We derive conditions to ensure such networks remain stable and convergent, thus moving beyond the limitations of prior weight design methods. Furthermore, instead of manually tuning these weights, in Chapter 6 we develop an approach to learn optimal interaction weights from data, enabling directed MAS networks to automatically discover appropriate positive and negative coupling strengths. By lifting the restrictive assumptions of static and sign-fixed weights, we aim to significantly broaden the applicability of consensus strategies to dynamic, complex interaction behaviors.

2.4 Stability and Robustness Analysis of Weighted Consensus

Ensuring the stability and robustness of MAS is essential for any control or learning application. Classical approaches to guarantee stable consensus (agreement) rely on Lyapunov stability theory, wherein one constructs an energy-like Lyapunov function that decreases along system trajectories, thereby ensuring convergence to the consensus state. For example, in a consensus algorithm one can often use a quadratic Lyapunov function based on disagreement among agents and show it decreases under the update rules, proving that the agents' states converge to an equilibrium. Lyapunov-based analyses using Lyapunov functions and invariance principle [43] have been the cornerstone for stability proofs in countless MAS studies due to their generality and the intuition they provide (i.e., defining a potential function that the system is minimizing) [5]. An alternative and complementary perspective is offered by contraction theory, which focuses on the convergence of trajectories with respect to each other. If a dynamical system is contractive, any two trajectories (starting from different initial conditions) will exponentially converge toward each other over time [44]. However, in these methods only convergence to the consensus manifold can be established. In scenarios with non-linear interconnections when analyzing the system dynamics from the nodes' perspective through the graph Laplacian, common tools fail at ensuring strong stability and robustness properties such as uniform asymptotic stability or input-to-state stability, which often rely on the design of strict Lyapunov functions [43]. In such a scenario, analysis through edge-based perspective introduced in [45] and further developed in [22,46] provides an alternative advantageous solution, where the consensus problem is analyzed using the relative state between neighboring agents, i.e., the edges of the graph as opposed to analyzing the evolution of the nodes. This framework allows for more intuitive analysis of

nonlinear interconnections (edge weights) and the consensus problem within this framework would be recast as stability of the origin which allows for asymptotic stability of the consensus manifold via strict Lyapunov functions [40]. Through partitioning of the graph into spanning and co-spanning trees, the edge agreement would be reduced to analysis of spanning tree component. In Chapter 5, we adopt similar framework to address the stability and robustness. Stability and robustness for consensus [46, 47] collision avoidance [48] and connectivity maintenance [49] have been explored using edge agreement in the sense of input-to-state stability but only for directed cycles and directed spanning tree [48, 49] using barrier functions with only positive edge weights. In this work, we extend it to consider more general directed graphs. Our framework allows for generalization to more complex desired configurations beyond collision avoidance and connectivity maintenance and we relax the assumption that all edge weights must be positive.

2.5 Learning Interaction Rules from Observation

Inverse Optimal Control (IOC) and Inverse Reinforcement Learning (IRL) in the context of learning from demonstration provides a framework for inferring the underlying objective or cost that makes an observed behavior optimal. In essence, IOC and IRL ask why agents behave as they do by recovering a cost function or parameters that could have led to the demonstrated MAS trajectories- assuming that the expert (e.g., human) employed an optimal strategy. This assumption aligns with motor control theory, which frames CNS as an adaptive optimizer [50].

Unlike behavioral cloning or imitation learning—which replicate behavior without understanding its underlying rationale—IOC infers the cost function that the expert is implicitly minimizing. This typically involves parameterizing the cost as a weighted sum of features, with

the goal of learning the weight vector that explains the demonstrated trajectories [51, 52].

While extensively explored in single-agent settings, IOC remains underdeveloped for MAS. Most existing work focuses on centralized settings that require full access to system dynamics. Recently, decentralized extensions have emerged. For example, [53] proposed a distributed IOC framework in which agents learn their local objectives using only partial trajectory data. Similarly, [54] framed multi-agent coordination as a differentiable game to enable joint policy learning from demonstrations.

In this work, we formulate our problem as a spatio-temporal optimal control problem and learn the interaction rules from observation by solving for the optimality conditions. In contrast to the common IOC or IRL framework of learning a global policy or value function that assumes the cost as weighted sum of features and learning those weights, our approach focuses on learning the local interaction rules. This aligns well with decentralized control objectives and opens the door to interpretable models of multi-agent coordination. In other words, we want to learn what to communicate given observations of MAS executing a task. Chapter 6 formalizes this framework, derives necessary and sufficient conditions for optimality, and demonstrates how to recover interaction rules from trajectory data.

2.6 Summary

In summary, although substantial progress has been made in developing consensus algorithms, edge weight design, and stability analysis for MAS, critical research gaps remain. Many existing approaches rely on static, nonnegative, and symmetric interaction assumptions that fail to capture the dynamic, state-dependent, and even antagonistic relationships observed in real-world

scenarios. Moreover, while human-inspired models and IRL offer promising frameworks for learning adaptive interaction rules, their integration with robust, decentralized stability guarantees is still underexplored. Addressing these challenges is essential for advancing more versatile and interpretable coordination strategies in complex, dynamic environments.

Chapter 3: Modeling Human-Human Collaboration: A Connection Between Inter-Personal Motor Synergy and Consensus Algorithms

This chapter is based on the paper published in the 2022 ACC [9].

3.1 Overview

In this chapter, we establish a theoretical connection between the uncontrolled manifold (UCM) characterization of inter-personal motor synergy (IPMS) and consensus algorithms used in decentralized control. We propose a control-theoretic model that captures the dynamics of human-human collaboration, demonstrating how coordinated behavior can emerge from consensus-type interactions. The model is validated using experimental data from a finger force matching task. This work bridges insights from motor neuroscience and control theory, offering a principled framework for modeling human coordination that can inform both human collaboration studies and the design of local interaction rules for multi-agent systems.

3.2 Introduction

A common challenge in engineering, mathematics, and physical and biological sciences is how to address indeterminacy or redundancy encountered in many systems [55], oftentimes

observed in overactuated systems. Humans have a complex body structure typically with more degrees-of-freedom (DOF) than those required for a particular motor task, and it is not clear how these mechanically or mathematically independent DOFs are controlled by the central nervous system (CNS) [56]. Even the seemingly simple task of grasping a glass of water requires the CNS to determine and synergistically control the forces and torques applied to the object by the thumb and fingers from infinite possibilities [57, 58]. In human motor control research, this type of control is known as motor synergy, with two key features of error compensation and dimension compression [14]. Motor synergies are defined as "neural organization of a multi-element system that (1) organizes sharing of a task among a set of elemental variables; and (2) ensures task-specific co-variation among elemental variables with the purpose to stabilize performance variables" [10]. Consequently, if one of the elements deviates from its share of task such that it adversely affects the task goal, the other elements act in a manner to compensate for the introduced error in task performance.

The UCM analysis [14] provides a way to quantify synergies. More specifically, UCM analysis indicates which DOFs are potentially controlled by the CNS. Within this approach, for a given performance variable (e.g., total force or moment required for grasping a glass of water), the space of elemental variables (e.g., finger forces and moments produced in the grasping task) is decomposed into two orthogonal subspaces: i) the UCM subspace, in which variations of the elemental variables do not affect the performance variable; thus the CNS does not explicitly control variations within this space, and ii) the orthogonal complement to the UCM (ORT), where the CNS explicitly controls variations of elemental variables, as these variations change the performance variable [14]. Motor synergy is typically quantified as the normalized difference between variability along these two subspaces [10]. Similarly, if there is not a clear

performance variable, principal component analysis (PCA) is used for examining the variability in performance [59]. PCA finds relationships in high-dimensional datasets and maps them into a space with principal components (PC) that reflect the dataset's major dimensions of variation [60].

When a person works with others to lift furniture, for example, individual CNSs are required to work together and agree on a share of force that is enough to lift the furniture; thus, the two motor systems behaves cooperatively to achieve a common motor task. Previous studies showed inter-personal motor synergy (IPMS), with key properties of error compensation and dimension reduction, as the potential mechanism responsible for this cooperation among independent CNSs in performing shared motor tasks [9, 15]. The UCM and PCA approaches have been used for characterizing IPMS [15, 61].

The past decade has seen remarkable advances made in our understanding of motor synergies within a person [58, 62]. In contrast, our understanding of IPMS is extremely limited [15] though many of our daily tasks involve multiple people working together, such as lifting and carrying a piece of furniture together. Co-working is not just common among people but in many other settings, such as a team of robots that coordinate to achieve a desired task [63]. In the field of control theory and robotics, decentralized cooperation in multi-agent networked systems has been studied through the framework of consensus algorithms [6]. Consensus algorithms allow a team of agents to agree on a common objective in a decentralized fashion by only interacting with neighboring agents. Here, we aim to establish a connection between consensus algorithms and IPMS and use this relationship to model HHC. The result is an intuitive control-theoretic model that captures how two or multiple people can achieve a common objective in a decentralized fashion.

Control-theoretic neural mechanism models that can reproduce sensorimotor interactions

and motor synergies in intra-personal tasks exist [62]. In contrast, mathematical models that explain the neural control mechanisms underpinning IPMS are not yet available. Models of this kind not only enhance understanding of human-human collaboration (HHC), but they can also provide good foundations for MAS and facilitate intuitive human-robot collaboration. This paper draws inspiration from prior behavioral performance observed in an HHC experiment [9]. To elucidate the mechanisms responsible for IPMS, we provide a consensus based model that can reproduce ensemble-level behaviors of multiple CNSs working together.

3.3 Preliminaries

3.3.1 Inter-personal Interaction Experiment

The model provided in this paper is inspired by a previously conducted human interaction experiment [9]. In this experiment, 21 pairs of people (i.e., 42 participants) were asked to work together and perform a finger force production task by pressing on force sensors placed at two ends of a seesaw-like apparatus as shown in Figure 3.1. A monitor was placed in front of each co-worker to provide visual feedback on the task. Participants were required to maintain a target force for 23 seconds over 10 trials. They received visual feedback of the target force (i.e., $5N$) and their averaged combined force. They were asked to remain quiet during the experiment and not communicate verbally. The force profile of index and middle fingers of each participant were recorded during the experiment. Full details of the experiment can be found in [9].

A second-order Butterworth low-pass filter with 15 Hz cut-off frequency was used to eliminate the force sensor measurement noise [64]. From experiment, 210 combinations of data for all subjects and all trials (i.e., 21 pairs by 10 trials of finger force trajectories) were recorded.

Removing outliers (i.e., trials with finger forces less than $1N$, or 20% of the average), provided us with 182 trials of force data (Figure 3.2).

The experimental data indicated that most pairs could manage to produce the target force and maintain this objective with little deviation. Without verbal communication and only relying on visual feedback, co-workers formed IPMS; they minimized the error between the target force and the combined output by compensating for each other's error. In other words, they collaborated synergistically at inter-personal level to satisfy the goal at the task-level [9].

3.3.2 Notation

Denote by I the identity matrix, and by $\mathbf{1}_N$ the column vector with N entries equal to 1 (i.e., $\mathbf{1}_N = [1 \dots 1]^T$). Let $\hat{\square}$, $\bar{\square}$, $\dot{\square}$, $\square_d$ denote estimation, average, derivative, and desired value of $\square$, respectively. The operator $E[\cdot]$ denote the expected value (mean) of its argument.

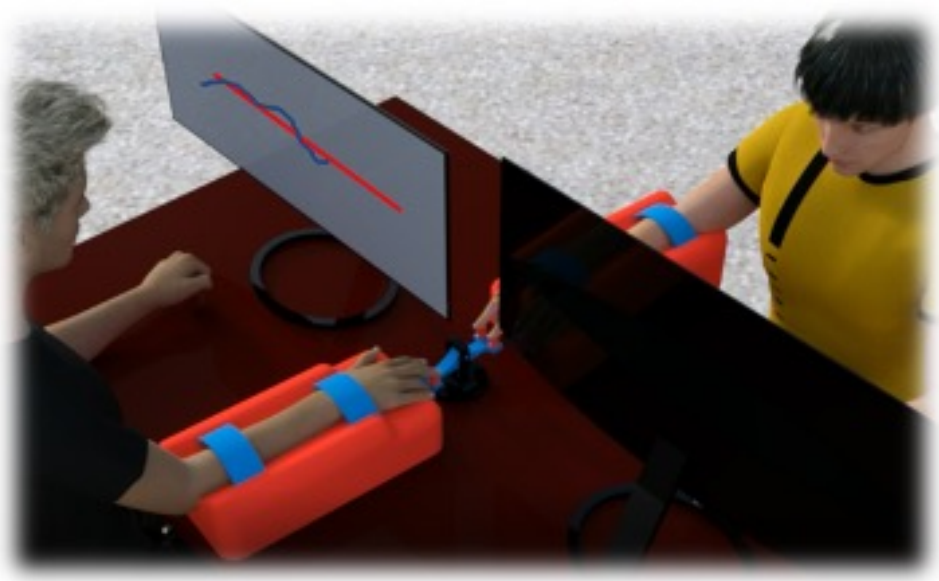


Figure 3.1: Human Interaction Experiment (taken from [9]).

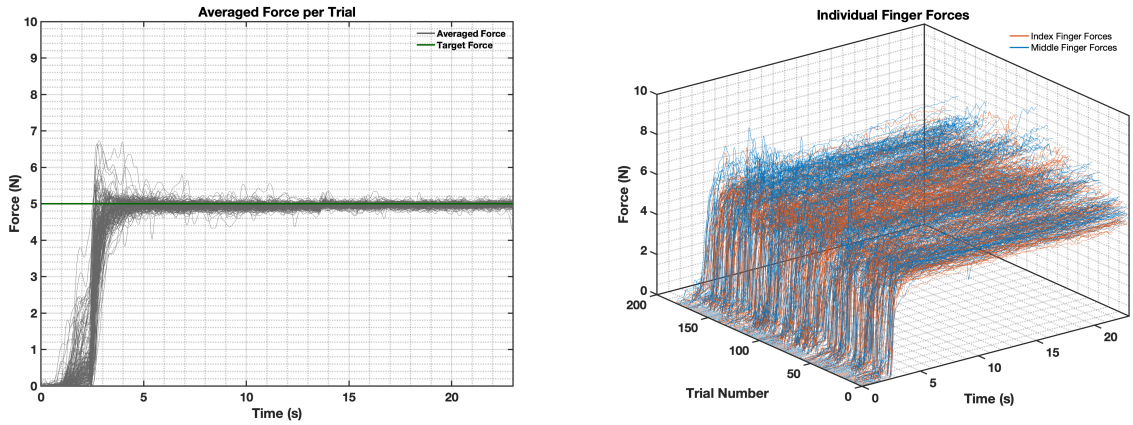


Figure 3.2: Demonstration of experimental co-working data. The left plot presents the average force generated per trial (182) by the four individual fingers. The right plot shows the individual finger data per trial. The orange corresponds to forces generated by index fingers, and the blue to middle fingers.

3.3.3 Topology of Interaction

We use notations from algebraic graph theory [5] to represent the interaction between co-working agents. A graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consists of a set of N nodes $\mathcal{V} = \{1, \ldots, N\}$, and a set of edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$, representing the interaction between agents. The adjacency matrix is defined such that its ij-th entry is given by $[A]_{ij} = a_{ij}$, where $a_{ij} = 1$ if $(i, j) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise. We interpret $(i, j) \in \mathcal{E}$ to mean certain information about the ith agent is available to the jth agent. Let $\mathcal{N}_i = \{j \in \mathcal{V} | (j, i) \in \mathcal{E}\}$ denote the set of neighbors for the ith agent. In this chapter, we assume edges between agents to be bidirectional and focus our analysis on undirected and unweighted graphs. An undirected graph is said to be connected if there is a path between any two agents. The entries of the graph Laplacian are defined as $[L]_{ij} = l_{ij}$, where

$$l_{ij} = \begin{cases} |\mathcal{N}_i|, & j = i \\ -a_{ij}, & j \neq i \end{cases} \tag{3.1}$$

and $|\mathcal{N}_i|$ denotes cardinality of agent i's neighbor set. The graph Laplacian has several useful spectral properties [5]. For an undirected graph, L is a symmetric positive semi-definite matrix with at least one zero eigenvalue ($\lambda_1 = 0$) with corresponding eigenvector $\mathbf{1}_N$, satisfying $L\mathbf{1}_N = 0$. Further, when $\mathcal{G}$ is connected, L has exactly one zero eigenvalue (with eigenvector $\mathbf{1}_N$).

3.3.4 Level of Error Compensation

The level of error compensation between agents, one of the features of motor synergies, can be captured by the sharing ratio, i.e., how the CNS shares the task among agents [62]. This property explains that if one agent produces less force than its expected share, the other agents would compensate for that so their average force matches the desired target force. To capture the level of error compensation, we sampled the share of the forces generated by each of the agents, s_i, for $i = 1, \ldots, 4$, from a normalized, truncated multivariate normal distribution with mean of $1/N$, and a $N \times N$ covariance, C_m, with $N = 4$, such that $s_i \in [0, 1]$ and $\sum_i s_i = 1$. To estimate C_m, we calculated the covariance matrix, C_j, of steady-state (i.e., 16-23s) finger forces, $Y = [Y_1, Y_2, \ldots, Y_N]^T$, for each trial $j = 1, 2, \ldots, n_{Trials}$ and then get the average of covariance across all trials (i.e., $n_{Trials} = 182$), where

$$
\begin{aligned}
C_j &= E[(Y - E[Y])(Y - E[Y])^T], \\
C_m &= \frac{1}{n_{Trials}} \sum_{j}^{n_{Trials}} C_j.
\end{aligned}
\tag{3.2}
$$

3.3.5 Agent Dynamics

We consider a network of double integrators to represent finger models contaminated with sensory and motor noises [62] for a collection of N agents. The dynamics of the ith agent in normal form is as follows:

$$\begin{aligned} \dot{z}_i &= A_c z_i + B_c u_i + w_i \\ y_i &= C_c z_i + v_i \end{aligned} \tag{3.3}$$

where $z_i = [y_i, \dot{y}_i]^T \in \mathbb{R}^2$ is the agent i's state (including the force, y_i, and rate of force, $\dot{y}_i$), and $u_i \in \mathbb{R}$, is the control input of agent i, for $i = 1, 2, \ldots, N$. The process noise and measurement noise are represented by Gaussian random vectors $w_i \sim \mathcal{N}(0, Q)$ and $v_i \sim \mathcal{N}(0, R)$ with zero mean and covariances Q and R, respectively. The noise vectors are all assumed to be mutually independent. Finally, (A_c, B_c, C_c) represent a chain of integrators system, i.e.,

$$A_c = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B_c = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C_c = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

Each agent is assumed to be aware of its states, but not necessarily others. Due to access to their state, agents could estimate the average share of their neighbors' contributions based on the visual feedback they received. We can see this intuitively in the task of lifting a heavy object together, as people tend to focus on their own state and on the object's state (i.e., height) which combines the states of all agents.

3.3.6 State Estimation Using Kalman Filter

The standard continuous-time Kalman filter [65] is employed to reduce the uncertainty in the state estimation of each agent, perturbed by sensory and motor noises [66]. Assuming $P(0) = P_0$, $\hat{z}(0) = \bar{z}_0$, the error covariance update is given by

$$\dot{P} = A_c P + P A_c^T + Q - P C_c^T R^{-1} C_c P. \tag{3.4}$$

So, the state estimate update is given by

$$\dot{\hat{z}} = A_c \hat{z} + B_c u + K(y - C_c \hat{z}). \tag{3.5}$$

where $K = P C_c^T R^{-1}$ is the Kalman gain.

3.3.7 Desired Averaged Dynamics

In the HHC experiments, agents were tasked with performing collaborative output-tracking of the target output y_t. Agents were only provided with the averaged estimated output contributed by the team, i.e.,

$$\hat{\bar{y}} = \frac{\hat{y}_o}{N} = \frac{\mathbf{1}_N^T}{N} \hat{y}, \tag{3.6}$$

and (presumably) a notion of their individual output contributions, where $\hat{y}_o = \sum_{i=1}^{N} \hat{y}_i$ is the overall estimated output contributed by the team and $\hat{y} = [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_N]^T$ is the individual agents'

estimated output. So, the kth derivative of the average output is

$$\hat{\bar{y}}^{(k)} = \frac{\hat{y}_o^{(k)}}{N} = \frac{\mathbf{1}_N^T}{N}\hat{y}^{(k)}. \tag{3.7}$$

The dynamics of $\hat{\bar{y}}$ can be represented as a chain of integrators with $\bar{z} = [\hat{\bar{y}}, \dot{\hat{\bar{y}}}]^T$ via

$$\dot{\bar{z}} = A_c\bar{z} + B_c\ddot{\hat{\bar{y}}}. \tag{3.8}$$

Typical performances of the overall output force contributions from the HHC experiment (Figure. 3.2) demonstrate a dominant second order behavior for the overall output. Thus, to model the experimental data, we wish to obtain a desired output response ($\hat{\bar{y}}_d$) for the average output as dictated by the second order desired system dynamics. The second order behavior of finger force production task has also been confirmed in our previous study [62].

$$\begin{aligned} \dot{\bar{z}}_d &= A_d\bar{z}_d + B_d y_t \\ \hat{\bar{y}}_d &= C_d\bar{z}_d. \end{aligned} \tag{3.9}$$

with state $\bar{z}_d = [\hat{\bar{y}}_d, \dot{\hat{\bar{y}}}_d]^T$ and the individual finger's target force $y_t = 5N$. Besides, the system (A_d, B_d, C_d) is assumed to be linear time-invariant and stable in accordance with experimental observations in [62]. The desired system dynamics can also be written in normal form [43], namely

$$\dot{\bar{z}}_d = A_c\bar{z}_d + B_c\ddot{\hat{\bar{y}}}_d$$

where $\ddot{\hat{\bar{y}}}_d = \alpha_d\bar{z}_d + \gamma_d y_t$, $\alpha_d = C_d A_d^2$ and $\gamma_d = C_d A_d B_d$.

3.4 Problem Formulation

The objective of this chapter is to address the challenge of modeling human-human collaborative motor behavior. Specifically, we seek to understand how multiple individuals coordinate their redundant DOFs in achieving a shared motor task without explicit verbal communication, relying primarily on indirect feedback mechanisms. To accomplish this, we focus on the following objectives: (1) Establishing a theoretical link between consensus algorithm and UCM theory. (2) Designing a control-theoretic model that captures essential features of human neuromotor control, including sensory and motor uncertainties. (3) Employing Lyapunov-based analysis methods to examine the stability and convergence properties of the proposed model, ensuring both effective task-level performance and robust inter-agent coordination.

3.5 Control Law

The objective for the HHC experiment was to collaboratively achieve and maintain a target output. We propose that the following node-level control law which solely relies on the information available to the agent, namely, the agent's own states and the average output states, can achieve the objective of HHC experiment.

$$u_i = \hat{\bar{\hat{y}}} + (\hat{\bar{y}} - \hat{y}_i) + \eta(\hat{\bar{\dot{y}}} - \hat{\dot{y}}_i) - y_t(\tfrac{1}{N} - s_i) \tag{3.10}$$

where $\eta > 0$ is a tuning parameter, and $\hat{\bar{\hat{y}}}$ will be discussed shortly. We believe that the proposed control law can explain how humans work together as described in the sequel.

Inspired by the UCM hypothesis [14], the system dynamics in equation (3.10) can be

decomposed into two orthogonal components:

I) The span$\{\mathbf{1}_N\}$ components, where changes lead to deviation from task goal. This component is equivalent to ORT subspace in UCM approach.

II) The components orthogonal to span$\{\mathbf{1}_N\}$, where changes do not lead to deviations from task goal and do not need to be explicitly controlled to achieve the task. This component is equivalent to UCM subspace in UCM approach.

Accordingly, we consider two components for the proposed ensemble-level control law $u = [u_1, \ldots, u_N]^T = u^{\parallel} + u^{\perp}$: (i) the task-level component $u^{\parallel}$ that ensures asymptotic convergence of the average output to the desired dynamics, and (ii) the inter-agent component $u^{\perp}$ represented by consensus algorithms to model collaboration between agents to agree on a common objective.

3.5.1 Ensemble-Level Dynamics

The control law provided in equation (3.10) can be manipulated to more clearly show how an agent behavior is affected by their neighbors through the average output. In particular, note that we may rewrite the equation as

$$u_i = \hat{\bar{y}} + \frac{1}{N} \sum_{j=1}^{N} \left[(\hat{y}_j - \hat{y}_i) + \eta(\hat{\dot{y}}_j - \hat{\dot{y}}_i) - y_t(s_j - s_i) \right]. \tag{3.11}$$

Stacking these into a single vector, the ensemble-level control law can be expressed as

$$u = \hat{\bar{y}}\mathbf{1}_N + \frac{1}{N}\left(y_t L s - L\hat{y} - \eta L\hat{\dot{y}} \right) \tag{3.12}$$

where $s = [s_1, \ldots, s_N]^T$, and L is the graph Laplacian associated with the complete graph, i.e., where for every agent $\mathcal{N}_i = \{j \in \mathcal{V} \mid j \neq i\}$. From this, and due to the fact that for an undirected graph $\mathbf{1}_N^T L = 0$, we get that

$$\begin{aligned} u^{\parallel} &= \tfrac{1}{N}\mathbf{1}_N\mathbf{1}_N^T u = \hat{\bar{y}}\mathbf{1}_N, \\ u^{\perp} &= (I - \tfrac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)u = \tfrac{1}{N}\left(y_t L s - L\hat{y} - \eta L\hat{\dot{y}}\right). \end{aligned} \tag{3.13}$$

where we use the property that $(I - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)L = L$ [5].

3.5.2 Task-Level Convergence Analysis

From equation (3.13) it follows that only the term $\hat{\bar{y}}$ affects the task-level component of the agents dynamics. Thus, we will design this term so that the averaged trajectories converge to that of the desired dynamics. This is captured by the following theorem.

Theorem 3.1. *For the agents with dynamics given as equation* (3.3)*, the averaged output of* (3.6) *asymptotically approaches the output of equation* (3.9) *if* $\hat{\bar{y}} = \alpha_d \bar{z} + \gamma_d y_t - \frac{1}{2}B_c^T P_c(\bar{z} - \bar{z}_d)$ *where* P_c *is the solution to*

$$P_c(A_c + B_c\alpha_d) + (A_c + B_c\alpha_d)^T P_c - P_c B_c B_c^T P_c + Q = 0$$

for $Q = Q^T \succ 0$*, with* $A_c + B_c\alpha_d$ *stable and* $(Q, A_c + B_c\alpha_d)$ *observable.*

Proof. Using the Lyapunov candidate

$$V = (\bar{z} - \bar{z}_d)^T P_c(\bar{z} - \bar{z}_d) \tag{3.14}$$

for some $P_c = P_c^T \succ 0$, it can be shown that the trajectories of $\bar{z}$ approach those of $\bar{z}_d$ asymptotically by the Lyapunov stability theorem [43]. □

As a result, $u^{\parallel}$ ensures the convergence of the span$\{\mathbf{1_N}\}$ components of the dynamics. Now that we know the averaged dynamics converge, we show the behavior of the UCM dynamics at inter-agent level.

3.5.3 UCM Convergence Analysis

In the previous section, we showed that the error between $\bar{z}$ and $\bar{z}_d$ vanishes after some time. Now, we demonstrate that the consensus algorithm can achieve the desired level of error compensation exhibited by the individuals in the HHC experiments as captured by the sharing of the task among agents (s_i). Let $\Psi^T = [\Psi_1^T, \Psi_2^T]$, where, $\Psi_1 = \frac{1}{N}[y_1 - s_1 y_t, \ldots, y_N - s_N y_t]^T$, $\Psi_2 = \dot{\Psi}_1 = \frac{1}{N}[\dot{y}_1, \ldots, \dot{y}_N]^T$. Let $\xi^T = [\xi_1^T, \xi_2^T]$. Denote by $\xi = \Pi\Psi = [\xi_1, \xi_2]^T$ the disagreement vector, and note that $\xi_2 = \dot{\xi}_1$, where $\Pi = (I - (1/N)\mathbf{1_N}\mathbf{1_N}^T)$ is the projection matrix onto the orthogonal complement to span$\{\mathbf{1_N}\}$. Using the facts that $\Pi\mathbf{1_N} = 0_N$ and $\Pi L = L\Pi$ we thus have

$$\dot{\xi} = \begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \end{bmatrix} = \begin{bmatrix} 0_N & I_N \\ -L & -\eta L \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}. \tag{3.15}$$

In other words, if consensus is achieved then $\xi \to 0$ as $t \to \infty$, which suggests that $\Psi(t) \to T$ as $t \to \infty$, where

$$T = \left[\Psi \in \mathbb{R}^{2N} | \Psi_1 \in \text{span}\{\mathbf{1_N}\}, \Psi_2 \in \text{span}\{\mathbf{1_N}\} \right].$$

The convergence analysis of equation (3.15) can be performed through Lyapunov analysis similar to Theorem 1 of [67]. This is captured by the following Theorem.

Theorem 3.2. *For any $\eta > 0$, the UCM component of the control law in equation (3.13), $u^{\perp}$, will lead to consensus on the deviation from the share of the average force, i.e., Ψ will asymptotically converge to the set T.*

Proof. Proof follows from the choice of the positive definite Lyapunov function candidate:

$$V = \frac{1}{2}\xi_1^T L \xi_1 + \frac{1}{2}\|\xi_2\|^2, \tag{3.16}$$

with $\dot{V} = -\eta {\xi_2}^T L \xi_2$. From LaSalle's invariance principle [43], the dynamics will converge to the largest invariant set where $\dot{V} = 0$, which occurs only when $\xi_2 = 0$. If $\xi_2 = 0$, it follows from (3.15) that $\Psi_2 =\in \text{span}\{\mathbf{1_N}\}$. Furthermore, $\xi_2 = 0$ is invariant if $\xi_1 = 0$, or equivalently, $\Psi_1 \in \text{span}\{\mathbf{1_N}\}$. Thus, T becomes the largest invariant set, $\Psi \to q$ for some $q \in T$. □

Due to the convergence of the average dynamics under Theorem 1, for static reference y_t, $\hat{\bar{y}}_i$ will in fact tend to 0. Using Theorem 3.1 and 3.2, we showed that $\Pi y \to \Pi y_t s$ and $\hat{\bar{y}} \to \hat{\bar{y}}_d$ under the choice of u_i in (3.13), as desired.

3.6 Simulation Results

3.6.1 Choice of Parameters

The desired averaged dynamics were designed as a second-order system in standard form (Figure. 3.2). To define the desired behavior, we estimated the desired damping ratio, ζ, and natural frequency, ω_n, empirically for each 182 trials (e.g., $\zeta = 0.72 \pm 0.1$, and $\omega_n = 9.32 \pm 2.01$). We used the mean values $\zeta = 0.72$ and $\omega_n = 9.32$ as nominal values for simulation, which resulted

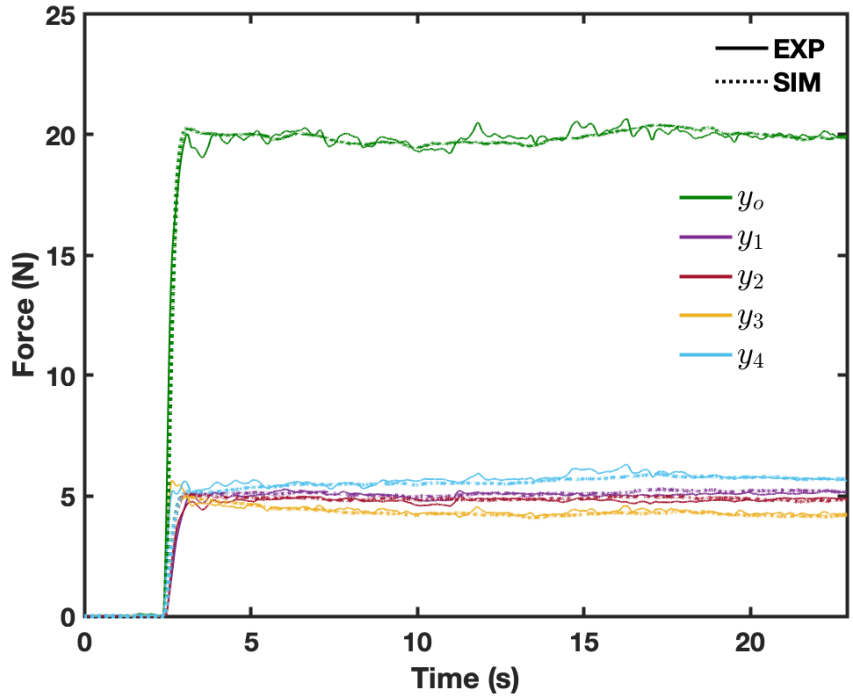


Figure 3.3: Comparison between simulation (dashed line) and experiment for a representative trial.

in

$$A_d = \begin{bmatrix} 0 & 1 \\ -82 & -13.04 \end{bmatrix}, \quad B_d = \begin{bmatrix} 0 \\ 82.004 \end{bmatrix}, \quad C_d = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

The $\eta = 1/\tau$ is a positive gain for consensus on the rate of force production, where τ is the reaction time. This rate has been shown to be the determining response parameter of the reaction time in humans when they receive visual feedback for performing finger force production tasks [68]. The reaction time between the presentation of visual information and the initiation of a movement correction for finger force production has been reported between 100-180 ms, with the average value of 135 ms [69]. For this reason, we used the nominal value of $\eta = 7.41$ for our simulation results.

The sharing ratios were sampled from a normalized, truncated multivariate normal distri-

bution as $s_i \sim \mathcal{N}(1/4, C_m)$, for $i = 1, \ldots 4$, and the covariance,

$$C_m = \begin{bmatrix} 0.0242 & 0.0038 & -0.0069 & -0.0106 \\ 0.0038 & 0.0217 & -0.0072 & -0.0087 \\ -0.0069 & -0.0072 & 0.0270 & -0.0015 \\ -0.0106 & -0.0087 & -0.0015 & 0.0335 \end{bmatrix}.$$

The measurement noise variance, $R = 2.24$, was estimated from the average across 182 trials of the experimental force variance in the transient section (i.e., 2-16s). The process noise covariance $Q = 9.5 \times 10^{-2} I_{2x2}$ was taken from [70].

3.6.2 Model Validation

We examined the validity of the model from different aspects: First, we show that simulation results accomplished the task goal of attaining and sustaining the target force similar to the HHC experiment. For that, we examined the root mean square error (RMSE) between the overall output and the target force for all trials in both simulation and experiment in the steady-state. We found that the mean RMSE ± standard error across trials for simulation (0.217 ± 0.088) closely matched that of the experiment (0.246 ± 0.094). Furthermore, the proposed model is capable of replicating individual trials as well as the force variability across trials.

3.6.2.1 Replicating a Trial

With the right choice of parameters for the desired behavior and the sharing ratio, the model can replicate individual trials. As an example, Figure. 3.3 shows that the simulation results for

a particular trial closely match the combined and individual agent's outputs of the experiment. The parameters for this sample trial were: $\zeta = 0.80$, $\omega_n = 7.86$, $\eta = 6$, and the sharing ratio $s^T = \begin{bmatrix} 0.2562, 0.2458, 0.2118, 0.2861 \end{bmatrix}$, all the other parameters including the measurement and process noises remained intact. Given the randomness caused by the measurement and process noise, we ran the simulation with these particular choice of parameters for 5 times and got the average of the 5 simulations. Table 3.1 summarizes RMSE (e.g., mean ± standard error) between the sample experiment and the averaged simulation results.

3.6.2.2 Examining Force Variability Using PCA and UCM

We calculated PCA for individual agents' output in steady-state. Table 3.2 indicates that for both experiment and simulation, about 90% of variance is along the UCM, defined by PC_{1-4} (see explained variance in Table 3.2). These PC's are nearly orthogonal to span$\{\mathbf{1_N}\}$ (ORT), as shown in Table 3.2 by evaluating their angles with span$\{\mathbf{1_N}\}$. This finding suggests that the task-level component of the control law ensures the convergence of span$\{\mathbf{1_N}\}$ components of the dynamics and is an indication of dimension compression of motor synergies which is replicated by our model.

To further validate the results of our simulation, we compared experiment and simulation in terms of inter-agent synergy using UCM [10]. For the steady-state data, we computed the UCM and ORT bases and projected the data onto those bases. Finally, we computed variance along UCM (V_{UCM}) and ORT (V_{ORT}) as well as synergy index (i.e., ΔV) for both experiment and simulation (Table 3.3). Synergy index [10] was defined as $\Delta V = \frac{V_{UCM}/3 - V_{ORT}}{V_{TOT}/4}$. As shown in Table 3.3, the proposed control law can reproduce the synergistic interaction between agents.

Table 3.1: The Level of RMSE (Mean ± Standard Error) Between Simulation and Experimental Forces For a Representative Trial.

Force	RMSE (N)
e_{y_o}	0.43 ± 0.02
e_{y_1}	0.22 ± 0.01
e_{y_2}	0.22 ± 0.04
e_{y_3}	0.32 ± 0.01
e_{y_4}	0.33 ± 0.03

Table 3.2: Summary of PCA results. The explained variance (i.e., percentage of the total variance explained by each PC), and the angle of each PC with span$\{\mathbf{1_N}\}$ are reported.

Variable		**PC**			
		PC_1	PC_2	PC_3	PC_4
Explained variance	Experiment	49.40%	45.09 %	4.69 %	0.81%
	Simulation	40.17%	33.23 %	26.51 %	0.09%
Angle with span$\{\mathbf{1_N}\}$	Experiment	87.68°	90.1°	87.4°	3.48°
	Simulation	90.74°	87.91°	91.19°	2.52°

Table 3.3: Summary of UCM results

	V_{UCM}	V_{ORT}	ΔV
Experiment	2.2075	0.0020	1.3285
Simulation	2.6945	0.0026	1.3282

3.7 Summary

In this chapter, we developed a control-theoretic framework to understand neuromotor coordination in cooperative tasks, inspired by our previous HHC experiment [9]. We simulated two people performing a collaborative finger force matching task. This fine motor task resembles the task of lifting a heavy object by a group of people, providing a baseline for studying collaborative behavior. By controlling for factors like strength and weight of individuals, we could isolate the underlying coordination mechanism. Using Lyapunov analysis and the consensus protocol, we provided a control strategy for collaborative output-tracking, which was related to the UCM analysis and proved to achieve the desired transient and steady-state behavior at both the task and individual levels. The simulated results closely matched the experiment, suggesting consensus dynamics can model IPMS. We showed that consensus dynamics do not affect task-level performance, analogous to how CNS handles redundant degrees of freedom (UCM). This intuitive control law, grounded in motor control research, lays an effective foundation for developing coordination mechanisms in multi-agent systems.

However, as the nature of the given human experiment dictated, the proposed model assumed a fixed complete unweighted graph topology, as the share of contribution for completing the task was explicitly provided. However, many real-world collaborative tasks involve more complex, implicit, dynamic interactions. There is a need to learn these evolving interaction behaviors by inferring edge weights. In the next Chapter, we utilize the GNN based approach to capture the spatial and temporal evolution of trajectories in complex network interactions such as crowd navigation. We examine how the topology of interaction, as well as enforcing the degree to which neighbors affect one another, would impact the performance of modeling human interactions.

Chapter 4: Complex Network Interactions: Geometric Graph Neural Network Modeling of Human Interactions in Crowded Environments

This chapter is based on the paper published in IFAC-PapersOnLine 2024 [4].

4.1 Overview

In Chapter 3, we introduced a consensus-based framework to model human motor coordination strategies during collaborative tasks. This provided valuable insights into how consensus algorithm can recover human behavior in handling redundant degrees of freedom to achieve coordination. However, the model relied on a fixed, unweighted interaction graph, as dictated by the structure of our human experiment. Real-world settings—such as pedestrian navigation in crowded environments—feature dynamic and complex interaction patterns driven by social and psychological factors. This chapter focuses on learning human interaction patterns in crowds, where motion is governed by social dynamics and psychological factors. Drawing upon connections between GNN and weighted consensus protocol, we extend the Social Spatio-Temporal GNN (STGCNN) framework [19] by integrating geometric kernel functions and refined interaction neighborhoods inspired by empirical studies of crowd navigation from psychological literature. This knowledge-based graph topology improves the interpretability of the weighted adjacency matrices and enhances pedestrian trajectory prediction.

4.2 Introduction

Over the last decade, predicting how humans move and interact has become an active field of research, driven by applications in autonomous driving, surveillance, and the creation of socially aware robots (see [71,72] and references therein). The challenge lies in anticipating human motion due to its inherently complex and stochastic nature, influenced by internal and external stimuli, social dynamics, and psychological decision-making factors as described in [73]. This challenge is magnified in crowded spaces, where the interplay of these factors becomes even more intricate (see [11] for a review).

Despite this complexity, humans demonstrate an impressive ability to navigate crowds. As we move, we instinctively stay close to our group while maintaining a social distance from others. Our behavior adapts based on the type of interaction at play. Understanding these implicit interactions is crucial for autonomous systems, enabling them to predict and adapt to the unpredictable behaviors of pedestrians in the absence of explicit communication.

Various modeling-based and learning-based algorithms have been proposed to understand human interactions when navigating through crowds, as reviewed in [11]. Model-based approaches, such as the social force model by [17] and velocity-based methods like the reciprocal n-body collision avoidance technique introduced by [18], aim to develop intuitive strategies for explaining human interactions based on first principles and social factors. In contrast, learning-based methods leverage deep learning algorithms, including long short-term memory (LSTM) networks [74], convolutional neural networks (CNNs), generative adversarial networks (GANs) [75], and graph neural networks (GNNs) [19], to predict human motions directly from data.

Both model-based and learning-based approaches have their advantages and disadvantages.

Model-based algorithms typically have fewer parameters, making them computationally efficient and enabling careful reasoning about human intentions. However, they may overlook complex aspects of human behavior. Our proposed model takes the form of a geometric graph neural network, a structure inspired by the Social Spatio-Temporal Graph Neural Network (STGCNN) architecture proposed by [19]. However, we go beyond conventional GNN formulations by integrating geometric kernel functions and a refined neighborhood of interaction definition, drawing insights from psychological studies on human crowd behaviors ([21, 73]).

Psychological studies have empirically shown that humans base their navigation decisions on local interactions, particularly with neighbors within a limited field of view and proximity. [21]. They demonstrate that collective behavior in human crowds emerges from local interactions, with individuals linearly combining the influence of multiple neighbors within their vicinity. Their model incorporates a weighted agreement protocol, where the influence of neighbors decays with distance. At its core, both the GNN framework and state-dependent weighted consensus rely on the principle of local information aggregation to drive the evolution of each agent or node's state. This connection motivates our design of interpretable, state-dependent adjacency matrices.

Our hybrid approach integrates this domain knowledge into the GNN structure, resulting in a more interpretable, context-aware weighted adjacency matrix. The outcome is a framework that better captures the complexity of human interactions in crowds and offers improved predictive performance for pedestrian trajectory forecasting.

4.3 Preliminaries

4.3.1 Notation

The hat notation ($\hat{\square}$) refers to the estimated values. The Euclidean distance (ℓ^2 norm) between two vectors a and b is $\|a - b\|$.

4.3.2 Agent Dynamics:

Following the work in [19], we assume pedestrian trajectories follow a bi-variate Gaussian distribution. The parameters $(\mu_t^i, \sigma_t^i, \rho_t^i)$ represent the mean, standard deviation, and correlation coefficient of the Gaussian distribution, respectively, where we parameterize mean $\mu_t^i = (\mu_{p_x}, \mu_{p_y})_t^i$, standard deviation $\sigma_t^i = (\sigma_{p_x}, \sigma_{p_y})_t^i$, and correlation coefficient $\rho_t^i = (\rho_{p_x}, \rho_{p_y})_t^i$, for $i = 1, \ldots, N$. The pedestrian trajectories are given by $x_t^n = (p_{x_t}^i, p_{y_t}^i) \sim \mathcal{N}(\mu_t^i, \sigma_t^i, \rho_t^i)$, where $(p_{x_t}^i, p_{y_t}^i)$ are random variables capturing the probability distribution of pedestrian i's position at time t in the 2D space. The model is trained by minimizing the negative log-likelihood loss function:

$$J^i(W) = -\sum_{t=1+T_{\text{obs}}}^{T_{\text{pred}}} \log(\mathbb{P}(x_t^n | \hat{\mu}_t^i, \hat{\sigma}_t^i, \hat{\rho}_t^i)), \tag{4.1}$$

Here, $\hat{\mu}_t^i, \hat{\sigma}_t^i, \hat{\rho}_t^i$ are the estimated parameters of the predicted trajectory distribution, $\hat{x}_t^i$.

4.3.3 Graph Representation of Crowd Interaction:

We employ a graph representation rooted in algebraic graph theory to model pedestrian interactions within crowd [5]. Consider $G_t = (\mathcal{V}_t, \mathcal{E}_t)$ as a spatial graph at time t, comprising N nodes representing the relative positions of pedestrians in the scene at time t. The node

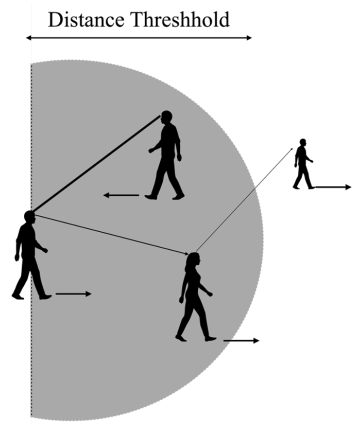


Figure 4.1: A visual representation of the neighborhood of interaction for a pedestrian (shown in black), incorporating a 180-degree field of view and a 5-meter distance threshold, selected based on empirical observations from psychological studies as described in [21]. Pedestrians within the shaded region, which extends up to 90 degrees on either side of the walking direction, are considered part of the neighborhood. The thickness of the connecting lines decreases with increasing distance, indicating that closer neighbors have a stronger influence on the pedestrian's motion (corresponding to Graph 2 neighborhood definition with distance-based weighting according to our kernel definitions)

set is $\mathcal{V}_t = \{v_t^1, \ldots, v_t^N\}$, and edge set $E = \{\mathcal{E}_t^{ij} = (v_t^i, v_t^j)\} \subseteq \mathcal{V}_t \times \mathcal{V}_t$ captures the interactions among agents. The adjacency matrix is defined as $A_t = [a_t^{ij}]$, determining the neighborhood of interaction, where $a_t^{ij} = w_t^{ij}$ if v_t^i and v_t^j are connected (agent i can receive information from agent j); otherwise, $a_{ij} = 0$. The interaction neighborhood will be defined in more details later. The weights w_t^{ij} quantify how strongly agents influence each other's movements and are determined through geometric kernel functions inspired by psychological studies on human crowd behaviors. The graph Laplacian is defined as $L_t = \Delta_t - A_t$, where Δ_t denotes the diagonal matrices of node degrees, taken as the sum of the weights for each neighbor: $\Delta_t = \sum_{j=1}^{N} a_t^{ij}$.

4.3.4 From State-Dependent Weighted Consensus To GNN Structure

In classical consensus algorithms, each agent updates its state as a weighted combination of the differences between its own state and those of its neighbors. For example, the discrete-time

consensus update rule is given by [5]

$$x(k+1) = Wx(k), \tag{4.2}$$

where $x(k)$ stacks the states of all agents at time step k, and W is a matrix that encodes the (possibly weighted) connections among agents—often subject to stochastic or row-stochastic constraints to ensure convergence. In weighted consensus settings, these weights can depend on the agents' current states; for example, the weight might decay with distance between the agents or be set to zero if they are outside the range. To link this to graph structures, we can see that the matrix W in (4.2) can be derived from the graph Laplacian L, specifically, suppose that $x^i(k)$ denotes the state of agent i. A single-agent update in discrete time can be written as

$$x^i(k+1) = x^i(k) + \sum_{(i,j)\in\mathcal{E}} a_{ij}(k)\left(x^j(k) - x^i(k)\right), \tag{4.3}$$

Collecting these into a vector form, one obtains

$$x(k+1) = (I - L(k))x(k). \tag{4.4}$$

where $L(k)$ is the (possibly time-varying or state-dependent) graph Laplacian constructed from the edge weights $a_{ij}(k)$. If we define $W = I - L(k)$, we get (4.2).

A GNN also aggregates information locally from neighbors. A generic (single-layer) GNN update can be written as

$$f(V^{(l)}, A) = \sigma\left(\mathcal{L}V^{(l)}W^{(l)}\right), \tag{4.5}$$

where $V^{(l)}$ is the stack of all node features at layer l, and $W^{(l)}$ is the matrix of trainable parameters at layer l. $\mathcal{L}$ is the normalized graph Laplacian matrix built from the adjacency matrix, as explained in [76], and σ is the activation function. Under a linear activation and a single-layer setup, (4.5) reduces to a discrete-time consensus-like operation in a multi-agent system. Mapping the graph convolutions to consensus update provides an intuitive connection between modern graph-based learning approaches and classical distributed control algorithms for multi-agent coordination. The Laplacian matrix as mentioned in Chapter 3, provides information on topology of interaction. However, without careful designing of this Laplacian matrix, specifically the topology of interaction and quality of interaction, we may not be able to capture the complex interactions among pedestrians in real-world cases. Using geometric kernels, we can specify these state-dependent interactions as will be explained later.

4.4 Problem Formulation

The main objective in this chapter is to predict the future trajectory positions of N pedestrians over a time horizon of T_{pred}, given their observed trajectories from $t = 1$ to $t = T_{\text{obs}}$. This task is particularly challenging in crowded environments, where pedestrian motion is governed by complex interactions and social dynamics.

4.4.1 Topology of Interaction

Diverging from the approach of social-STGCNN [19], which assumes a complete graph to establish the neighborhood set for each agent and weighs the importance of neighboring agents based solely on their distance, this work takes a more nuanced perspective. Our goal is

to experiment with different neighborhood sets inspired by psychological literature on human interactions in crowds, as described in [21], aiming to enhance model performance in predicting future pedestrian motion. This exploration seeks to identify the crucial factors in accurately modeling human interactions across various datasets.

Neighborhood of Interaction:

According to [77], people have a 180° horizontal field of view and face toward their walking direction ($\Delta x_t^i = x_{t+1}^i - x_t^i$). Therefore, interactions are expected to be influenced by neighbors within their field of view (±90°) who are approaching them. Empirical experiments on human crowds by [21] reveal that the impact of neighbors diminishes exponentially to zero at a distance of 4–5 m. The following sections explain how we incorporate these key points in constructing our weighted adjacency matrix.

1. ***Graph 1 (View):***

   To account for the 180° field of view, we introduce the following kernel-based weight. It stipulates that if the angle between walking direction of pedestrian i and j is smaller than 90°, they mutually influence each other; otherwise, the weight is set to zero.

$$w_{view,t}^{ij} = \begin{cases} K_{ij} & \Delta x_t^i \cdot \Delta x_t^j > 0 \\ 0 & \text{o.w.} \end{cases} . \tag{4.6}$$

   K_{ij} will be defined shortly.

2. ***Graph 2 (View, Thresh):***

   In a different scenario, we incorporate a threshold of $\epsilon = 5m$ distance, selected based on

empirical observations from psychological studies according to [21] to ensure that only nearby pedestrians within the field of view influence each other (Fig. 4.1):

$$w^{ij}_{view,thresh,t} = \begin{cases} K_{ij} & \Delta x^i_t \cdot \Delta x^j_t > 0, \quad \alpha^{ij}_t < \epsilon \\ 0 & \text{o.w.} \end{cases} \tag{4.7}$$

where $\alpha^{ij}_t = \|x^i_t - x^j_t\|$ is the distance between agent i and j.

3. ***Graph 3 (Approach):***

   In this scenario, when the distance between agents decreases, signifying that they are approaching each other and likely to influence each other, we define the kernel-based weight as:

$$w^{ij}_{appr,t} = \begin{cases} K_{ij} & \alpha^{ij}_{t+1} > \alpha^{ij}_t \\ 0 & \text{o.w.} \end{cases} \tag{4.8}$$

4. ***Graph 4 (View, Approach):*** We also consider a case that encompasses both of the above scenarios:

$$w^{ij}_{view,appr,t} = \begin{cases} K_{ij} & \Delta x^i_t \cdot \Delta x^j_t > 0, \quad \alpha^{ij}_{t+1} > \alpha^{ij}_t \\ 0 & \text{o.w.} \end{cases} \tag{4.9}$$

It is worth noting that [78] also considers the field of view and direction of movement. However, they consider each of the factors in separate graphs, and then use a multi-layer perceptron (MLP) to combine them into a single graph. In this work, we choose to combine the two conditions into a single graph to follow psychological studies and better explain the adjacency matrix.

Kernel Functions for Assigning Neighbors Importance Through Edge Weights: We employ kernel functions to capture the strength of interaction between pedestrians. We examine the impact of two distinct kernel functions on the model's accuracy in predicting pedestrian trajectories.

1. **Kernel 1 (Inverse of Norm):** Similar to [19], we use the inverse of the L_2 norm ($\alpha_t^{ij} = \|x_t^i - x_t^j\|$) as a similarity measure, considering that further away neighbors exert less influence:

$$K_{ij} = \begin{cases} \frac{1}{\alpha_t^{ij}} & \alpha_t^{ij} \neq 0 \\ 0 & o.w. \end{cases}$$

2. Alternatively, in this case, we use the exponential decay as the kernel function:

$$K_{ij} = \begin{cases} \exp(-\alpha_t^{ij}) & \alpha_t^{ij} \neq 0 \\ 0 & o.w. \end{cases}$$

to reflect that the weights decay as agents move away from each other, according to [21].

4.4.2 Model Architecture:

Our model leverages the same network architecture as introduced in [19], consisting of spatial and temporal components. The key difference from previous work is in the graph representation, which incorporates geometric kernels and neighborhood definitions inspired by psychological studies on crowd dynamics, as explained earlier. Fig. 4.2 provides a visual overview of the network architecture adopted in this paper.

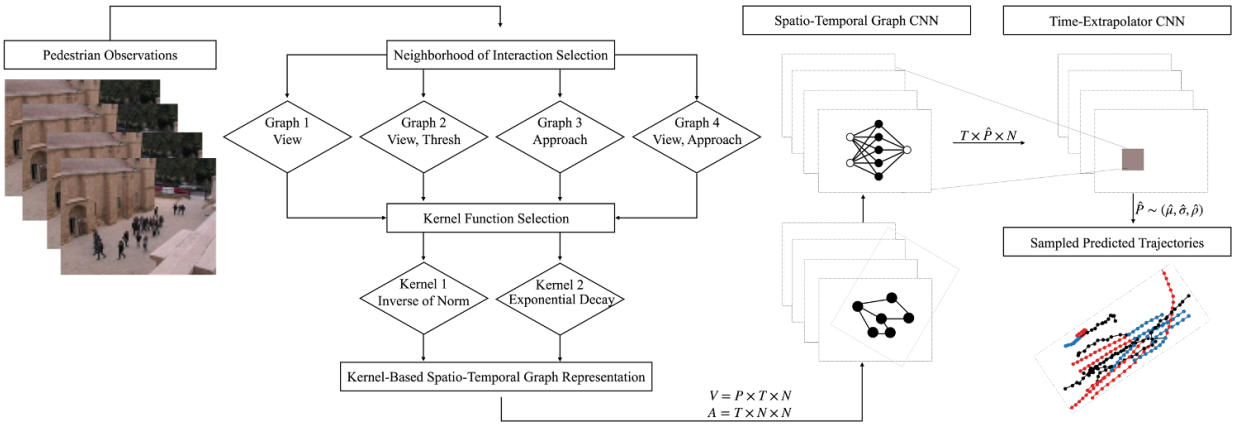


Figure 4.2: Illustration of our geometric GNN model. From observed pedestrian trajectories over T frames, we construct a kernel-based spatio-temporal graph by defining neighborhood relationships and weighting edges using kernel functions. This graph is processed by a spatio-temporal graph CNN, similar to the architecture proposed by [19], to generate a spatio-temporal embedding. A temporal CNN then predicts the distribution of future trajectories. V represents the set of nodes, $P = 2$ is the dimension of pedestrian positions, and $\hat{P}$ denotes the dimensions of the embedding from the predicted Gaussian distribution. Our approach integrates domain knowledge from psychological studies, resulting in a more context-aware and interpretable adjacency matrix.

1. **Kernel-Based Spatio-Temporal Graph Representation:** The first step in our network is to construct the graph representation of pedestrian interactions. We utilize the proposed neighborhood of interactions and kernel functions one by one and report the modeling accuracy for each of them.

2. **Spatio-Temporal Graph CNN:**

   In the second step, to extract meaningful features from the graph representation, we employ spatio-temporal graph convolutions, as described by [79]. This approach is particularly suitable for our problem as it effectively captures both spatial and temporal dependencies in pedestrian trajectories. The colvolution operation compute node embeddings by aggregating information from neighboring nodes in a recursive fashion using (4.5).

3. **Time-Extrapolator CNN:**

Taking the features from the previous step as input, this network applies temporal convolutions to predict probability distributions of future trajectories for all pedestrians within the time interval from $t = T_{\text{obs+1}}$ to $t = T_{\text{pred}}$.

4.4.3 Datasets and Evaluation Metrics

To train and assess the performance of our model, we utilize two well-established pedestrian datasets, in line with established methodologies in prior research. These datasets include: (1) ETH Dataset, collected by [80], with two scenes: ETH and Hotel, and (2) UCY Dataset, given by [81]: with 3 scenarios: ZARA-01, ZARA-02, and UNIV. In total, our model is evaluated across five distinct datasets, encompassing diverse nonlinear trajectories from 1536 pedestrians within real-world crowded environments. To ensure consistency in data representation across various datasets, we adopt the preprocessing approach outlined in Social-GAN by [75]. To facilitate fair comparisons, we employ a leave-one-out training methodology, as suggested [74]. This involves training and validating on four datasets and subsequently testing on the fifth. This procedure is consistently applied across the **four** interaction neighborhood scenarios and the **two** proposed kernel functions as well as other baseline methods for best evaluation. For the quantitative assessment of our approach, we employ two widely recognized metrics [74]:

Average Displacement Error (ADE) computes the average L2 distance between the ground truth and our model's predictions over all predicted time steps. It serves as a measure of trajectory accuracy along the path.

Final Displacement Error (FDE) quantifies the distance between the predicted final position and the ground truth final position at the end of the prediction period T_{pred}.

Given our assumption of Gaussian distribution for pedestrian trajectories, the estimated trajectory is a distribution. For fair comparison with previous work, we follow [19]'s evaluation approach. We generate 20 samples from the predicted trajectory distribution and compare the best-performing sample (closest to the ground truth) against the ground truth to calculate ADE and FDE. This evaluation approach has been widely adopted by previous studies, including those by [20, 75].

4.4.4 Training Setup:

Our training setup closely aligns with the architecture and parameters outlined in [19], ensuring a consistent baseline for evaluation. We employ the Parametric Rectified Linear Unit (PReLU) as our activation function, denoted by σ. The training batch size is set to 128 and the model is trained for a total of 250 epochs for each dataset. The learning rate is initialized at 0.01 and adjusted to 0.002 after 150 epochs to enhance the model's adaptability over the training duration. Stochastic Gradient Descent (SGD) is employed for optimization, in line with the methodology proposed by [19]. Consistent with the referenced work, our model comprises 1 layer of Spatio-Temporal Graph CNN and 5 layers of Temporal-Extrapolator CNN. This allows evaluating the impact of kernel functions and interaction topologies.

4.5 Results

The results presented in Table 4.1 show that our proposed methods, incorporating different neighborhood definitions and kernel functions, achieve competitive performance compared to previous approaches across various pedestrian datasets. Our model has $7,563$ parameters, with

Table 4.1: ADE/FDE Results for Various Methods across 5 Pedestrian Datasets

Method	**ETH**	**HOTEL**	**UNIV**	**ZARA01**	**ZARA02**	**Average**
social-LSTM ([74])	1.09/2.35	0.79/1.76	0.67/1.40	0.47/1.00	0.56/1.17	0.72/1.54
social-GAN ([75])	0.81/1.52	0.72/1.61	0.60/1.26	0.34/0.69	0.42/0.84	0.58/1.18
Social-BiGAT ([82])	0.69/1.29	0.49/1.01	0.55/1.32	**0.30**/0.62	0.36/0.75	0.48/1.00
social-STGCNN ([19])	**0.64**/1.11	0.49/0.85	**0.44/0.79**	0.34/0.53	0.30/0.48	0.44/0.75
Ours (Kernel 1: Inverse of Norm)						
- View (Graph 1- Kernel 1)	**0.64/1.09**	0.41/0.63	0.45/0.85	0.35/0.59	0.30/0.5	**0.43/0.73**
- View, thresh (Graph 2- Kernel 1)	0.76/1.48	0.40/0.69	0.45/0.85	0.33/**0.52**	0.29/0.47	0.44/0.80
- Approach (Graph 3- Kernel 1)	0.82/1.29	0.39/0.58	0.46/0.868	0.34/0.53	0.30/0.50	0.46/0.75
- View, approach (Graph 4- Kernel 1)	0.75/1.37	0.38/0.57	0.47/0.88	0.34/0.57	0.30/0.50	0.44/0.77
Ours (Kernel 2: Exponential Decay)						
- View (Graph 1- Kernel 2)	0.80/1.54	0.47/0.77	0.46/0.84	0.37/0.58	0.43/0.67	0.51/0.88
- View, thresh (Graph 2- Kernel 2)	0.80/1.40	0.47/0.77	0.45/0.82	0.32/0.53	**0.27/0.45**	0.47/0.80
- Approach (Graph 3- Kernel 2)	0.86/1.37	**0.34/0.47**	0.47/0.86	0.36/0.56	0.28/0.46	0.47/0.74
- View, approach (Graph 4- Kernel 2)	0.76/1.36	0.47/0.83	0.48/0.88	0.33/**0.52**	0.30/0.48	0.47/0.81

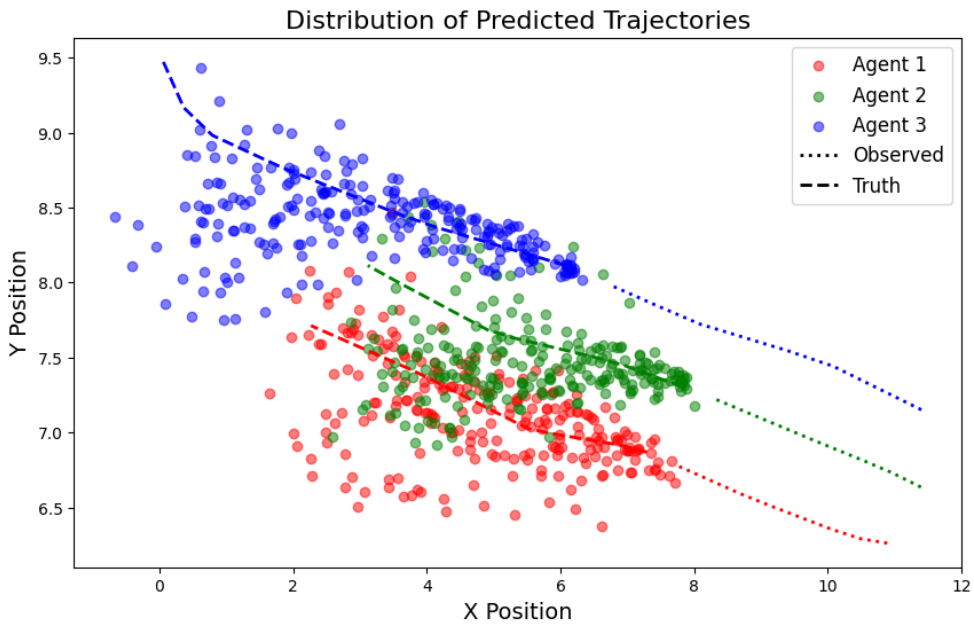


Figure 4.3: Visualizing the predicted distribution of pedestrian trajectories for a representative scene (UNIV). Sampled trajectories from the predicted Gaussian distribution are depicted as dot points in the scatter plot, while observed and true trajectories are shown as dashed and dotted lines respectively.

an average inference time of 0.002 seconds, based on multiple single inference steps.

On the ETH dataset, our method with Graph 1 (using pedestrian's field of view in neighborhood definition) and Kernel 1 (inverse of the norm) achieves the best performance, following the Social-STGCNN ADE of 0.64 and improving the FDE to 1.09. For the HOTEL dataset, our method with Graph 3 (using approach dynamics in neighborhood definition) and Kernel 2 (exponential decay) significantly outperforms all existing methods, achieving the best ADE of 0.34 and FDE of 0.47. On the UNIV dataset, the previous Social-STGCNN method by [19] maintains the best performance with an ADE of 0.44 and FDE of 0.79. However, our method with Graph 2 (using field of view and $5m$ distance threshold for neighborhood definition) and Kernel 2 closely follows the best performance with an ADE of 0.45 and FDE of 0.82. Other graph variations and kernel functions also show very close performance on this dataset, further highlighting the

importance of using geometric definitions. For the ZARA01 dataset, the Social-BiGAT method by [82] achieves the best ADE of 0.30, while our method with Graph 2 and Kernel 1 achieves the best FDE of 0.52. On the ZARA02 dataset, our method with Graph 2 and Kernel 2 outperforms all existing methods, achieving the best ADE of 0.27 and FDE of 0.45.

Overall, our method with Graph 1 and Kernel 1 achieves the best average ADE of 0.43 and FDE of 0.73 across all datasets, surpassing previous methods. Our results suggest that methods incorporating attention mechanisms, whether through softmax (e.g., social-BiGAT by [82]) or geometric kernel functions, consistently enhance accuracy, underscoring the importance of kernel functions for graph representations in GNN methods. Furthermore, our results highlight the effectiveness of integrating domain knowledge, such as pedestrian's field of view, $5m$ distance threshold, and approach dynamics, into neighborhood definitions for modeling human interactions in crowded environments. While [19] found no improvement with the exponential decay kernel, our results suggest that performance outcomes can vary and improve depending on the interaction neighborhood definition.

While no single method consistently outperforms others across all datasets, our proposed methods exhibit competitive performance. Performance discrepancies may arise from unaccounted factors like landmarks or scene information, highlighting the potential impact of external factors. Notably, [83] demonstrated significant accuracy improvements by integrating scene information into their graph attention network. However, our primary objective was to demonstrate how incorporating domain knowledge can enhance data-driven trajectory prediction, rather than finding the best-performing model. Thus, we maintained the same parameters as the referenced work.

4.6 Summary

In summary, this case study introduces a geometric GNN architecture featuring various kernel functions and interaction topologies inspired by psychological studies. This framework lays a foundation for better understanding and modeling of human interactions. Integrating a 180° field of view in the interaction neighborhood and employing the inverse of norm kernel function to weigh neighboring agents' importance contribute to overall accuracy improvements, reflected in enhanced ADE and FDE metrics across all datasets. Our results suggest that different topologies of interaction and kernel functions may perform better under different scenarios. However, determining the robust optimal interaction topologies that best explain complex interactions across domains remains an open challenge. To address the challenge of identifying optimal graph topology and interaction encodings, we propose an inverse optimal control (IOC) approach to uncover interaction strategies in dynamic graphs from behavioral observations. As the kernels simply parameterize the edge weights in the graph topology of multi-agent systems, we seek to investigate if they can be learned from data in a more rigorous fashion. In Chapter 5 and 6, we lay the foundations to achieve this objective.

Chapter 5: Stability and Robustness Conditions for Nonlinear Interaction Rules

5.1 Overview

This chapter develops directed potential-based interaction rules for complex networks. We provide a comprehensive framework for analyzing the stability and robustness of MAS whose interactions are governed by state-dependent edge weights that may be both positive (cooperative) and negative (antagonistic). By leveraging a Lyapunov-based analysis and an edge-based representation, we derive explicit algebraic and spectral conditions under which the weighted consensus dynamics converge—even in the presence of nonlinear, directed, and signed interactions. These results not only guarantee convergence to desired configurations but also provide robustness against disturbances. The stability guarantees established here serve as the theoretical foundation for the next chapter on learning interaction rules from data.

5.2 Introduction

Potential-based control methods using artificial potential energies or edge tension energies have been widely used for multi-agent coordination (e.g., [38]). Traditional analyses assume undirected graphs with positive edge weights, simplifying the analysis since the graph Laplacian remains positive semidefinite. However, many real-world systems exhibit directed

interactions—often due to limited sensing—and can include repulsive interactions represented by negative weights, complicating analysis since the Laplacian may no longer remain positive semidefinite.

Recent work has addressed potential-based control in directed networks by employing the *edge Laplacian*—a matrix that encodes flows of information among agents and elucidates how spanning trees and cycles influence convergence [39]. In [22, 40], the authors extend potential-based methods to directed networks via a structural Lyapunov matrix, whose positive semidefiniteness characterizes stable directed topologies. We follow a similar philosophy, employing an energy-based Lyapunov analysis in edge space, and provide the following specific contributions: (1) Introduction of an edge-based potential energy framework that explicitly accommodates positive and negative edge weights for digraphs. (2) Derivation of algebraic and spectral conditions on network topology and edge weights ensuring convergence to general desired configurations. (3) A rigorous Lyapunov-based analysis that explicitly addresses directed, nonlinear, signed interactions via the edge Laplacian. These results highlight how designing interaction weights appropriately ensures stability, and how the edge Laplacian helps unify and extend classical potential-based methods to general directed, signed networks.

5.3 Preliminaries

This section establishes the notation, presents the graph representations, and describes the multi-agent dynamics that form the basis for our analysis for the next two chapters.

5.3.1 Notation

We use $\mathbb{R}$ to denote the set of real numbers, $\mathbb{R}^d$ to denote the set of real vectors with d elements, and $\mathbb{R}^{n\times m}$ to represent the set of real $n\times m$ matrices. The vector of all ones in $\mathbb{R}^d$ is denoted by $\mathbf{1}_d$. Let $I_d \in \mathbb{R}^{d\times d}$ denote the $d\times d$ identity matrix, and let $\mathbf{1}_S(x(t))$ be the indicator function that returns 1 if $x \in S$ and 0 otherwise. When the input to the indicator function is vector-valued, we assume that it is applied element-wise. For a given set S, $|S|$ refers to its cardinality. The square diagonal matrix formed from the elements in a vector a is written as $\mathrm{diag}(a)$. For any vector a, $[a]_k$ represents its k-th element, and a^T denotes its transpose. For a matrix A, $[A]_{ik}$ is the entry in the i-th row and k-th column. The Euclidean distance (ℓ^2 norm) between two vectors a and b is $\|a-b\|$. The Kronecker product is represented by $\otimes$, and the Hadamard (element-wise) product by $\odot$. We denote by $\sigma(A)$ the set of singular values of a matrix $A \in \mathbb{R}^{N\times m}$. The set of continuously differentiable functions is C^1, and $\mathcal{L}^1([\delta,\Delta])$ denotes the set of Lebesgue-integrable functions over $[\delta,\Delta]$. The symbols $\mathcal{N}(\cdot)$ and $\mathcal{R}(\cdot)$ denote the null space and range space of the matrix, respectively.

5.3.2 Graph Theory

A weighted directed graph $G = (\mathcal{V},\mathcal{E},w)$ is a tuple consisting of a set of nodes $\mathcal{V}$, an ordered edge set $\mathcal{E}$, and a weight function w. The node set is given by $\mathcal{V} = \{1,\ldots,N\}$, where each node represents an agent in the system. The edge set $\mathcal{E} \subset \mathcal{V}\times\mathcal{V}$ defines the communication topology (connectivity among agents), where an edge $e = (i,j)$ denotes a directed connection from node j (the tail) to node i (the head). The set of in-neighbors of node i, i.e., the nodes that have directed edges pointing to i is denoted by $\mathcal{N}_i^{\mathrm{in}} = \{j \mid (j,i) \in \mathcal{E}\}$. A directed path is a sequence

of consecutive edges that follow the direction of connectivity. A directed tree is a subgraph where every node, except a designated root, has exactly one incoming edge, ensuring that the root has no parent but maintains a path to all other nodes. A spanning tree is a tree subgraph that connects all nodes in $\mathcal{V}$.

The real-valued weight function $w : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ assigns a scalar (positive or negative) value to each edge, representing how strongly agents influence their neighbors. The weights of all edges are often collected in a diagonal matrix $\mathbb{W} \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|}$, such that for an edge $e_k = (j,i) \in \mathcal{E}$ the corresponding diagonal entry is $[\mathbb{W}]_{kk} = w_k = w_{ij}$. The in-degree and out-degree of each node are captured in the diagonal matrices $\mathbf{\Delta}^{\text{in}}, \mathbf{\Delta}^{\text{out}} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$, defined as $(\mathbf{\Delta}^{\text{in}})_{ii} = \sum_{\{j:(j,i)\in\mathcal{E}\}} w_{ij}$ and $(\mathbf{\Delta}^{\text{out}})_{ii} = \sum_{\{j:(i,j)\in\mathcal{E}\}} w_{ij}$. The Laplacian matrix in this work is defined in terms of the in-degree matrix $L = \mathbf{\Delta}^{\text{in}} - A$, where the weighted adjacency matrix A is given by $A_{ij} = w_{ij}$, if $(j,i) \in \mathcal{E}$, and 0 otherwise. Alternatively, using incidence matrices, the in-degree graph Laplacian is expressed as $L = D^{\text{in}} W D^T$, where the incidence matrices D^{in} and D^{out} identify directed edges entering and leaving nodes, respectively. They are defined as

$$[D^{\text{in}}]_{ik} = \begin{cases} 1, & \text{if } e_k = (j,i) \in \mathcal{E}, \\ 0, & \text{otherwise,} \end{cases} \qquad [D^{\text{out}}]_{ik} = \begin{cases} 1, & \text{if } e_k = (i,j) \in \mathcal{E}, \\ 0, & \text{otherwise.} \end{cases}$$

The full incidence matrix is then given by $D = D^{\text{in}} - D^{\text{out}}$.

The weighted edge-Laplacian of a digraph is given by $L_{e_w} = D^T D^{\text{in}} \mathbb{W}$ [47]. The spectral properties of edge Laplacian for a non-singular positive edge weights have discussed extensively in [22]. In particular, for a weighted digraph with positive edge weights, L_{e_w} shares the same non-zero eigenvalues as L. Moreover, in a weighted digraph with a rooted out-branching and

positive edge weights, the algebraic and geometric multiplicity of the zero eigenvalue of L_{e_w} are both equal to $m-N+1$, i.e., $dim[\mathcal{N}(L_{e_w})] = m-N+1$ [22]. For a directed acyclic graph and a weighted digraph with a single globally reachable node, $\mathcal{N}(L_{e_w}) = \mathcal{N}(D^{in}\mathbb{W})$. If there are $r < N$ nodes in the digraph, each with an in-degree of at least one, then the dimension of null space of L_{e_w} is $dim[\mathcal{N}(L_e)] = m-r$. Additionally, if a digraph contains multiple globally reachable nodes, they must form directed cycles and $\mathbf{1}_N \in \mathcal{R}(D^{in})$. For a quasi-strongly connected digraph (with positive edge weights), L_{e_w} has exactly $N-1$ non-zero eigenvalues, all in the open right-half plane, with $rank(L_{e_w}) = rank(L) = N-1$ as shown in [46].

For directed graphs with strictly positive edge weights, $w_k > 0$, it has been established that the graph Laplacian matrix L has a single zero eigenvalue, with all the other non-zero eigenvalues in the open right-half complex plane if and only if the digraph has a rooted out-branching) [5]). A node $i \in \mathcal{V}$ that can reach every other node through a directed path is termed a *globally reachable node*. A digraph G is classified as a *rooted-out-branching* if it contains at least one globally reachable node and has no directed cycles. If a graph with cycles contains a globally reachable node, a spanning tree $G_\tau \subseteq G$ exists, where the root (globally reachable node) has in-degree of zero, and all other nodes have in-degree of one. The remaining edges form a disjoint subgraph G_c, ensuring $G = G_\tau \cup G_c$. By appropriate permutation of edge indices (we follow the convention in [22]), the incidence matrix can be written as

$$D = [D_\tau \quad D_c],$$

where $D_\tau \in \mathbb{R}^{N(N-1)}$ corresponds to the incidence matrix of a given spanning tree, G_τ and $D_c \in \mathbb{R}^{N(m-N+1)}$ corresponds to the co-spanning tree from the remaining edges (i.e., cycle edges).

Since D_τ is full column-rank, its pseudo-inverse exists, and the co-spanning tree edges can be reconstructed from spanning tree edges via a linear transformation

$$D_\tau T = D_c, \quad \text{where } T := (D_\tau^T\, D_\tau)^{-1}\, D_\tau^T\, D_c. \tag{5.1}$$

So, the incidence matrix of digraph can alternatively be written as

$$D = D_\tau\, R \quad \text{where} \quad R = [I_{N-1} \quad T] \tag{5.2}$$

Given this partitioning, the following lemma follows for the unweighted edge Laplacian of a graph with rooted out-branching:

Lemma 5.1 ([22])**.** *For a directed graph that contains a rooted out-branching (i.e., a directed spanning tree), the unweighted edge Laplacian can be expressed in block form as*

$$L_e \sim \begin{bmatrix} D_\tau^T D^{\text{in}} R^T & D_\tau^T D^{\text{in}} N_\tau \\ 0_{(m-N+1)\times(N-1)} & 0_{(m-N+1)\times(m-N+1)} \end{bmatrix}. \tag{5.3}$$

Here, the columns of N_τ form the orthonormal basis for the nullspace of the matrix R, $\mathcal{N}(R)$, i.e., $R\, N_\tau = 0$ and $N_\tau^T\, N_\tau = I_{m-N+1}$.

Proof. Define the transformation matrix $\mathcal{T} = [R^T \quad N_\tau]$ which is square and invertible since R^T is full column rank and its left inverse exist as $(R\, R^T)^{-1} R$. Plus, N_τ completes it to form a basis of $\mathbb{R}^m$. Then, the inverse can be written as $\mathcal{T}^{-1} = \left[\left((R\, R^T)^{-1} R\right)^T \quad N_\tau\right]^T$. Performing a similarity

transformation using $\mathcal{T}$, we have

$$\mathcal{T}^{-1} L_e \mathcal{T} = \begin{bmatrix} D_\tau^T D^{\text{in}} R^T & D_\tau^T D^{\text{in}} N_\tau \\ 0_{(m-N+1)\times(N-1)} & 0_{(m-N+1)\times(m-N+1)} \end{bmatrix}. \tag{5.4}$$

□

This block-diagonalization in lemma 5.1 separates the dynamics associated with the spanning tree (upper left block) from the cycle dynamics (lower right block, which is zero). This reduced-order representation is especially useful for stability and convergence analysis, as it shows that the essential dynamics are captured by the matrix $D_\tau^T D^{\text{in}} R^T$. For more details on the partitioning, refer to [22].

Assumption 5.1. *In this work, we assume digraphs with at least a single globally reachable nodes (i.e., a spanning tree exist), since consensus is achievable only in such digraphs [84].*

5.3.3 Multi-Agent Dynamics and Interaction Representation

Consider a system of N agents, each with state vector $x_i(t) \in \mathbb{R}^d$ at time t. The ensemble state of the system is

$$x(t) = [x_1^T(t), \ldots, x_N^T(t)]^T \in \mathbb{R}^{dN}.$$

The interactions among agents are captured by a signed weighted digraph $G = (\mathcal{V}, \mathcal{E}, w)$, with edge weights w_k that depends on the state (e.g., relative distance) and control policy $u(s)$, that will be described in the next section. The number of nodes and edges are given by $N = |\mathcal{V}|$, and $m = |\mathcal{E}|$, respectively. The dynamics are modeled using weighted consensus dynamics (i.e.,

"agreement control"). At the node level, for each $i \in \mathcal{V}$, the dynamics are given by:

$$\dot{x}_i(t) = \sum_{k=(j,i)\in\mathcal{E}} w_k\Big(\alpha_k\big(x(t)\big), u_k(s)\Big)\big(x_j(t) - x_i(t)\big), \quad \alpha_k = \|x_i(t) - x_j(t)\|. \tag{5.5}$$

This equation describes how agent i updates its state based on the weighted differences with its neighbors. In compact form, the ensemble dynamics are

$$\dot{x}(t) = -\left[L^{\text{in}}\Big(w\big(\alpha\big(x(t)\big), u(s)\big)\Big) \otimes I_d\right] x(t). \tag{5.6}$$

Given the state-dependency of graph Laplacian $L^{\text{in}}\Big(w\big(\alpha\big(x(t)\big), u(s)\big)\Big)$, the usual eigen-value analysis of the graph Laplacian is insufficient for the stability analysis. In this case, an edge-based representation may be more appropriate as it can decouple the communication topology from the state-dependent edge weights. This representation, detailed in the following sections, is particularly advantageous for systems with heterogeneous communication structures.

5.3.3.1 Edge Dynamics

Following [5], we define the coordinate transformation from node states to edge states as:

$$z(t) = [D^T \otimes I_d]x(t), \quad z(t) = [z_1^T(t), \ldots, z_m^T(t)]^T \in \mathbb{R}^{dm} \tag{5.7}$$

where $z_k = x_i - x_j$ represents the difference in state between agents connected via an edge $e_k = (j, i)$. This representation transforms the consensus dynamics into the edge space, yielding:

$$\dot{z}(t) = -[D^T \; D^{in} \; \mathbb{W} \otimes I_d] \; z = -[L_{e_w} \otimes I_d] \; z. \tag{5.8}$$

Using tree-cycle partitioning as in (5.3.2) for a graph containing a spanning tree, we can partition the edge state as

$$z = [z_\tau^T \quad z_c^T]^T, \quad z_\tau \in \mathbb{R}^{d(N-1)}, \quad z_c \in \mathbb{R}^{d(m-N+1)},$$

where z_τ and z_c correspond to the edge states of the spanning and co-spanning tree, G_τ and G_c. Let $\mathbf{D}^{\text{in}} = D^{\text{in}} \otimes I_d$, $\mathbf{D} = D \otimes I_d$, $\mathbf{R} = R \otimes I_d$, $T = \mathbf{T} \otimes I_d$, and $\mathbf{W} = \mathbb{W} \otimes I_d$, $\boldsymbol{\mathcal{T}} = \mathcal{T} \otimes I_d$, using (5.1) and (5.2), the following are true:

$$z_\tau = \mathbf{D}_\tau^T\, x, \quad z_c = \mathbf{T}^T\, z_\tau, \quad z = \mathbf{R}^T\, z_\tau, \tag{5.9}$$

Let $q = \boldsymbol{\mathcal{T}}^{-1} z$, and $\mathbf{N}_\tau = N_\tau \otimes I_d$ where $\mathcal{T}$ is introduced in (5.4), we can see that

$$q = \left[\left((\mathbf{R}\,\mathbf{R}^T)^{-1}\mathbf{R}\right)^T \quad \mathbf{N}_\tau\right]^T \mathbf{R}^T\, \mathbf{D}_\tau^T\, x = [x^T \mathbf{D}_\tau \quad \mathbf{0}_{m-n+1}^T]^T, \tag{5.10}$$

which implies that the first $n-1$ components of q represent the rooted-out-branching, so, the edge agreement in (5.8) can be captured by the following reduced order system

$$\dot{z}_\tau = -\mathbf{D}_\tau^T\, \mathbf{D}^{in}\, \mathbf{W}\, \mathbf{R}^T\, z_\tau, \tag{5.11}$$

which will be convenient for analyzing the full system behavior. The consensus problem with positive edge weights and in particular $\mathbb{W} = I_m$ has been well studied in the literature. It is established that consensus under directed network is reached if and only if $z_t \to 0$ as $t \to \infty$ as discussed in [5, 22, 47]. However, when dealing with nonlinear edge weights, the convergence properties become more complex [49]. To the best of our knowledge, no prior study has explored

multi-agent consensus in the edge-space framework for formation-like control under nonlinear, signed edge weights.

5.3.3.2 Interaction Weight Design

For each edge $e_k = (j,i) \in \mathcal{E}$, the interaction weights capture the dependence on both a state-dependent term (e.g., distance between agents i and j) and a control variable. We define the interaction weights as follows:

$$[w(\alpha(x(t)),u(s))]_k = \int_{\delta}^{\alpha_k(x(t))} u_k(s)\,ds = \int_{\delta}^{\Delta} u_k(s)\mathbf{1}_{\{s<\alpha_k(x(t))\}}\,ds, \tag{5.12}$$

where $\mathbf{1}_{\{s<\alpha_k(x(t))\}}$ is the indicator function. Defining the weights in this fashion guarantees that they are continuous and differentiable. The diagonal weight matrix is given by $\mathbb{W}(\alpha(x),u(s)) = \operatorname{diag}(w(\alpha(x),u(s))) \in \mathbb{R}^{m\times m}$. The control variable (policy), $u_k(s)$ is a non-parametric, scalar-valued function for any $0 \le \delta \le s \le \Delta < \infty$. We assume that $u_k(s) \in \mathcal{L}^1([\delta,\Delta])$ and $[\alpha(x(t))]_k = \alpha_k(x(t)) = \|x_i(t) - x_j(t)\|$ represents the Euclidean distance between agents (nodes) i and j at edge $k = (j,i) \in \mathcal{E}$.

Let $\phi_k(s) := u_k(s)\mathbf{1}_{\{s<\alpha_k(x(t))\}}$, $\phi = \{\phi_1,\ldots,\phi_m\}$ and using the definition provided for interaction weight, the dynamics can be written as:

$$\dot{x}_i(t) = -\int_{\delta}^{\Delta} \sum_{\{k\in\mathcal{E}\,|\,k=(j,i)\}} \phi_k(s)\Big(x_i(t) - x_j(t)\Big)\,ds. \tag{5.13}$$

We can similarly rewrite the dynamics at the ensemble-level:

$$\dot{x}(t) = f(x(t),\, u(s)) = \int_{\delta}^{\Delta} \Big[\mathbf{D}^{\text{in}}\Big(\text{diag}\big(\phi(s)\big) \otimes I_d\Big)\mathbf{D}^T x(t)\Big] ds, \tag{5.14}$$

Allowing the edge weights, defined in (5.12), to take both positive and negative values enables the modeling of attractive as well as repulsive interactions that may arise among MAS interaction. Such dual behavior is common in MAS and can be imposed through distance-based artificial potential functions $\mathcal{P}(x(t)) \in \mathbb{R}_{\geq 0}$ (also referred to as edge tension energy [5]), whose critical points correspond to desired configurations. In this framework, the control law is designed as a gradient descent controller on $\mathcal{P}(x(t))$. Although this approach is relatively straightforward for undirected networks, its extension to directed networks requires additional considerations regarding the topology and edge weight properties.

To connect the interaction weights with the potential energy, we define the edge-based potential energy on edge $k = (j,i)$, with relative state $z_k = x_i - x_j$, by

$$\mathcal{P}_k(\alpha_k) = \mathcal{P}_{ij}(\alpha_k) = \int_{\delta}^{\alpha_k} s\, w_k(s)\, ds, \quad \text{with } \alpha_k = \|z_k\|. \tag{5.15}$$

Since w_k is continuous, this function is continuously differentiable. The global potential energy is then given by

$$\mathcal{P}(z) = \sum_{k=(j,i)\in\mathcal{E}} \mathcal{P}_k(\|z_k\|).$$

Using the chain rule, the gradient of the potential with respect to the edge state is

$$\frac{\partial \mathcal{P}_{ij}}{\partial z_i} = \frac{\partial \mathcal{P}_{ij}(\|z_k\|)}{\partial \|z_k\|} \frac{z_k^T}{\|z_k\|} = w_k(\|z_k\|) z_k^T,$$

so, the edge weight function may also be expressed as

$$w_k(\|z_k\|) = \frac{d\mathcal{P}_k(\|z_k\|)/d\|z_k\|}{\|z_k\|}. \tag{5.16}$$

Recall

$$u_k(s) = \frac{d}{ds} w_k(s) \Longrightarrow w_k(s) = w_k(s') + \int_{s'}^{s} u_k(\bar{s})\, d\bar{s}, \tag{5.17}$$

and that

$$\frac{d\mathcal{P}_k(s)}{ds} = s\, w_k(s). \tag{5.18}$$

Substituting the integrated form of $w(s)$ from (5.17) we obtain:

$$\frac{d\mathcal{P}_k(s)}{ds} = s\left[w_k(s') + \int_{s'}^{s} u_k(\bar{s})\, d\bar{s}\right]. \tag{5.19}$$

Integrating with respect to s (with a variable of integration t), we find:

$$\mathcal{P}_k(s) = \mathcal{P}_k(s') + \int_{s'}^{s} t\left[w(s') + \int_{s'}^{t} u_k(\bar{s})\, d\bar{s}\right] dt, \tag{5.20}$$

which becomes

$$\mathcal{P}_k(s) = \mathcal{P}_k(s') + \frac{w_k(s')}{2}\left(s^2 - s'^2\right) + \int_{s'}^{s} t\left(\int_{s'}^{t} u_k(\bar{s})\, d\bar{s}\right) dt, \tag{5.21}$$

which suggest a quadratic component in the potential energy with the double-integral term capturing the cumulative adjustments arising from the nonlinear state-dependent variation of the weight.

Assumption 5.2. *For every edge, k, with relative state $z_k = x_i - x_j$ and $\alpha_k = \|z_k\|$, we assume there exists a continuously differentiable function $\beta_k(\cdot) : [\delta, \Delta] \to \mathbb{R}$ such that the per-edge potential energy in (5.15) satisfies*

$$\mathcal{P}_k(\alpha_k) \geq \beta_k(\alpha_k) \quad \forall\, \alpha_k \in [\delta, \Delta]. \tag{5.22}$$

Such a function exists since $\mathcal{P}_{ij}$ is continuously differentiable over a bounded domain. We further assume that there exist a z^* at which the edge potential achieves its minimum, so $\beta_k(\|z_k^*\|)$ is a lower bound for $\mathcal{P}_{ij}$, and also $w_k(z_k^*) = 0$, so z^* is a critical point.

Assuming that $s^* = \text{argmin}_s \mathcal{P}_k(s)$, it follows that

$$w_k(s^*) = \frac{1}{s^*} \frac{d\mathcal{P}_k(s^*)}{ds} = 0, \tag{5.23}$$

and from (5.17) (with $s = s^*$):

$$0 = w_k(s^*) = w_k(s') + \int_{s'}^{s^*} u_k(\bar{s})\, d\bar{s} \Longrightarrow w(s') = -\int_{s'}^{s^*} u_k(\bar{s})\, d\bar{s}. \tag{5.24}$$

Assumption 5.2 ensures that even if the edge weights $w_k(s)$ might be negative for certain values of s, the cumulative potential energy is bounded below. In other words, the cumulative effect of the interaction along each edge is prevented from decreasing without bound. This is crucial for ensuring local exponential stability and for robust convergence of the system. Given equations (5.15) and (5.16), function β_k provides a lower bound on the allowable values of w_k:

$$w_k(\rho) \geq \frac{1}{\rho} \frac{\partial \beta_k(\rho)}{\partial \rho}. \tag{5.25}$$

5.4 Problem Formulation

The primary goal of this chapter is to establish rigorous conditions on the interaction weights and the network topology that guarantee the convergence and robustness of the interaction control. Specifically, we ask:

Problem 5.4.1. ***(Stability of Weighted Consensus with Signed Edge Weights)*** *Consider the multi-agent dynamics defined in* (5.14)*, where the weight associated with each directed edge* (j,i) *is given by* (5.12) *and it may take both positive and negative values. The problem is to determine under what conditions on* graph topology *and* interaction weights *we can ensure that the system is stable and robust. Specifically, we seek to establish:(1) Stability conditions ensuring convergence to desired behavior (e.g., desired configuration,* z^**). (2) Robustness guarantees in the presence of bounded disturbances.*

5.5 Stability and Robustness Analysis

In this section, we establish general Lyapunov-based conditions under which the MAS (5.14) remains stable and converges to the largest invariant set, even when some interaction weights may be negative. We first treat the general digraph case and discuss how cycles affect convergence by enlarging the space of possible limit points. We then present the simplified directed tree case, where there are no cycles and the corresponding invariant set reduces substantially, guaranteeing stronger convergence statements. We conclude with a local exponential stability theorem that leverages the local convexity of the potential function near its minima, and we provide robustness guarantees to bounded disturbances.

5.5.1 Lyapunov Analysis via Edge-Based Potential Energy

Let the Lyapunov function, $V(z) = \mathcal{P}(z) - \beta$, where $\beta = \min_z \mathcal{P}(z) = \mathcal{P}(z^*)$, where z^* is the desired configuration. The derivative of the Lyapunov function is $\dot{V}(z) = \frac{\partial V(z)}{\partial z}\ \dot{z}$ can be derived in compact form as

$$\dot{V}(z) = -z^T\ \mathbf{W}\ \mathbf{D}^T\ \mathbf{D}^{\text{in}}\ \mathbf{W}\ z, \tag{5.26}$$

Let $y = \mathbf{W}\ z$, then the above equation becomes

$$\dot{V}(x) = -y^T\ [\mathbf{D}^T\ \mathbf{D}^{\text{in}}\ ]\ y, \tag{5.27}$$

where $L_e = \mathbf{D}^T\ \mathbf{D}^{\text{in}}$ is the edge Laplacian for an unweighted digraph. We can see that (5.27) is in quadratic form and its characteristics depend on the symmetric part of the edge Laplacian, given the asymmetry of L_e:

$$\dot{V}(x) = -\frac{1}{2} y^T\ [\mathbf{D}^T\ \mathbf{D}^{\text{in}}\ + \mathbf{D}^{\text{in}^T}\ \mathbf{D}]\ y, \tag{5.28}$$

The negative semi-definiteness of $\dot{V}(x)$ suggests that the system is stable in the sense of Lyapunov in that if the energy is finite at time $t = t_0$, it remains finite for all $t > t_0$. By LaSalle's Invariance Principle, the system converges to the largest invariant set where $\dot{V}(x) = 0$. Given assumption 5.1, and the similarity transformation in (5.4), L_e is a positive semi-definite matrix whose zero eigenvalues is associated with the cycles in the graph, $\dim(\mathcal{N}(L_e)) = m - N + 1$ and it has $N - 1$ non-zero eigenvalues corresponding to the tree structure as shown in [47]. The largest invariant set where $\dot{V} = 0$ corresponds to the $\mathcal{N}(L_e)$, i.e., $\Gamma = \{y \mid y \in \mathcal{N}(L_e)\}$.

It is worth noting that under special choice of graph topology the Lyapunov function

becomes strict. A particularly important special graph topology occurs when the digraph is a directed tree, meaning that there is exactly one globally reachable node (the root) and no directed cycles. This topology has applications in leader-follower scenarios [47]. In that scenario, we have:

$$\dim(\mathcal{N}(L_e)) \;=\; m-(N-1) \;=\; 0,$$

and, $G = G_\tau$, and our Lyapunov function becomes strict and we can focus on the spanning tree component of the graph as shown previously in the literature [22, 49]. In fact, in this case $R = I_{N-1}, z = z_\tau$, and $\mathbf{D} = \mathbf{D}_\tau$. Let $y_\tau = \mathbf{W}\, z_\tau$. Using the same similarity transformation as in (5.4), equation (5.26) can be written as

$$\dot{V} = -z_\tau^T\, \mathbf{R}\, \mathbf{W}\, \mathbf{D}_\tau^T\, \mathbf{D}^{\text{in}}\, \mathbf{R}^T\, \mathbf{W} R^T z_\tau = -y_\tau^T\, L_{e_\tau}\, y_\tau, \tag{5.29}$$

where L_{e_τ} is the top-left (or spanning tree) block of the transformed edge Laplacian in (5.3). As discussed in section 5.3.2 and proved in [22], the eigenvalues of L_{e_τ} are in the open right half-plane, so $\dot{V}$ is negative semi-definite, even if some of the edge weights may have negative values. So, we have

$$\dot{V} \leq -\|y_\tau\|^2 \lambda_{\min}(L_{e_\tau}),$$

Hence, the system will converge to the largest invariant set, $M = \{z \in G \;\mid\; \dot{V} = 0 \;\; and \;\; \dot{z} = 0\}$ by LaSalle's invariance principle [43]. Additionally, $\dot{V} = 0$, when $y_\tau = 0$ which implies that $\mathbf{W}(z_\tau)\; z_\tau = 0$. So, the largest invariant set will become $\Omega = \{z_\tau \in \mathbb{R}^{dm} \mid \mathbb{W}\; z_\tau = 0\}$. Note that when $z = 0$, we will achieve consensus, and $z_\tau = z_\tau^*$ corresponds to the critical points of $\mathcal{P}(z_\tau)$. This result is valid for any potential energy function. In the next section, we provide sufficient

conditions for converging to the desired configuration, z_τ^*, under directed tree topology.

5.5.2 Local Exponential Convergence to Desired Behavior

While the above Lyapunov arguments show global convergence to the largest invariant set, we next establish local exponential convergence to the desired behavior. Although we have a weak Lyapunov function, we get to choose the edge weights. So, next we establish the conditions on edge weights under which the system is converging to the desired configurations. Suppose there exists a configuration z^* for which $\mathbb{W}(z^*)\, z^* = 0$. This z^* could represent connectivity maintenance or collision avoidance [48] in multi-robot systems.

Lemma 5.2. *The Jacobian of the dynamics in equation* (5.14) *takes the form of a weighted Laplacian. In particular, the Jacobian can be expressed as*

$$\frac{\partial f(x)}{\partial x} = -\mathbf{D}^{\text{in}} \mathcal{W}(z) \mathbf{D}^T, \tag{5.30}$$

similarly, the Jacobian of dynamics in edge space (5.8) *is given by*

$$\frac{\partial f(z)}{\partial z} = -\mathbf{D}^T \mathbf{D}^{\text{in}} \mathcal{W}(z), \tag{5.31}$$

where $\mathbf{D}^{\text{in}}$ *is the incidence matrix (appropriately extended for high-dimensional systems) for a weighted digraph, and* $\mathcal{W}$ *is a block-diagonal matrix given as*

$$\mathcal{W}(z) = blkdiag\left(w_1(\|z_1\|) I_d + \frac{dw_1(\|z_1\|)}{ds} \frac{z_1 z_1^T}{\|z_1\|}, \ldots, w_m(\|z_m\|) I_d + \frac{dw_m(\|z_m\|)}{ds} \frac{z_m z_m^T}{\|z_m\|} \right), \tag{5.32}$$

this structure directly links the network's topology with its stability properties.

Proof. Refer to appendix. □

Theorem 5.1. *Consider the multi-agent dynamics*

$$\dot{x} = -\Big[L^{\text{in}}(w(\alpha(x), u(s))) \otimes I_d\Big]\, x,$$

with the edge state defined by $z = \mathbf{D}^T x$*, and the nonlinear weight functions are collected in the block-diagonal matrix* $\mathbf{W} = [\mathbb{W}(z) \otimes I_d]$*. Suppose there exists a desired configuration* z^* *such that* $\mathbf{W}(z^*)\, z^* = 0$*. Assume the directed graph* $G = (\mathcal{V}, \mathcal{E}, w)$ *follows assumption 5.1. Further, suppose that for each edge* k *the weights satisfy* $\frac{dw_k(\|z_k\|)}{d\|z_k\|} > 0$ *for all* z_k *in a neighborhood of* z_k^**, then the system is locally exponentially stable in the subspace orthogonal to the invariances. Equivalently, small deviations* $z_k - z_k^*$ *in directions orthogonal to* z_k^* *experience a positive incremental pull, driving the network state back toward* z^**.*

Proof. Note that the transformed dynamics is given by

$$\dot{z} = f(z) = -L_e\, \mathbf{W}(z)\, z \tag{5.33}$$

Define the error variable $\xi = z - z^*$. We would like $\xi \to 0$. Using equation (5.33), the error

dynamics become

$$
\begin{aligned}
\dot{\xi} = \dot{z} - \dot{z}^* &= -L_e\ \mathbf{W}(z)\ z\ +\ L_e\ \overbrace{\mathbf{W}(z^*)z^*}^{\text{By design} = 0} \\
&= -L_e \mathbf{W}(z)\ (z - z^*) = -L_e \mathbf{W}(z)\xi = F(\xi) \\
&= -L_e \mathbf{W}(\xi + z^*)\xi = f(z) = f(z^* + \xi)
\end{aligned}
\tag{5.34}
$$

Linearizing around $\xi = 0$ (which corresponds to $z = z^*$ and $f(z^*) = 0$), the linearized error dynamics:

$$
\dot{\xi} = \left.\frac{\partial f(z)}{\partial z}\right|_{z=z^*} \xi + \underbrace{\epsilon(\xi)}_{\textit{Higher Order Terms}},
$$

with $f(z) = \mathbf{D}^T\ f(x) = \mathbf{D}^T\ \mathbf{D}^{\text{in}}\ \mathbf{W}\ z = -L_e\ \mathbf{W}(z)\ z$ as shown in equation (5.33), we have

$$
\left.\frac{\partial f(z)}{\partial z}\right|_{z=z^*} = \mathbf{D}^T\ \left.\frac{\partial f(x)}{\partial z}\right|_{z=z^*} = -L_e\ \text{blkdiag}\left(w_k(z^*)\, I_d + \frac{dw_k(z^*)}{ds}\frac{z_k^* z_k^{*T}}{\|z_k^*\|}\right) = -L_e \mathcal{W}(z^*).
$$

For $\xi \in B_\delta(z^*)$, the higher order terms vanish as $\|\xi\| \to 0$, so, for $\dot{\xi} = 0$, we need $\left.\frac{\partial f(z)}{\partial z}\right|_{z=z^*} \xi = 0$. Two cases arise:

$$
\begin{cases}
\left.\frac{\partial f(z)}{\partial z}\right|_{z=z^*} = 0 \Rightarrow L_e\ \mathcal{W}(z^*) = 0 \Rightarrow \begin{cases} w_k(z^*) = 0 \textit{ (which is true) and } \frac{dw_k(z^*)}{ds} => 0 \\ W(z^*)z^* \in \mathcal{N}(L_e) \textit{ (local minima)} \end{cases} \\
\frac{z_k^* z_k^{*T}}{\|z_k^*\|}(z_k - z_k^*) = 0 \quad \Rightarrow \quad (z_k - z_k^*) \perp z_k^*
\end{cases}
\tag{5.35}
$$

One way to achieve local exponential stability is for each edge weight $w_k(z)$ to be monotically increasing in a neighborhood of $\|z^*\|$, implying $\frac{dw_k(\|z^*\|)}{ds} > 0$ (i.e., the potential is locally convex),

small deviations $z_k - z_k^*$ in directions orthogonal to z_k^* have a positive incremental effect. In fact, for a neighborhood $\Omega_\delta = \{z : \|z - z^*\| \leq \delta\}$, with a sufficiently small $\delta > 0$, z^* is locally asymptotically stable given that $\mathcal{P}(z)$ achieves its minimum at z^*, local convexity near z^* ensures that trajectories which start in Ω_δ cannot exit it since $V(z)$ is non-increasing and no other equilibrium exists inside, so they converge to z^*. In other words, within this neighborhood $A = \frac{\partial f(z)}{\partial z}$ is Hurwitz on the subspace orthogonal to the trivial invariance directions. Hence $\zeta(t)$ converges to 0.

Note that again for the case of directed tree, using the transformation in (5.4),

$$\left.\frac{\partial f(z)}{\partial z}\right|_{z=z^*} = -\mathbf{D}_\tau^T\, \mathbf{D}^{\text{in}}\, \mathbf{R}^T\, \mathcal{W}(z^*) = -L_{e_\tau}\, \mathcal{W}(z^*).$$

Let $A_\tau = \frac{\partial f(z)}{\partial z}\big|_{z=z^*} = -L_{e_\tau}\mathcal{W}(z^*)$. The local convexity condition suggests that A_τ is Hurwitz. Formally, this sufficient condition can be shown using the following quadratic Lyapunov function using the disagreement vector. In fact, since A_τ is Hurwitz, there exists a positive symmetric matrix M [43] such that $A_\tau^T\, M + M\, A_\tau = -Q$ for some choice of $Q = Q^T > 0$. Let the Lyapunov function

$$V = \xi^T\, M\, \xi, \tag{5.36}$$

such that $\lambda_{\min}(M)\|\xi\|^2 \leq V \leq \lambda_{\max}(M)\|\xi\|^2$. The derivative along ξ becomes

$$\dot{V}(\xi) = \dot{\xi}^T\, M\xi + \xi^T\, M\dot{\xi} \tag{5.37}$$

By picking a sufficiently small ball $B_\delta(z^*)$, the derivative along ξ becomes

$$\begin{aligned}\dot{V}(\xi) &= f(z^*+\xi)^T\ M\ \xi + \xi^T\ M\ f(z^*+\xi) \\ &= \xi^T\ (A_\tau^T\ M + M\ A_\tau)\ \xi + \underbrace{\epsilon(\xi)^T\ M\ \xi + \xi^T\ M\ \epsilon(\xi)}_{\textit{small if } \|\xi\| \textit{ is small}} \\ &\leq -\xi^T\ Q\ \xi,\end{aligned} \tag{5.38}$$

Which is negative definite when $\xi \neq 0$. So, the origin $\xi = 0$ (or $z = z^*$) is a locally asymptotically stable equilibrium under the chosen Lyapunov candidate and the trajectories converge to the desired behavior if they start sufficiently close. In fact, letting $\gamma = \frac{\lambda_{\min(Q)}}{\lambda_{\max}(M)} > 0$, then

$$\dot{V}(\xi) \leq -\gamma\|\xi\|^2 \leq -\gamma V.$$

□

A particular example of potential functions that ensures this local convexity condition is the class of Morse functions, whose structure guarantees that each equilibrium is an isolated minimum with a non-singular Hessian, leading to local exponential convergence as described in Theorem 1.

5.5.3 Robustness to Bounded Disturbances

Real-world systems often experience external perturbations or unmodeled dynamics. Let $\eta(z,t)$ be an additive, bounded disturbance such that $\|\eta(z,t)\| \leq \bar{\eta}$ for some finite $\bar{\eta} > 0$. The edge dynamics become

$$\dot{z} = -L_e\ \mathbf{W}(z)\ z\ +\ \eta(z,t).$$

Repeating the Lyapunov analysis with $V(z) = \mathcal{P}(z) - \beta$,

$$\dot{V}(z) = -y^T \; L_e \; y \; + \; y^T \; \eta$$

In the general case of assumption 5.1, we only have a weak Lyapunov function. Again, by restricting attention to the same local neighborhood Ω_δ around z^* (where the potential is strictly convex), and choosing δ such that $\|z - z^*\| \leq \delta$, we define $V(\xi) = \xi^T M \xi$, with $M \succ 0$, we can linearize the dynamics around z^* to obtain $\dot{\xi} = A\xi + d(\xi, t)$, where A is Hurwitz in Ω_δ and $d(\xi, t)$ accounts for higher-order terms and the bounded disturbance $\eta(z, t)$ (with $\|\eta(z, t)\| \leq \bar{\eta}$). Then, the derivative of V along trajectories satisfies

$$\dot{V}(\xi) \leq -\alpha(\|\xi\|) + \gamma(\|\eta\|),$$

for some class-$\mathcal{K}$ functions α and γ. This local input-to-state stability (LISS) result shows that, within a neighborhood of z^*, the error ξ converges exponentially to a neighborhood whose size depends on $\bar{\eta}$.

For directed tree case, we obtain:

$$\dot{V} \leq -y_\tau^T \; L_{e_\tau} \; y_\tau + y_\tau^T \eta.$$

Applying Young's inequality we have

$$y_\tau^T \;\; \eta \leq \frac{\epsilon}{2} \|y_\tau\|^2 + \frac{1}{2\epsilon} \;\; \|\eta\|^2 \quad \forall \epsilon > 0,$$

So,

$$\dot{V} \leq -y_\tau^T \, L_{e_\tau} \, y_\tau + \frac{\epsilon}{2}\|y_\tau\|^2 + \frac{1}{2\epsilon}\|\eta\|^2,$$

let $c_1 = \lambda_{\min}(L_{e_\tau})$, and choose $0 < \epsilon < 2\, c_1$ then

$$-y_\tau^T \, L_{e_\tau} \, y_\tau + \frac{\epsilon}{2}\|y_\tau\|^2 = -\left(c_1 - \frac{\epsilon}{2}\right)\|y_\tau\|^2 = -c_2\|y_\tau\|^2,$$

where $c_2 := c_1 - \frac{\epsilon}{2} > 0$. So, we obtain the bound on $\dot{V}$:

$$\dot{V} \leq -c_2\|y_\tau\|^2 + \frac{1}{2\,\epsilon}\|\eta\|^2 \leq -c_2\|y_\tau\|^2 + \frac{\bar{\eta}^2}{2\,\epsilon}$$

Thus, the system is input-to-state stable (ISS) [85] with respect to bounded perturbations. Refer to 8.1, and 8.3 in supplementary material for definitions of ISS.

Under the local convexity condition above, there exists a constant $\delta > 0$ such that $\|z - z^*\| \leq \delta$, with $\|\eta\| < \epsilon$. So, we can show that the system is robust to such perturbations as

$$\dot{V}(z) \leq -\gamma \, V(z) + \mathcal{X}(\|\eta\|)$$

holds for some $\gamma > 0$ and class-$\mathcal{K}$ function $\mathcal{X}$, with $V = \xi^T \, M \, \xi$. This inequality implies that if the disturbances are sufficiently small, the state remains ultimately bounded in a neighborhood of the desired configuration.

5.6 Summary

In this chapter, we presented a framework for analyzing the stability and robustness of MAS governed by nonlinear, state-dependent and potentially signed edge weights. Using energy-based Lyapunov analysis in the edge space, we showed how to ensure convergence in directed networks with mixed positive-negative interactions. Specifically, we showed under the assumptions that the digraph is a directed tree, the network achieves local exponential convergence to the desired configurations if the edge potential is locally convex around its critical points. We illustrated how defining energy functions on each edge (e.g., Morse functions) yields weight functions that can be both attractive and repulsive, which can inform design of interaction rules in MAS. Additionally, we established that if bounded disturbances are present, the system remains ISS, ensuring that the trajectories remain within a neighborhood of z^*. These theoretical insights provide a rigorous foundation for the development of data-driven methods to learn interaction rules from observations, which will be explored in the next chapter.

Chapter 6: Optimality Conditions for the Interaction Weights

6.1 Overview

In the previous chapter, we established a Lyapunov-based stability and robustness framework for MAS whose interactions depend on state-dependent and potentially signed interaction weights. Although theseese results ensure convergence and robustness, it remains challenging to design or choose energy functions or edge weights that ensure optimality with respect to desired performance criterias. In this chapter, we address this challenge by adopting an optimal control (OC) viewpoint to learn these interaction weights from observations. In particular, we formulate an OC problem in which we recover the state-dependent edge weights, characterized by the function $u_k(s)$ (as defined in (5.12)), that best explain the observed MAS behavior, assuming that the graph topology satisfies Assumption 5.1.

6.2 Introduction

In MAS, modeling how agents interact is crucial for tasks such as distributed control, formation coordination, and cooperative learning. Often, these interactions are captured by weights on a graph—where each edge weight reflects how strongly an agent influences or is influenced by its neighbors. Classical consensus and formation control methods typically assume

fixed or symmetrically positive interaction weights often optimized via convex formulations to maximize spectral metrics like algebraic connectivity [33, 36]—to ensure convergence and robustness. However, as discussed in Chapter 2, such approaches are limited when facing real-world complexities. In practice, MAS encounter dynamic, state-dependent, and even antagonistic interactions that static, nonnegative weights cannot capture.

Alternative strategies, such as potential-based methods inspired by physical forces or biologically plausible heuristics [5], provide a more intuitive design of state-dependent interactions. Yet, these are often handcrafted and lack the flexibility required for complex environments. Recent learning-based approaches—including self-attention encoders [41] and Graph Neural Network (GNN) models [19, 20, 42]—offer promising routes to recover interaction weights from data. However, these methods generally assume time-invariant or exclusively positive weights and offer limited stability guarantees.

Building on the stability guarantees established through Lyapunov analysis (Chapter 5), this chapter develops an OC framework for learning optimal, state-dependent interaction weights from observations. In our formulation, the unknown interaction weights are modeled on a fixed graph with a predetermined number of edges, and we present rigorous derivations of necessary and sufficient optimality conditions for learning robust, state-dependent interaction weights from observations. Additionally, we propose an algorithmic solution for recovering interaction behavior from data. We also discuss the role of data richness in ensuring the unique identification of these interaction rules. In summary, this chapter integrates the robust theoretical model from Chapter 5 with a spatio-temporal OC methodology to yield interpretable and stable interaction weight functions directly from observed data.

6.3 Notation

We use the same convention as previous chapter. The hat notation ($\hat{\square}$) refers to the observed values, and an o subscript ($\square_o$) denotes nominal or reference values. An asterisk superscript ($\square^*$) indicates optimal or true values. The inequality $A \geq B$ denote that $(A - B)$ is positive semi-definite.

6.4 Problem Formulation: Spatio-Temporal Optimal Control Problem

In this chapter, we consider the following class of Bolza OC problems [86]:

$$\begin{aligned}
\min_{u(s)} J(x,u) = \min_{u(s)} \int_{t_0}^{t_f} \int_{\delta}^{\Delta} & \underbrace{Q\left(x(\tau), \hat{x}(\tau)\right)}_{\text{Temporal term}} + \underbrace{G\left(u(s)\right)}_{\text{Spatial term}} ds d\tau \\
& + \psi\left(x(t_f), \hat{x}(t_f)\right) \\
\text{s.t.} \quad & \dot{x} = f(x(t), u(s)), \\
& \text{Given } x(t_0) = \hat{x}(t_0)
\end{aligned} \tag{6.1}$$

where $x(t) = [x_1^T(t), \ldots, x_N^T(t)]^T \in \mathbb{R}^{dN}$ denote the true state of N agents. The (unknown) control variable (edge policy), $u(s) = [u_1(s), \ldots, u_m(s)]$ is a function of inter-agent interaction, where u_k is the edge policy associated with edge k and is a non-parametric, scalar-valued function for any $0 \leq \delta \leq s \leq \Delta < \infty$. We assume that $u_k(s) \in \mathcal{L}^1([\delta, \Delta])$. The integrand $Q\left(x(\tau), \hat{x}(\tau)\right)$ is a temporal cost penalizing the deviations between the estimated ($\hat{x}(t)$) and true trajectories over time, and $G\left(u(s)\right)$ is a spatial regularization term on the (unknown) control policy. Through this regularization, we can bias the learned policy to follow the stability analysis we established in Chapter 5, so the resulting learned potential inherits Morse-like properties, namely having

well-defined minima and the possibility of modeling both attraction and repulsion. The terminal cost, $\psi(x(t_f),\hat{x}(t_f))$, penalizes for deviations between the estimated and true trajectories at the final time, t_f. The dynamics of the system are given as the same weighted consensus as defined in chapter 5:

$$f(x,u(s)) = -\Big(L^{\text{in}}(w(\alpha(x),u(s))) \otimes I_d\Big)x, \tag{6.2}$$

with

$$[w(\alpha(x(t)),u(s))]_k = \int_{\delta}^{\alpha_k(x(t))} u_k(s)\,ds = \int_{\delta}^{\Delta} u_k(s)\mathbf{1}_{\{s<\alpha_k(x(t))\}}\,ds, \tag{6.3}$$

for each edge k and $[\alpha(x(t))]_k = \alpha_k(x(t)) = \|x_i(t) - x_j(t)\|$ represents the Euclidean distance between agents (nodes) i and j at edge $k = (j,i) \in \mathcal{E}$. The diagonal weight matrix is given by $\mathbb{W}(\alpha(x),u(s)) = \text{diag}(w(\alpha(x),u(s))) \in \mathbb{R}^{m\times m}$.

The problem of OC is to solve for the optimal policy $u^*(s) = [u_1^*(s),\ldots,u_m^*(s)]$ that minimizes the cost $J(x,u(s))$ and recovers the optimal interaction weights,

$$w^*(\alpha(x^*),u^*(s)) = [w_1^*(\alpha_1(x^*),u_1^*(s)),\ldots,w_m^*(\alpha_m(x^*),u_m^*(s))],$$

given observed trajectories $T_{obs} = \{(x^*(t)\}$. In practice, this problem is solved through forward-backward sweeping [87]. The OC problem defined in (6.1) differs from the classical Bolza OC in that it introduces an additional integral over spatial interaction distances and we call this a spatio-temporal OC problem. As a result, we rederive the optimality conditions using tools from calculus of variation.

6.5 Optimality Condition

To solve for the optimal interaction weights, we will construct the necessary conditions for optimality using the Pontryagin's Minimum Principle [86]. Assuming that the observed trajectories, T_{obs} are optimal (i.e., solving the optimal control problem in (6.1), those conditions must be satisfied.

Theorem 6.1. *Given a directed network of MAS with dynamics defined in* (6.2)*, and*

$$\mathcal{I}\left(x(\tau), u(s)\right) = Q(x(\tau), \hat{x}(\tau)) + G(u(s)), \tag{6.4}$$

with the temporal and regularization costs given as:

$$Q(x(\tau), \hat{x}(\tau)) = \frac{1}{2}\frac{\|x(\tau) - \hat{x}(\tau)\|^2}{\Delta - \delta}, \qquad G(u(s)) = \frac{1}{2}\frac{\|u(s) - u_o(s)\|^2}{t_f - t_0}, \tag{6.5}$$

for nominal policy $u_o(s) = [u_{o_1}, \ldots, u_{o_m}]$*, with* $u_{o_k} \in \mathcal{L}^1([\delta, \Delta])$ *for each edge, the optimal control,* $u^*(s)$ *solving problem defined in* (6.1) *satisfies the following optimality condition:*

$$u^*(s) = u_o(s) - \frac{1}{t_f - t_0}\int_{t_0}^{t_f} \sum_{i=1}^{N} \sum_{(j,i)\in\mathcal{E}} \lambda_i^{*^T}(x_i^*(\tau) - x_j^*(\tau))\mathbf{1}_{s<\|x_i^* - x_j^*\|}\, d\tau, \tag{6.6}$$

which depends on the graph topology, as dictated by the set of in-neighbors, the co-state, the edge state, z, and its norm. The learned control $u^*(s)$ *minimizes the cost, resulting in the optimal*

interaction weights using (5.12)*:*

$$w^*(\|x_i^* - x_j^*\|, u^*) = \int_{\delta}^{\|x_i^* - x_j^*\|} u^*(s)\, ds, \quad \forall (j,i) \in \mathcal{N}_i^{\text{in}}. \tag{6.7}$$

Proof. Follows from calculus of variation, refer to appendix for more details. The Hamiltonian is given by

$$H = \int_{\delta}^{\Delta} \mathcal{I}(x, u(s)) + \lambda^T \tilde{f}(x, u(s))\, ds,$$

where $\tilde{f}(x, u(s))$ is given from (8.5), and the co-state trajectory $\lambda \in \mathbb{R}^{Nd}$ satisfies:

$$\dot{\lambda}^T = (x - \hat{x})^T + \lambda^T \mathcal{D}^{\text{in}}\, \mathcal{W} \mathcal{D}^T, \tag{6.8}$$

where

$$\mathcal{W}(z) = \text{blkdiag}\left(w_1(\|z_1\|) I_d + \frac{dw_1(\|z_1\|)}{ds} \frac{z_1 z_1^T}{\|z_1\|}, \ldots, w_m(\|z_m\|) I_d + \frac{dw_m(\|z_m\|)}{ds} \frac{z_m z_m^T}{\|z_m\|} \right),$$

and the boundary condition is given by:

$$\lambda(t_f) = -\left(\frac{\partial \psi}{\partial x} \right)^T \Bigg|_{x = x(t_f)}.$$

□

Corollary 6.1. *The OC problem defined in* (6.1) *satisfies the second-order sufficient condition for*

optimality [88]. In particular, the Hamiltonian

$$H(x,u,\lambda) = \int_{\delta}^{\Delta} \Big[\mathcal{I}(x,u(s)) + \lambda^T \tilde{f}(x,u(s)) \Big] ds$$

is convex in the control variable $u(s)$ in a neighborhood of the optimal solution $(x^*, u^*(s), \lambda^*)$. Consequently, the Hessian of H with respect to $u(s)$ is strictly positive definite, ensuring that $u^*(s)$ is a strict local minimizer of the cost functional.

Proof. We can see that the Hessian of the Hamiltonian at the optimum is:

$$\frac{\partial^2 H}{\partial u^2}(x^*, u^*, \lambda^*) = \int_{\delta}^{\Delta} \frac{\partial^2 I(x^*, u^*)}{\partial u^2} + \lambda^T \frac{\partial^2 \tilde{f}(x^*, u^*)}{\partial u^2} ds.$$

Since $\mathcal{I}(x,u(s))$ is quadratic in $u(s)$ (due to the regularization term $G(u(s)) = \frac{1}{2}\|u(s) - u_o(s)\|^2$), the second derivative $\frac{\partial^2 \mathcal{I}(x,u(s))}{\partial u(s)^2}$ is positive definite. This ensures that the contribution from $\mathcal{I}(x,u(s))$ to the Hessian is strictly positive.

For the dynamics $\tilde{f}(x,u(s))$, the term $\frac{\partial^2 \tilde{f}(x,u(s))}{\partial u(s)^2}$ is zero because $\tilde{f}(x,u(s))$ is linear in $u(s)$. Therefore, the Hessian simplifies to:

$$\frac{\partial^2 H}{\partial u^2}(x^*, u^*(s), \lambda^*) = \int_{\delta}^{\Delta} \frac{\partial^2 I(x^*, u^*(s))}{\partial u^2(s)} ds.$$

Since $\frac{\partial^2 I(x^*, u^*(s))}{\partial u^2(s)} > 0$ for all $s \in [\delta, \Delta]$, the Hessian is strictly positive definite. Therefore, the second-order sufficient condition for optimality is satisfied, ensuring that the solution $u^*(s)$ corresponds to a strict minimum of the cost functional. Thus, the control problem defined by (8.23) satisfies the second-order sufficient condition for optimality. □

Remark 6.1. No assumptions are made on the nominal policy, $u_o(s)$, in our approach.

6.5.1 Learning Interaction Weight Algorithm

Based on Theorem 6.1 and the stability analysis in Chapter 5, we learn the optimal interaction weights by iteratively updating the control policy $u(s)$ using forward-backward sweeping [87] to satisfy both the first-order (necessary) and second-order (sufficient) optimality conditions. Algorithm 1 outlines our iterative procedure using gradient descent with an Armijo line search. In each iteration, the algorithm performs a discrete-time forward simulation, a backward integration of the co-state, and then computes the gradient of the cost with respect to $u(s)$. The algorithm terminates when the norm of the gradient is below a specified tolerance ϵ.

Note that in practice, as seen in algorithm 1, we make the assumption that observations may arise from a system which is only approximately optimal. So, the terminating conditions are based on how closely (as denoted by the, ϵ, threshold) the optimal control policy, $u(s)$ satisfies the necessary condition for optimality [90]. Since the algorithm need to solve for the forward optimal control problem repeatedly in an inner loop that may lead to a computational bottleneck.

6.6 Richness of data

So far, we leveraged PMP to derive optimality conditions for the unknown policy $u(s)$, which maps inter-agent distances to interaction strengths with an assumption that we have access to rich enough data to recover interaction behavior. However, a critical aspect of any learning or system identification procedure is the quality of training data. In other words, the observed data must be sufficiently rich, or what is known in adaptive control as persistently exciting (as

Algorithm 1 Interaction Weights Learning

1: **Input:** Initial state x_0, initial control policy $u(s) = u_0$, time discretization: $k = 0, 1, \ldots, K$ with time step Δt, spatial discretization: $s \in \{\delta = s_0 < s_1 < \cdots < s_m = \Delta\}$ with step Δs, observed trajectories $\hat{x}$, network topology $\mathcal{E}$, maximum iterations $N_{\max}$, and tolerance ϵ.
2: **Output:** Optimal interaction weights $w^*(\alpha(x^*), u^*(s))$, optimal control policy $u^*(s)$, and reconstructed state trajectories $\{x[k]\}_{k=0}^{K}$.
3: Set the iteration counter: iter $\leftarrow 0$
4: **while** iter $< N_{\max}$ **do**
5: **Forward Pass:**
6: **for** $k = 0$ to $K - 1$ **do**
7: Compute the current edge weights via discrete integration:

$$w[k] = \sum_{s=\delta}^{\Delta} u(s) \mathbf{1}_{\{s < \|x_i[k] - x_j[k]\|\}} \Delta s.$$

8: Update the state: $x[k+1] = x[k] + \Delta t \left(-\mathbf{D}^{\text{in}} \mathbf{W}[k] \mathbf{D}^T x[k] \right)$.
9: **end for**
10: **Backward Pass:**
11: Set terminal co-state:

$$\lambda[K] = -\left(\frac{\partial \psi}{\partial x} \right)^T \Big|_{x = x[K]}.$$

12: **for** $k = K - 1$ downto 0 **do**
13: Update co-state using backward Euler:

$$\lambda[k] = \lambda[k+1] + \Delta t \left(\frac{\partial \mathcal{I}(x[k], u(s))}{\partial x} - \lambda[k+1]^T \frac{\partial \tilde{f}(x[k], u(s))}{\partial x} \right)^T.$$

14: **end for**
15: Compute the gradient of the cost with respect to $u(s)$

$$\tilde{J}_u(s) = u(s) - u_o(s) + \frac{1}{K} \sum_{k=0}^{K-1} \sum_{i=1}^{N} \sum_{(j,i) \in \mathcal{N}_i^{\text{in}}} \lambda_i[k]^T \left(x_i[k] - x_j[k] \right) \mathbf{1}_{\{s < \|x_i[k] - x_j[k]\|\}}.$$

16: **if** $\left\| \tilde{J}_u(s) \right\| < \epsilon$ **then**
17: Terminate the algorithm.
18: **else**
19: Use the **Armijo Line Search** [89] to determine
20: step size, ℓ
21: Update control policy, u. using gradient descent:
22: $u(s) = u(s) - \ell \tilde{J}_u$
23: **end if**
24: Increment the iteration counter: iter $\leftarrow$ iter $+ 1$ and repeat from step 5.
25: **end while**

defined in 6.1. In our context, "data richness" refers to conditions under which the demonstration trajectories contain enough variation to uniquely identify the underlying policy, $u(s)$. In other words, the demonstration trajectories of the MAS must *adequately excite* all relevant distance regimes $\alpha \in [\delta, \Delta]$ so that no two distinct policies $u(\cdot)$ can produce the same observed behavior.

Definition 6.1. [91] A signal $\phi(t) \in \mathbb{R}^n$ is said to be persistently exciting over an interval $[t_0, t_0+T]$ if there exist constants $\alpha_0 > 0$ and $T_0 > 0$ such that

$$\int_{t_0}^{t_0+T_0} \phi(t)\ \phi(t)^T\ dt \geq \alpha_0\ I_n.$$

Assume that we have collected state measurements at T different time instances, $t_1, \ldots, t_T$. As the stability and robustness analysis in Chapter 5 was based on the edge variable $z_k = x_i - x_j$, the data need to capture a variety of z_k values to find the control policy that leads to desired configuration.

For each time t_i, we can construct the edge state vectors for each time using $z(t_i) = \mathbf{D}^T x(t_i)$. To analyze the information content over the range of inter-agent distances, we construct the data matrix, $\Phi \in \mathbb{R}^{T \times m}$ whose rows correspond to time snapshots and whose columns corresponds to the edges. Define the feature

$$\phi_{i,k} = \|z_k(t_i)\| \quad for \quad 1 \leq i \leq T, 1 \leq k \leq m.$$

If the data comes from q independent trials, let $\Phi^{(l)} \in \mathbb{R}^{T \times n_s}$ denote the feature matrix

constructed from the lth trial. We then form the overall feature matrix as:

$$\Phi_{\text{total}} = \begin{bmatrix} \Phi^{(1)} \\ \Phi^{(2)} \\ \vdots \\ \Phi^{(M)} \end{bmatrix} \in \mathbb{R}^{(qT)\times m}.$$

The data is said to be sufficiently rich if Φ_{total} has full column rank (i.e., its columns are linearly independent, indicating diverse distance measurements). To quantify this, we perform a singular value decomposition (SVD): $\Phi_{\text{total}} = U\Sigma V^\top$, where $\Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_{\min(m,\, qT)})$ with singular values ordered as $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(m,\, qT)}$, then the data is persistently exciting if $\sigma_{\min} = \min_{1\leq j\leq \min(m,\, qT)} \sigma_j > \epsilon$, for some small threshold $\epsilon > 0$. Moreover, a moderate condition number

$$\kappa(\Phi_{\text{total}}) = \frac{\sigma_1}{\sigma_{\min}}$$

indicates that the mapping from $u(s)$ to the observed data is well-conditioned. This ensures that the inverse problem of recovering $u(s)$ from the demonstration trajectories is well-posed.

Remark 6.2. If the PE condition is not satisfied in the range of $[\delta, \Delta]$, then the corresponding components of $u^*(s)$ are not observable from the data. In such cases, the regularization term $G(u(s))$ in the cost function forces the learned policy to converge to the nominal policy $u_o(s)$ outside the subspace spanned by the excited data.

6.7 Simulation Example

In this section, we present a simulation example to examine the effectiveness of our IOC algorithm. We first generate a sufficiently rich dataset of trajectories that will serve as input (observation) for the algorithm.

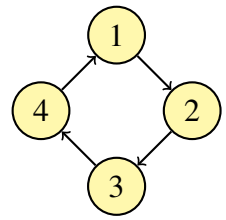


Figure 6.1: Directed graph with 4 agents. The arrows indicate the direction of interaction.

Table 6.1: Edge parameters for the simulation.

Edge	**Parameters**
Edge 1 (1→2)	$k_1 = 0.5, \quad \alpha_1^* = 2$
Edge 2 (2→3)	$k_2 = 1, \quad \alpha_2^* = 0.8$
Edge 3 (3→4)	$k_3 = 1.2, \quad \alpha_3^* = 1$
Edge 4 (4→1)	$k_4 = 1.8, \quad \alpha_4^* = 0.5$

6.7.1 Simulation Setup

For the observation, we consider a MAS defined on a digraph with four nodes and four directed edges (Figure 6.1). For each edge, the true potential function is defined as a function of inter-agent distances, s:

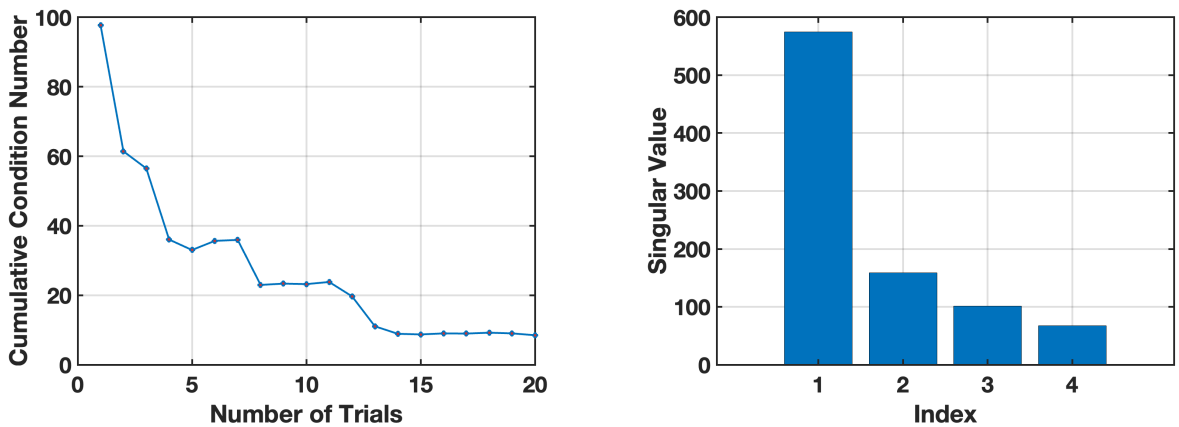
$$P_k(s) = \frac{1}{2} k_k \, (s - \alpha_k^*)^2 \tag{6.9}$$

where k_k is the stiffness parameter for edge k, z_k^* is the desired distance for edge agents k, and z is the current inter-agent distance. So, the corresponding edge weight, w_k, and control policy, u_k are given by

$$w_k(s) = \frac{k_k \, (s - \alpha_k^*)}{s + \varepsilon}, \quad u_k(s) = \frac{k_k \, \alpha_k^*}{s^2 + \epsilon},$$

with $\epsilon = 1e^{-6}$. The parameters for each edge is summarize in Table 6.1.

The simulation uses a forward Euler with a time step $dt = 0.01$ over a total simulation time of $T_f = 20$ seconds. To generate demonstrations: the simulation runs 20 trials. In each trial, the



(a) Cumulative condition number.

(b) Singular values of Φ_{total}.

Figure 6.2: Evaluation of richness of data. (a) Cumulative condition number of the aggregated feature matrix as a function of the number of trials. (b) Bar plot of singular values of the aggregated feature matrix, Φ_{total}.

agents are initialized with random positions drawn from different (e.g., small, medium, large) domains to ensure a wide range of inter-agent distances. A trial is accepted only if the final inter-agent distances for all edges are within a tolerance of 0.01 of their respective desired distances, i.e., $\|\alpha_k(T_f) - \alpha_k^*\| < 0.01, \quad \forall k$. After collecting the trajectories from all trials, a feature matrix Φ_{total} is constructed from inter-agent distances across time and trials. Figure 6.2a shows how the condition number decreases with additional trials with more data diversity. A bar plot of singular values of the aggregated feature matrix, Φ_{total} is shown in (Figure 6.2b). Note that the largest and smallest singular values differ by about an order of magnitude, indicating an acceptable condition number. This suggests that the collected data is sufficiently rich to enable reliable recovery of the interaction weights.

Once the demonstrations are generated, we follow algorithm (1) to recover the edge policy, u_k from the observation. Motivated by the local convexity conditions on potential energy from Chapter 5, we use $u_o = \frac{1}{s^2+\epsilon}$, as our nominal policy.

6.7.2 Simulation Results

The simulation results confirm the effectiveness of our spatio-temporal OC framework across multiple performance metrics. As shown in Figure 6.3 the overal cost converges after within approximately 25 iterations. Figures 6.4 compares the recovered (learned) edge policies with the true ones, revealing that edges 2 and 3 have slightly higher root mean square error (RMSE) values, primarily due to the larger peaks in their policies when distance between agents are very small. We also computed the potential function from edge policy using equation (5.15) and compared it with true potential policy given in (6.9). As shown in Figure 6.5, the learned potential follows the true potential function Morse-like property. To illustrate the impact of varying initial conditions, Figure 6.6 presents the evolution of edge weights and corresponding agent trajectories for three representative test trials side by side. Specifically, the first trial (Figures 6.6a and 6.6b) exhibits mixed positive and negative edge weights, the second trial (Figures 6.6c and 6.6d) features all positive edge weights, and the third trial (Figures 6.6e and 6.6f) shows all negative edge weights. In each case, the learned edge weights closely track the true weights (with RMSE ≤ 0.06) and the agent trajectories align well with the observation (RMSE ≤ 0.12). Overall, these results confirm that our OC framework reliably recovers the desired interaction rules and maintains consistent performance despite variations in initial conditions.

6.8 Summary

In this chapter, we developed a novel spatio-temporal OC framework for recovering optimal, state-dependent interaction weights in MAS, grounded in the stability guarantees established in Chapter 5. By formulating a Bolza OC problem and deriving necessary optimality conditions via

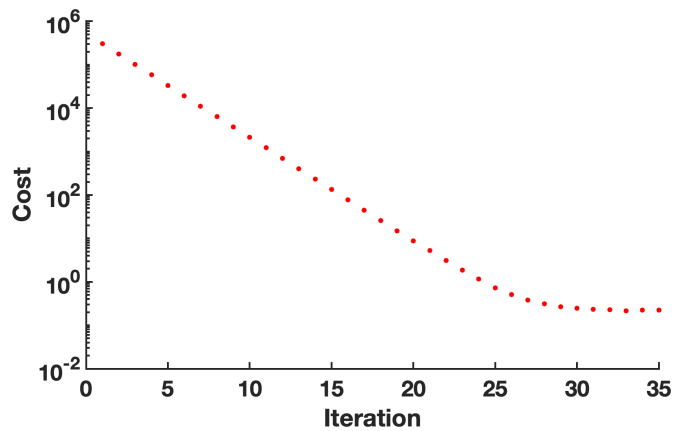


Figure 6.3: Cost evolution during the OC learning process.

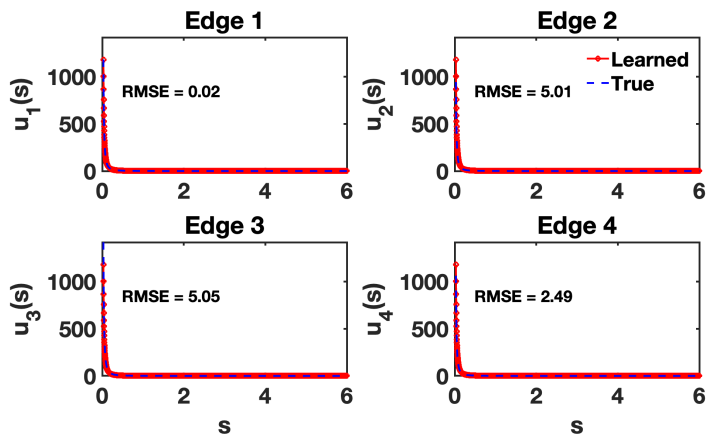


Figure 6.4: Comparison of learned (red markers) and true (blue dashed) edge policies $u(s)$ for each edge.

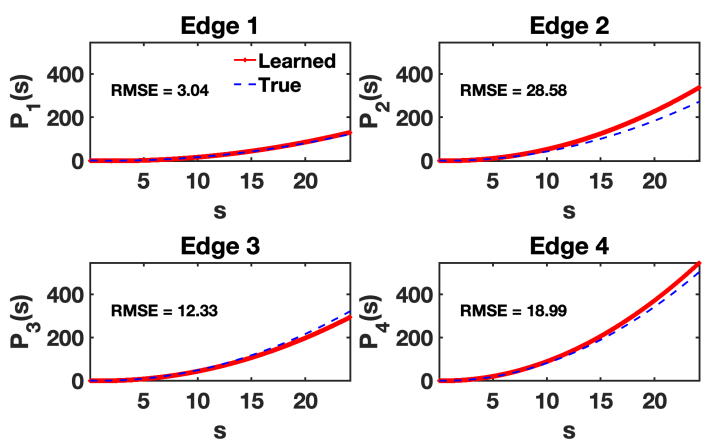
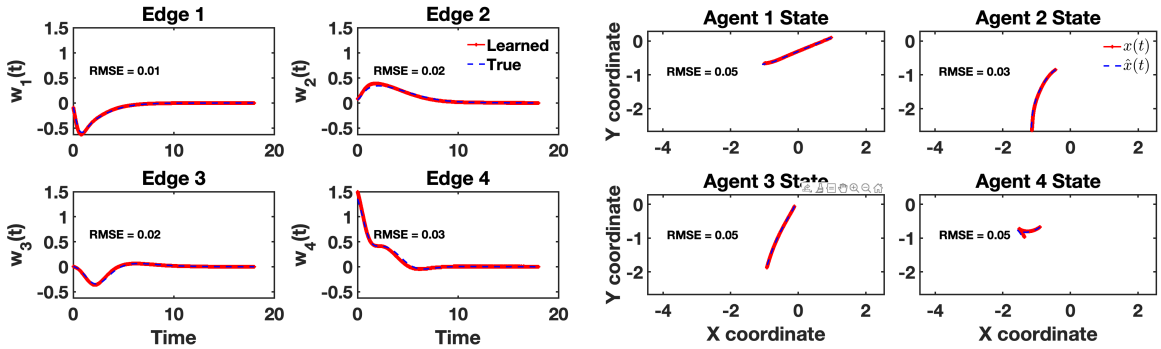
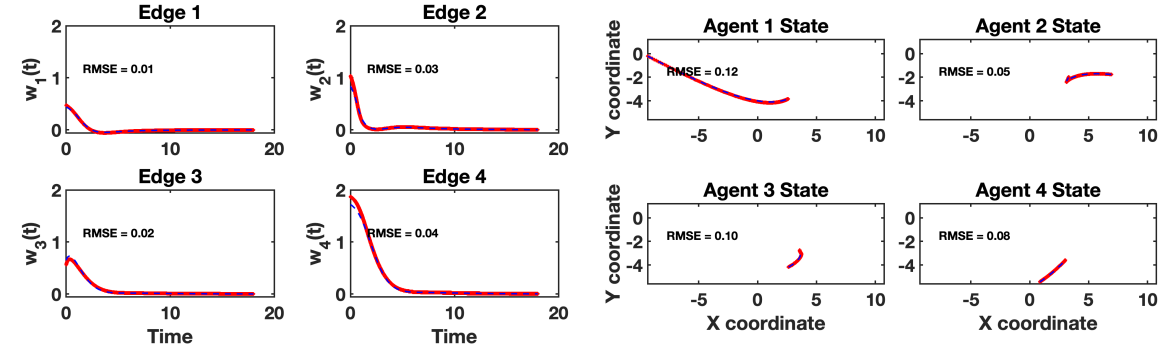


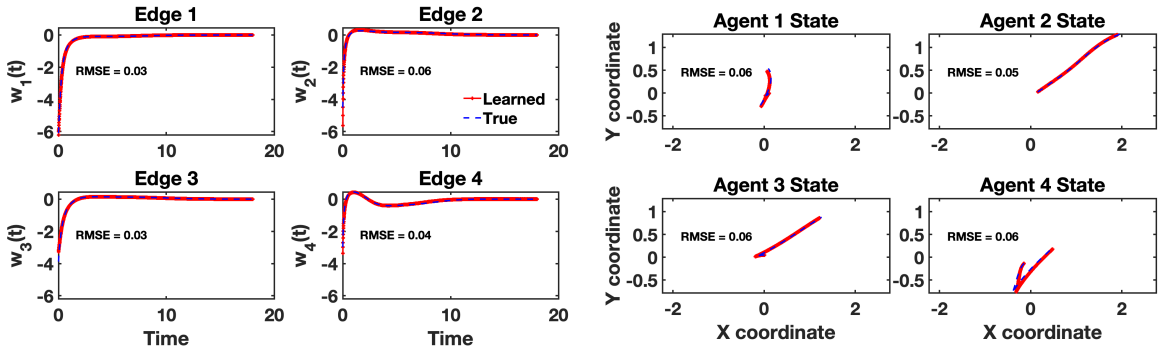
Figure 6.5: Comparison of learned (red markers) and true (blue dashed) potential policies $P(s)$ for each edge.



(a) Edge weights (b) Agent Trajectories



(c) Edge weights (d) Agent Trajectories



(e) Edge weights (f) Agent Trajectories

Figure 6.6: Representative trials: evolution of edge weights and corresponding agent trajectories. Top row (a,b): Mixed positive and negative edge weights; Middle row (c, d): All positive edge weights; Bottom row (e, f): All negative edge weights.

PMP, we established that, given persistently exciting data, the inverse problem of recovering the edge policy $u(s)$ is well-posed. The numerical experiments demonstrate how the learned policy remains near the chosen nominal Morse-like function yet still accurately reconstructs observation. This integration of theoretical stability guarantees with data-driven methods offers a reliable and interpretable path toward designing interaction rules for MAS.

Chapter 7: Conclusion

This dissertation presented a framework for modeling and learning interaction behaviors in distributed intelligent systems. Motivated by the adaptability and robustness observed in human coordination, the proposed approach integrates data-driven modeling with control-theoretic guarantees to learn interaction structures that are stable, robust, and interpretable. A key contribution was the integration of consensus theory with the principles of human motor synergy. By linking the consensus theory with the UCM hypothesis from human motor control, we showed that consensus-based models can capture the implicit coordination mechanisms observed in human tasks, providing a biologically inspired foundation for robust multi-agent interactions.

Furthermore, we extended the edge-based analysis to more general multi-agent interaction models by generalizing the concept of directed potential energy. While the edge-based framework was originally developed in prior work [40, 49, 92], we built upon it to define edge tension energies that accommodate directed graphs with both positive and negative edge weights, capturing complex interactions commonly observed in human coordination. This formulation, grounded in inter-agent distances, is particularly suited for systems with range-based sensors that provide only relative position or velocity measurements. By shifting from a node-based to an edge-based perspective, we relaxed the assumption of homogeneous agent dynamics and captured pairwise interactions under nonlinear interconnections and signed or directed edge weights. Using

Lyapunov methods, we derived sufficient conditions for local stability and robustness under these generalized settings. We also proposed a spatio-temporal OC formulation to recover interaction weights directly from observations and derived necessary and sufficient conditions for optimality. Overall, we showed that human interaction behaviors that emerge in physical coordination tasks can be robustly modeled and learned via nonlinear agreement dynamics.

Despite these contributions, several opportunities for future research remain. Although the optimality condition offers a compelling geometric edge-based interpretation, solving OC problem in practice is challenging. In this work, we employed gradient-based optimization methods for demonstration purposes. However, such methods are prone to local minima and may not scale well to more complex systems. Future efforts could explore more scalable and efficient solvers—such as collocation or test shooting methods, as discussed in [93]—to improve convergence properties, especially in large-scale systems. Additionally, our formulation assumes fully persistently exciting data across the edge state space; However, as the numerical example suggests, perfect condition number may not be required to recover potential function. Future work could explore relaxed richness conditions that still guarantee reliable learning without requiring exhaustive trajectory data.

Finally, although our model has been validated through simulations, real-world testings, such as on crowd navigation datasets, remains to be explored. In those complex scenarios, while true interaction policies are not directly observable, predictive performance can still be evaluated using metrics like average and final displacement errors, as demonstrated in Chapter 4. Overall, this work contributes to a deeper understanding of interaction learning in MAS and lays a principled foundation for developing distributed intelligent systems that are more robust, human-like, and capable of handling complex, nonlinear, and heterogeneous interactions.

Chapter 8: Supplementary Material

In this section, we provide some useful lemmas and also detailed proofs of some of the theorems in Chapter 5 and 6.

APPENDIX A

Lemma 8.1. *[94] The Hadamard product (i.e., element-wise multiplication) of two compatible vectors $a \in \mathbb{R}^d$ and $b \in \mathbb{R}^d$ can be expressed as the matrix multiplication of the corresponding diagonal matrix of one vector by the other vector:*

$$a \odot b = diag(a)b = diag(b)a, \tag{8.1}$$

$$diag(a)diag(b) = diag(a \odot b), \tag{8.2}$$

Lemma 8.2. *The diagonal term in (5.14) can be interchanged as:*

$$\mathbf{D}^{\text{in}} \left(diag\big(\phi(s)\big) \otimes I_d \right) \mathbf{D}^T x(t) \; = \mathbf{D}^{\text{in}} \; diag \; \big(\mathbf{D}^T x(t)\big) \; \left(\phi(s) \otimes \mathbf{1}_d \right) \tag{8.3}$$

Proof. Follows from the Hadamard properties in Lemma 8.1. It can be easily seen that

$$\operatorname{diag}\big(\phi(s)\big) \otimes I_d = \operatorname{diag}\big(\phi(s) \otimes \mathbf{1}_d\big).$$

So, the integrand can be rearranged, which is useful for the subsequent Jacobian derivation.

$$\begin{aligned} \mathbf{D}^{\text{in}} \left(\operatorname{diag}\big(\phi(s)\big) \otimes I_d\right) \mathbf{D}^T x(t) &= \mathbf{D}^{\text{in}} \operatorname{diag}\big(\phi(s) \otimes \mathbf{1}_d\big)\, \mathbf{D}^T x(t) \\ &= \mathbf{D}^{\text{in}} \operatorname{diag}(\mathbf{D}^T x(t)) \left(\phi(s) \otimes \mathbf{1}_d\right). \end{aligned} \tag{8.4}$$

□

Corollary 8.1. *For the special case of $d = 1$, the interaction dynamics simplify to:*

$$\mathbf{D}^{\text{in}}\, diag\big(\phi(s)\big)\, \mathbf{D}^T x(t) \;= \mathbf{D}^{\text{in}}\, diag\, \big(\mathbf{D}^T x(t)\big)\, \phi(s)$$

According to Lemma 8.2, the integrand in (5.14) becomes:

$$\tilde{f}(x(t), u(s)) = -\mathbf{D}^{\text{in}} \operatorname{diag}(\mathbf{D}^T x(t)) \left(\phi(s) \otimes \mathbf{1}_d\right) \tag{8.5}$$

Proof of lemma 5.2

Proof. Let the integrand in equation (5.14) be:

$$\tilde{f}(x(t), u(s)) = \mathbf{D}^{\text{in}} \left(\operatorname{diag}\big(\phi(s)\big) \otimes I_d\right) \mathbf{D}^T x(t).$$

As before, define the function $\phi(x,s)$ component-wise as

$$\phi_k(x,s) = u_k(s)\,\mathbf{1}_{\{s<\alpha_k(x)\}},$$

where its derivative with respect to x_j is given by

$$\frac{\partial}{\partial x_j}\big(\phi_k(x)\big) = \frac{\partial}{\partial x_j}\Big(u_k(s)\,\mathbf{1}_{\{s<\alpha_k(x)\}}\Big) = u_k(s)\,\delta\big(\alpha_k(x)-s\big)\,\frac{\partial \alpha_k(x)}{\partial x_j}.$$

Here, the derivative of the indicator function is interpreted in the distributional sense, yielding the Dirac delta function $\delta(\alpha_k(x)-s)$ (with the usual properties $\int_{-\infty}^{\infty}\delta(S)\,ds = 1$ and $\int_{-\infty}^{\infty} g(S)\delta(S)\,ds = g(0)$). In matrix form, this yields

$$\frac{\partial \phi(x,s)}{\partial x} = \Lambda(x,s)\,\frac{\partial \alpha(x)}{\partial x},$$

with

$$\Lambda(x,s) := \operatorname{diag}\Big(u(s)\odot\delta\big(\alpha(x)-s\big)\Big),$$

with the kth diagonal entry given by $[\Lambda(x,s)]_{kk} = u_k(s)\,\delta\big(\alpha_k(x)-s\big)$.

We now characterize the Jacobian of $\tilde{f}(x,s)$, as defined in (8.5). Using Lemma 8.2, and chain rule we have

$$\frac{\partial \tilde{f}}{\partial x} = -\mathbf{D}^{\text{in}}\Big(\operatorname{diag}\big(\phi(x,s)\big)\otimes I_d\Big)\mathbf{D}^T - \mathbf{D}^{\text{in}}\operatorname{diag}\big(\mathbf{D}^T x\big)\Big[\Big(\Lambda(x,s)\,\frac{\partial \alpha(x)}{\partial x}\Big)\otimes \mathbf{1}_d\Big]. \tag{8.6}$$

Let $z = \mathbf{D}^T x$, which maps the node space to the edge space. Then, by the chain rule,

$$[\frac{\partial \alpha}{\partial x}] = [\frac{\partial \alpha}{\partial z}][\frac{\partial z}{\partial x}].$$

It can be shown that

$$\left[\frac{\partial \alpha_i}{\partial z_j}\right]_{i,d_j-(d-1):d_j} = \begin{cases} \frac{z_i^T}{\|z_i\|}, & \text{if } i = j, \\ 0, & \text{if } i \neq j, \end{cases} \tag{8.7}$$

so that in compact form,

$$\left[\frac{\partial \alpha}{\partial z}\right] = \begin{bmatrix} \frac{z_1^T}{\|z_1\|} & 0^T & \dots & 0^T \\ 0^T & \frac{z_2^T}{\|z_2\|} & \dots & 0^T \\ \vdots & \vdots & \ddots & \vdots \\ 0^T & 0^T & \dots & \frac{z_m^T}{\|z_m\|} \end{bmatrix}_{m \times md}. \tag{8.8}$$

This shows that $\frac{\partial \alpha}{\partial z}$ is block diagonal, with each block given by a row vector of size d.

Since, $[\frac{\partial z}{\partial x}] = \mathbf{D}^T$. So, we can write $[\frac{\partial \alpha}{\partial x}]^T = \mathbf{D}[\frac{\partial \alpha}{\partial z}]^T$, which ties the operation to the graph's topology as follows

$$[\frac{\partial \alpha}{\partial x}]_{kj} = \frac{\partial \alpha_k}{\partial x_j} = \begin{cases} \frac{(x_i - x_j)^T}{\|x_i - x_j\|} & \textit{if } \exists k \in (j,i) \\ -\frac{(x_i - x_j)^T}{\|x_i - x_j\|} & \textit{if } \exists k \in (i,j) \\ 0^T & o.w \end{cases}. \tag{8.9}$$

So, equation (8.6) can be rewritten as:

$$\frac{\partial \tilde{f}}{\partial x} = -\mathbf{D}^{\text{in}}\Big(\text{diag}\big(\phi(x,s)\big) \otimes I_d\Big)\mathbf{D}^T - \mathbf{D}^{\text{in}}\,\text{diag}\big(\mathbf{D}^T x\big)\Big[\Big(\Lambda(x,s)\,\frac{\partial \alpha(x)}{\partial z}\mathbf{D}^T\Big) \otimes \mathbf{1}_d\Big]$$

Which is equivalent to

$$\frac{\partial \tilde{f}}{\partial x} = -\mathbf{D}^{\text{in}} \underbrace{\left[\underbrace{\left(\text{diag}\big(\phi(x,s)\big) \otimes I_d\right)}_{\tilde{W}_1} + \underbrace{\text{diag}\big(\mathbf{D}^T x\big) \left[\left(\Lambda(x,s) \, \frac{\partial \alpha(x)}{\partial z} \right) \otimes \mathbf{1}_d \right]}_{\tilde{W}_2} \right]}_{\tilde{W}} \mathbf{D}^T, \tag{8.10}$$

The equivalence follows from the block diagonal form of $\frac{\partial \alpha(x)}{\partial z}$ as given in (8.8), we have:

$$\left[\left(\Lambda(x,s) \, \frac{\partial \alpha(x)}{\partial z} \mathbf{D}^T \right) \otimes \mathbf{1}_d \right] = \left[\left(\Lambda(x,s) \, \frac{\partial \alpha(x)}{\partial z} \right) \otimes \mathbf{1}_d \right] \mathbf{D}^T.$$

Here, the first term inside the bracket, $\tilde{W}_1$, is a diagonal matrix, and the second term, $\tilde{W}_2$, is a block-diagonal matrix.

Given the definition of

$$z = \mathbf{D}^T x = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_m \end{bmatrix}, \quad \text{with } z_k = \begin{bmatrix} z_{k1}, z_{k2}, \ldots, z_{kd} \end{bmatrix}^T \in \mathbb{R}^d,$$

the matrix $\text{diag}(\mathbf{D}^T x)$ is given by

$$\text{diag}(\mathbf{D}^T x) = \text{diag}(z) = \begin{bmatrix} \text{diag}(z_1) & 0 & \ldots & 0 \\ 0 & \text{diag}(z_2) & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \text{diag}(z_m) \end{bmatrix}_{md \times md}. \tag{8.11}$$

Moreover, consider the term

$$\left(\Lambda(x,s)\,\frac{\partial\alpha(x)}{\partial z}\right)\otimes\mathbf{1}_d =$$

$$\begin{bmatrix} \delta(\alpha_1(x)-s)\,u_1(s)\,\mathbf{1}_d\,\frac{z_1^T}{\|z_1\|} & 0^T & \dots & 0^T \\ 0^T & \delta(\alpha_2(x)-s)\,u_2(s)\,\mathbf{1}_d\,\frac{z_2^T}{\|z_2\|} & \dots & 0^T \\ \vdots & \vdots & \ddots & \vdots \\ 0^T & 0^T & \dots & \delta(\alpha_m(x)-s)\,u_m(s)\,\mathbf{1}_d\,\frac{z_m^T}{\|z_m\|} \end{bmatrix}. \tag{8.12}$$

Also, note that

$$\operatorname{diag}(z_k)\,\mathbf{1}_d = z_k, \quad \text{for each } k. \tag{8.13}$$

Thus, we can define the diagonal entries of $\tilde{W}_1$ and $\tilde{W}_2$ as

$$[\tilde{W}_1]_{ii} = u_i(s)\,\mathbf{1}_{\{s<\alpha_i\}}\,I_d = \phi_i(x,s)\,I_d, \tag{8.14}$$

and

$$[\tilde{W}_2]_{ii} = u_i(s)\,\delta(\alpha_i - s)\,\frac{z_i z_i^T}{\|z_i\|}, \tag{8.15}$$

so that

$$[\tilde{W}]_{ii} = \phi_i(x,s)\,I_d + u_i(s)\,\delta(\alpha_i - s)\,\frac{z_i z_i^T}{\|z_i\|}. \tag{8.16}$$

Finally, combining equations (5.14) and (8.10), the Jacobian is given by

$$\frac{\partial f}{\partial x} = -\int_\delta^\Delta \frac{\partial \tilde{f}}{\partial x}\,ds = -\int_\delta^\Delta \mathbf{D}^{\text{in}}\,\tilde{W}\,\mathbf{D}^T\,ds, \tag{8.17}$$

Since we assumed a fixed topology, and using equations (5.12), and (8.16), we have a simplified form as

$$\int_{\delta}^{\Delta} [\tilde{W}]_{ii}\, ds = \int_{\delta}^{\Delta} \left[\phi_i(x,s)\, I_d + u_i(s)\delta(\alpha_i - s)\frac{z_i z_i^T}{\|z_i\|} \right] ds = w_i(\alpha_i) I_d + u_i(\alpha_i)\frac{z_i z_i^T}{\|z_i\|} \tag{8.18}$$

Using the definition of α_i, and the fact that, $u_i(\alpha_i) = \frac{\partial w_i(\|z_i\|}{\partial s})$, the above equation can be rewritten

$$\int_{\delta}^{\Delta} [\tilde{W}]_{ii}\, ds = w_i(\|z_i\|) I_d + \frac{dw_i(\|z_i\|)}{ds}\frac{z_i z_i^T}{\|z_i\|}. \tag{8.19}$$

With this formulation, the Jacobian interestingly recovers the graph edge weights plus some additional terms emphasizing the directional influence of the interaction, with $\frac{z_i z_i^T}{\|z_i\|}$ acting as a projection. Furthermore, given the $z_i z_i^T$ structure in each block, we have a rank-1 block diagonal matrix in the second term. Let

$$\mathcal{W}(z) = \text{blkdiag}\left(w_1(\|z_1\|) I_d + \frac{dw_1(\|z_1\|)}{ds}\frac{z_1 z_1^T}{\|z_1\|}, \ldots, w_m(\|z_m\|) I_d + \frac{dw_m(\|z_m\|)}{ds}\frac{z_m z_m^T}{\|z_m\|} \right), \tag{8.20}$$

So, the Jacobian becomes:

$$\frac{\partial f}{\partial x} = -\mathbf{D}^{\text{in}} \mathcal{W}(z) \mathbf{D}^T = -\mathcal{L} \tag{8.21}$$

Let $\mathbb{W}(z) = \text{diag}(w_1(\|z_1\|), \ldots, w_m(\|z_m\|)$ and $\mathbb{W}'(z) = \text{diag}\left(\frac{dw_1(\|z_1\|)}{ds}, \ldots, \frac{dw_m(\|z_m\|)}{ds}\right)$ and the

normalized edge direction matrix:

$$Z = \begin{bmatrix} \frac{z_1}{\sqrt{\|z_1\|}} & 0 & \dots & 0 \\ 0 & \frac{z_2}{\sqrt{\|z_2\|}} & \dots & 0 \\ \vdots & & \ddots & \\ 0 & \dots & & \frac{z_m}{\sqrt{\|z_m\|}} \end{bmatrix}$$

Then:

$$\mathcal{W} = \mathbb{W}(z) \otimes I_d + \mathbb{W}'(z) Z Z^T$$

.

For the Jacobian in edge space, using the transformation $z = \mathbf{D}^T x$, we have$\dot{z} = \mathbf{D}^T f(x) = f(z)$, so using similar expansion as above we get

$$\frac{\partial f}{\partial z} = -\mathbf{D}^T \mathbf{D}^{\text{in}} \mathcal{W}(z) \tag{8.22}$$

□

Consider the general nonlinear dynamical system of

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

with $f : \mathbb{R}^{\mathbb{N}} \times \mathbb{R}^{>} \to \mathbb{R}^{\mathbb{N}}$ continuously differentiable and $f(0,0) = 0$.

Definition 8.1. [[85]] A smooth function $V : \mathbb{R}^N \to \mathbb{R}_{\geq 0}$ is called an ISS-Lyapunov function for

the system (8) if there exists class $\mathcal{K}_\infty$ functions α_1, α_2, and $\mathcal{K}-$functions α_3 and $\mathcal{X}$, such that

$$\alpha_1(||x||) \leq V(||x||) \leq \alpha_2(||x||) \quad \textit{for any} \quad x \in \mathbb{R}^N, \quad \text{and}$$

$$\left(\frac{\partial V(x)}{\partial x}\right)^T f(x,u) \leq -\alpha_3(\|x\|)$$

for any $x \in \mathbb{R}^{\mathbb{N}}$ and any $u \in \mathbb{R}^{>}$ so that $\|x\| \leq \mathcal{X}(\|u\|)$.

Lemma 8.3 ([85])**.** *Consider the nonlinear dynamical system* (8)*, the system is ISS if it admits an ISS-Lyapunov function.*

APPENDIX B

Proof of Theorem 6.1

To derive the optimality conditions, we first construct the augmented cost as

$$\tilde{J}(x,\dot{x},u(s)) = \psi\left(x(t_f)\right) + \int_{t_0}^{t_f}\int_{\delta}^{\Delta}\left[\mathcal{I}\left(x(\tau),u(s)\right) + \lambda^T\left(\frac{\dot{x}}{\Delta-\delta} - \tilde{f}(x,u(s))\right)\right] ds d\tau \tag{8.23}$$

where $\lambda \in R^{d\times N}$ is the co-state vector enforcing the dynamic constraints.

Let $u(s) \rightarrowtail u(s) + \epsilon\mu(s)$ and consequently, due to Lipschitz dynamics, $x(t) \rightarrowtail x(t) + \epsilon\eta(t)$, and $\dot{x}(t) \rightarrowtail \dot{x}(t) + \epsilon\dot{\eta}(t)$. Using the calculus of variation, we need to set $\delta\tilde{J} \overset{!}{=} 0$ to find the optimal policy and first-order necessary conditions for optimality, where $\delta\tilde{J}$ is given by the Gateaux

derivative [86] as:

$$
\begin{aligned}
\delta\tilde{J} &= \lim_{\epsilon\to 0}\frac{1}{\epsilon}\left[\tilde{J}\left(x+\epsilon\eta,\dot{x}+\epsilon\dot{\eta},u+\epsilon\mu(s)\right)-\tilde{J}(x,\dot{x},u(s))\right] \\
&= \lim_{\epsilon\to 0}\frac{1}{\epsilon}\Bigg[\psi\left(x(t_f)+\epsilon\eta(t_f)\right)-\psi\left(x(t_f)\right)+\int_{t_0}^{t_f}\int_{\delta}^{\Delta} I\left(x(\tau)+\epsilon\eta(\tau),u(s)+\epsilon\mu(s)\right)-\mathcal{I}\left(x(\tau),u(s)\right) \\
&\quad +\lambda^T\left(\frac{1}{\Delta-\delta}\left(\dot{x}(\tau)+\epsilon\dot{\eta}(\tau)-\dot{x}(\tau)\right)-\tilde{f}\left(x(\tau)+\epsilon\eta(\tau),u(s)+\epsilon\mu(s)\right)+\tilde{f}\left(x(s),u(s)\right)\right)dsd\tau\Bigg]
\end{aligned}
\tag{8.24}
$$

From now on, we drop the dependence on s and t when appropriate for ease of reading. The Taylor expansion of $I(x+\epsilon\eta,u+\epsilon\mu)$ around (x,u) is given by

$$
\mathcal{I}\left(x+\epsilon\eta,u+\epsilon\mu\right)=\mathcal{I}\left(x,u\right)+\epsilon\frac{\partial\mathcal{I}}{\partial x}\eta+\epsilon\frac{\partial\mathcal{I}}{\partial u}\mu,
$$

Similarly, the Taylor expansion of $\tilde{f}(x+\epsilon\eta,u+\epsilon\mu)$ around (x,u) is given by

$$
\tilde{f}(x+\epsilon\eta,u+\epsilon\mu)=\tilde{f}(x,u)+\epsilon\frac{\partial\tilde{f}}{\partial x}\eta+\epsilon\frac{\partial\tilde{f}}{\partial u}\mu,
$$

where $\tilde{f}(x,u)$ is defined in (8.5) and using lemma 8.2 and 8.1

$$
\frac{\partial\tilde{f}}{\partial u}=-\mathcal{D}^{\text{in}}\text{diag}\left(\mathcal{D}^T x\right)\left(\text{diag}(\mathbf{1}_{s<\alpha(x)}\otimes\mathbf{1}_d)\right)=-\mathcal{D}^{\text{in}}\text{diag}\left(\mathcal{D}^T x\odot(\mathbf{1}_{s<\alpha(x)}\otimes\mathbf{1}_d)\right).
$$

The Jacobian $\frac{\partial\tilde{f}}{\partial x}$ is given in (8.10).

So, equation (8.24) can be simplified to

$$\delta\tilde{J} = \left.\frac{\partial\psi}{\partial x}\right|_{x=x(t_f)} \eta(t_f) + \int_{t_0}^{t_f}\int_{\delta}^{\Delta} \frac{\partial\mathcal{I}}{\partial x}\eta(\tau) + \frac{\partial\mathcal{I}}{\partial u(s)}\mu(s) + \lambda^T\left(\frac{1}{\Delta-\delta}\dot{\eta}(\tau) - \frac{\partial\tilde{f}}{\partial x}\eta(\tau) - \frac{\partial\tilde{f}}{\partial u(s)}\mu(s)\right) ds d\tau \quad (8.25)$$

Using integration by parts and assuming $\eta(t_0) = 0$ we can remove the dependency of $\dot{\eta}(\tau)$ as:

$$\int_{t_0}^{t_f} \lambda^T\dot{\eta}(\tau)d\tau = \lambda^T(t_f)\eta(t_f) - \int_{t_0}^{t_f} \dot{\lambda}^T\eta(\tau)d\tau. \quad (8.26)$$

So, equation (8.25) become:

$$\delta\tilde{J} = \left.\frac{\partial\psi}{\partial x}\right|_{x=x(t_f)} \eta(t_f) + \lambda^T(t_f)\eta(t_f) + \int_{t_0}^{t_f}\int_{\delta}^{\Delta} \frac{\partial\mathcal{I}}{\partial x}\eta(\tau) + \frac{\partial\mathcal{I}}{\partial u(s)}\mu(s) - \frac{1}{\Delta-\delta}\dot{\lambda}^T\eta(\tau) - \lambda^T\left(\frac{\partial\tilde{f}}{\partial x}\eta(\tau) + \frac{\partial\tilde{f}}{\partial u(s)}\mu(s)\right) ds d\tau \quad (8.27)$$

Factoring similar terms, we can further simplify the expression for δJ

$$\delta\tilde{J} = \left.\frac{\partial\psi}{\partial x}\right|_{x=x(t_f)} \eta(t_f) + \lambda^T(t_f)\eta(t_f) + \int_{t_0}^{t_f}\int_{\delta}^{\Delta} \left(\frac{\partial\mathcal{I}}{\partial x} - \frac{1}{\Delta-\delta}\dot{\lambda}^T - \lambda^T\frac{\partial\tilde{f}}{\partial x}\right)\eta(\tau) + \left(\frac{\partial\mathcal{I}}{\partial u(s)} - \lambda^T\frac{\partial\tilde{f}}{\partial u(s)}\right)\mu(s)\, ds d\tau, \quad (8.28)$$

which can be written as

$$\delta\tilde{J} = \frac{\partial\psi}{\partial x}\bigg|_{x=x(t_f)} \eta(t_f) + \lambda^T(t_f)\eta(t_f)$$
$$+ \int_{t_0}^{t_f} \Big[-\dot{\lambda}^T + \int_{\delta}^{\Delta} \Big(\frac{\partial \mathcal{I}}{\partial x} - \lambda^T \frac{\partial \tilde{f}}{\partial x} \Big) ds \Big] \eta(\tau)\, d\tau + \int_{t_0}^{t_f} \int_{\delta}^{\Delta} \Big(\frac{\partial \mathcal{I}}{\partial u(s)} - \lambda^T \frac{\partial \tilde{f}}{\partial u(s)} \Big) \mu(s)\, ds d\tau, \quad (8.29)$$

Finally, we need to set $\delta\tilde{J} = 0$ to formulate the necessary conditions for optimality:

The boundary condition is given by:

$$\lambda(t_f) = -\Big(\frac{\partial\psi}{\partial x} \Big)^T \bigg|_{x=x(t_f)}, \quad (8.30)$$

with $\psi(x(t_f)) = \|x(t_f) - \hat{x}(t_f)\|^2$, we have $\lambda(t_f) = x(t_f) - \hat{x}(t_f)$.

So, the co-state equation becomes:

$$\dot{\lambda}^T = \int_{\delta}^{\Delta} \Big(\frac{\partial \mathcal{I}}{\partial x} - \lambda^T \frac{\partial \tilde{f}}{\partial x} \Big) ds = (x - \hat{x})^T - \lambda^T \int_{\delta}^{\Delta} \frac{\partial \tilde{f}}{\partial x}\, ds \quad (8.31)$$

Given (8.17), the co-state equation become simplified as

$$\dot{\lambda}^T = (x - \hat{x})^T + \lambda^T \mathcal{D}^{\text{in}}\, \mathcal{W} \mathcal{D}^T, \quad (8.32)$$

with $\mathcal{W}$ given in (5.32) as

$$\mathcal{W}(z) = \text{blkdiag}\left(w_1(\|z_1\|) I_d + \frac{dw_1(\|z_1\|)}{ds} \frac{z_1 z_1^T}{\|z_1\|}, \ldots, w_m(\|z_m\|) I_d + \frac{dw_m(\|z_m\|)}{ds} \frac{z_m z_m^T}{\|z_m\|} \right),$$

We have an additional integral constraint, giving us the optimal policy

$$\tilde{J}_u := \int_{t_0}^{t_f} \frac{\partial \mathcal{I}}{\partial u} - \lambda^T \frac{\partial \tilde{f}}{\partial u(s)} \, d\tau \stackrel{!}{=} 0, \tag{8.33}$$

Expanding, we get

$$\begin{aligned} \tilde{J}_u &= u - u_o + \tfrac{1}{t_f - t_0} \int_{t_0}^{t_f} \lambda^T \mathcal{D}^{\text{in}} \text{diag}\left(\mathcal{D}^T x \odot (\mathbf{1}_{s<\alpha(x)} \otimes \mathbf{1}_d)\right) \\ &= u - u_o + \tfrac{1}{t_f - t_0} \int_{t_0}^{t_f} \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i^{in}} \lambda_i^T (x_i - x_j) \mathbf{1}_{s<\|x_i - x_j\|} \, d\tau \end{aligned} \tag{8.34}$$

By setting (8.34) to zero, we can obtain the optimal policy, $u^*(s)$, and consequently the optimal interaction weights as follows:

$$\begin{aligned} u^*(s) &= u_o(s) \\ &- \frac{1}{t_f - t_0} \int_{t_0}^{t_f} \sum_{i=1}^{N} \sum_{(j,i) \in \mathcal{N}_i^{\text{in}}} \lambda_i^{*^T} (x_i^*(\tau) - x_j^*(\tau)) \mathbf{1}_{s<\|x_i^* - x_j^*\|} \, d\tau, \end{aligned} \tag{8.35}$$

which depends on the graph topology, as dictated by the set of in-neighbors, the co-state, the edge state, z, and its norm. The learned control $u^*(s)$ minimizes the cost, resulting in the optimal interaction weights using (5.12):

$$w^*(\|x_i^* - x_j^*\|, u^*) = \int_{\delta}^{\|x_i^* - x_j^*\|} u^*(s) \, ds, \quad \forall (j,i) \in \mathcal{N}_i^{\text{in}}. \tag{8.36}$$

Bibliography

[1] Sarjana Oradiambalam Sachidanandam, Sara Honarvar, D Sawyer Elliott, Gregory Hicks, and Yancy Diaz-Mercado. Decision-making aid for human-machine teaming in multiplayer pursuit-evasion games. In *AIAA SCITECH 2025 Forum*, page 2272, 2025.

[2] Mohamed A Kamel, Xiang Yu, and Youmin Zhang. Formation control and coordination of multiple unmanned ground vehicles in normal and faulty situations: A review. *Annual reviews in control*, 49:128–144, 2020.

[3] Sarjana Oradiambalam Sachidanandam, Sara Honarvar, and Yancy Diaz-Mercado. Effectiveness of augmented reality for human swarm interactions. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 11258–11264. IEEE, 2022.

[4] Sara Honarvar and Yancy Diaz-Mercado. Geometric graph neural network modeling of human interactions in crowded environments. *IFAC-PapersOnLine*, 58(28):25–30, 2024.

[5] Mehran Mesbahi and Magnus Egerstedt. *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

[6] Reza Olfati-Saber, J Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[7] Craig W Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 25–34, 1987.

[8] William H Warren, J Benjamin Falandays, Kei Yoshida, Trenton D Wirth, and Brian A Free. Human crowds as social networks: collective dynamics of consensus and polarization. *Perspectives on Psychological Science*, 19(2):522–537, 2024.

[9] Sara Honarvar, Mia Caminita, Hossein Ehsani, Hyun Jun Kwon, Yancy Diaz-Mercado, Jin-Oh Hahn, Tim Kiemel, and Jae Kun Shim. Interpersonal motor synergy: coworking strategy depends on task constraints. *Journal of Neurophysiology*, 126(5):1698–1709, 2021.

[10] Mark L Latash, John P Scholz, and Gregor Schöner. Toward a new theory of motor synergies. *Motor control*, 11(3):276–308, 2007.

[11] Raphael Korbmacher and Antoine Tordeux. Review of pedestrian trajectory prediction methods: Comparing deep learning and knowledge-based approaches. *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[12] Nematollah Ab Azar, Aref Shahmansoorian, and Mohsen Davoudi. From inverse optimal control to inverse reinforcement learning: A historical review. *Annual Reviews in Control*, 50:119–138, 2020.

[13] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

[14] John P Scholz and Gregor Schöner. The uncontrolled manifold concept: identifying control variables for a functional task. *Experimental brain research*, 126:289–306, 1999.

[15] Michael A Riley, Michael J Richardson, Kevin Shockley, and Verónica C Ramenzoni. Interpersonal synergies. *Frontiers in psychology*, 2:38, 2011.

[16] Sara Honarvar, Jin-OH Hahn, Tim Kiemel, Jae K Shim, and Yancy Diaz-Mercado. Modeling human-human collaboration: A connection between inter-personal motor synergy and consensus algorithms. In *2022 American Control Conference (ACC)*, pages 677–682, 2022.

[17] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995.

[18] Jur Van Den Berg, Stephen J Guy, Ming Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. In *Robotics Research: The 14th International Symposium ISRR*, pages 3–19. Springer, 2011.

[19] Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14424–14432, 2020.

[20] Yingfan Huang, Huikun Bi, Zhaoxin Li, Tianlu Mao, and Zhaoqi Wang. Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6272–6281, 2019.

[21] Kevin W Rio, Gregory C Dachner, and William H Warren. Local interactions underlying collective motion in human crowds. *Proceedings of the Royal Society B: Biological Sciences*, 285(1878):20180611, 2018.

[22] Dwaipayan Mukherjee and Daniel Zelazo. Robustness of consensus over weighted digraphs. *IEEE Transactions on Network Science and Engineering*, 6(4):657–670, 2018.

[23] Wei Ren and Randal W Beard. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Transactions on automatic control*, 50(5):655–661, 2005.

[24] Reza Olfati Saber and Richard M Murray. Consensus protocols for networks of dynamic agents. *InProceedings of the 2003 American Control Conference*, 2003.

[25] Minyi Huang and Jonathan H Manton. Stochastic consensus seeking with noisy and directed inter-agent communication: Fixed and randomly varying topologies. *IEEE Transactions on Automatic Control*, 55(1):235–241, 2009.

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

[27] Herbert G Tanner, Ali Jadbabaie, and George J Pappas. Flocking in fixed and switching networks. *IEEE Transactions on Automatic control*, 52(5):863–868, 2007.

[28] William B Dunbar and Richard M Murray. Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica*, 42(4):549–558, 2006.

[29] Zhongkui Li, Zhisheng Duan, and Lin Huang. Leader-follower consensus of multi-agent systems. In *2009 American control conference*, pages 3256–3261. IEEE, 2009.

[30] Claudio Altafini. Consensus problems on networks with antagonistic interactions. *IEEE transactions on automatic control*, 58(4):935–946, 2012.

[31] Angela Fontan, Lingfei Wang, Yiguang Hong, Guodong Shi, and Claudio Altafini. Multi-agent consensus over time-invariant and time-varying signed digraphs via eventual positivity. *IEEE Transactions on Automatic Control*, 68(9):5429–5444, 2022.

[32] Abdollah Amirkhani and Amir Hossein Barshooi. Consensus in multi-agent systems: a review. *Artificial Intelligence Review*, 55(5):3897–3935, 2022.

[33] Stephen Boyd. Convex optimization of graph laplacian eigenvalues. In *Proceedings of the International Congress of Mathematicians*, volume 3, pages 1311–1319. Citeseer, 2006.

[34] S Yusef Shafi, Murat Arcak, and Laurent El Ghaoui. Graph weight design for laplacian eigenvalue constraints with multi-agent systems applications. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 5541–5546. IEEE, 2011.

[35] Arpita Ghosh, Stephen Boyd, and Amin Saberi. Minimizing effective resistance of a graph. *SIAM review*, 50(1):37–66, 2008.

[36] Chuangchuang Sun, Ran Dai, and Mehran Mesbahi. Weighted network design with cardinality constraints via alternating direction method of multipliers. *IEEE Transactions on Control of Network Systems*, 5(4):2073–2084, 2018.

[37] Jonathan Hermann and Ulrich Konigorski. Eigenvalue assignment for the laplacian matrix of directed graphs. In *2019 American Control Conference (ACC)*, pages 4036–4042. IEEE, 2019.

[38] Jorge Cortés and Magnus Egerstedt. Coordinated control of multi-robot systems: A survey. *SICE Journal of Control, Measurement, and System Integration*, 10(6):495–503, 2017.

[39] Daniel Zelazo, Simone Schuler, and Frank Allgöwer. Performance and design of cycles in consensus networks. *Systems & Control Letters*, 62(1):85–96, 2013.

[40] Pratik Mukherjee, Andrea Gasparri, and Ryan K Williams. Stable motion and distributed topology control for multi-agent systems with directed interactions. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 3455–3460. IEEE, 2017.

[41] Eduardo Sebastián, Thai Duong, Nikolay Atanasov, Eduardo Montijano, and Carlos Sagues. Learning to identify graphs from node trajectories in multi-robot networks. *arXiv preprint arXiv:2307.04374*, 2023.

[42] Fernando Gama, Ekaterina Tolstaya, and Alejandro Ribeiro. Graph neural networks for decentralized controllers. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5260–5264. IEEE, 2021.

[43] Hassan K Khalil and Jessy W Grizzle. *Nonlinear systems*, volume 3. Prentice hall Upper Saddle River, NJ, 2002.

[44] Winfried Lohmiller and Jean-Jacques E Slotine. On contraction analysis for non-linear systems. *Automatica*, 34(6):683–696, 1998.

[45] Daniel Zelazo, Amirreza Rahmani, and Mehran Mesbahi. Agreement via the edge laplacian. In *2007 46th IEEE Conference on Decision and Control*, pages 2309–2314. IEEE, 2007.

[46] Huaguang Zhang, Tao Feng, Guang-Hong Yang, and Hongjing Liang. Distributed cooperative optimal control for multiagent systems on directed graphs: An inverse optimal approach. *IEEE Transactions on Cybernetics*, 45(7):1315–1326, 2014.

[47] Zhiwen Zeng, Xiangke Wang, and Zhiqiang Zheng. Convergence analysis using the edge laplacian: Robust consensus of nonlinear multi-agent systems via iss method. *International Journal of Robust and Nonlinear Control*, 26(5):1051–1072, 2016.

[48] Esteban Restrepo, Antonio Loría, Ioannis Sarras, and Julien Marzat. Robust rendezvous control of uavs with collision avoidance and connectivity maintenance. In *2022 American Control Conference (ACC)*, pages 4733–4738. IEEE, 2022.

[49] Esteban Restrepo, Antonio Loría, Ioannis Sarras, and Julien Marzat. Edge-based strict lyapunov functions for consensus with connectivity preservation over directed graphs. *Automatica*, 132:109812, 2021.

[50] Emanuel Todorov. Optimality principles in sensorimotor control. *Nature neuroscience*, 7(9):907–915, 2004.

[51] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1, 2004.

[52] Wanxin Jin, Zhaoran Wang, Zhuoran Yang, and Shaoshuai Mou. Pontryagin differentiable programming: An end-to-end learning and control framework. *Advances in Neural Information Processing Systems*, 33:7979–7992, 2020.

[53] Wanxin Jin and Shaoshuai Mou. Distributed inverse optimal control. *Automatica*, 129:109658, 2021.

[54] Yizhi Zhou, Wanxin Jin, and Xuan Wang. D3g: Learning multi-robot coordination from demonstrations. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8898–8904. IEEE, 2024.

[55] Nikolai Bernstein. The co-ordination and regulation of movements. *The co-ordination and regulation of movements*, 1966.

[56] Yushin Kim, Kyung Koh, and Jae Kun Shim. Inter-dependence between mathematically independent variability components in human multi-finger force control. *Neuroscience research*, 158:16–20, 2020.

[57] Jae Kun Shim, Mark L Latash, and Vladimir M Zatsiorsky. Prehension synergies in three dimensions. *Journal of neurophysiology*, 93(2):766–776, 2005.

[58] Jae Kun Shim and Jaebum Park. Prehension synergies: principle of superposition and hierarchical organization in circular object prehension. *Experimental brain research*, 180(3):541, 2007.

[59] Verónica C Ramenzoni, Michael A Riley, Kevin Shockley, and Aimee A Baker. Interpersonal and intrapersonal coordinative modes for joint and single task performance. *Human Movement Science*, 31(5):1253–1267, 2012.

[60] Andreas Daffertshofer, Claudine JC Lamoth, Onno G Meijer, and Peter J Beek. Pca in studying coordination and variability: a tutorial. *Clinical biomechanics*, 19(4):415–428, 2004.

[61] David P Black, Michael A Riley, and Christopher K McCord. Synergies in intra-and interpersonal interlimb rhythmic coordination. *Motor control*, 11(4):348–373, 2007.

[62] Sara Honarvar, Chang-Sei Kim, Yancy Diaz-Mercado, Kyoung Koh, Hyun J Kwon, Tim Kiemel, Mia Caminita, Jin-Oh Hahn, and Jae K Shim. Unveiling the neuromechanical mechanisms underlying the synergistic interactions in human sensorimotor system. *Scientific Reports*, 11(1):1–16, 2021.

[63] Gennaro Notomista and Magnus Egerstedt. Constraint-driven coordinated control of multi-robot systems. In *2019 American Control Conference (ACC)*, pages 1990–1996. IEEE, 2019.

[64] Richard Lukas Bumann, Francisco J Valero-Cuevas, and Robert Riener. Using load-cells to unveil limitations to the human movement system, 2012.

[65] Frank L Lewis, Lihua Xie, and Dan Popa. *Optimal and robust estimation: with an introduction to stochastic control theory*. CRC press, 2017.

[66] Daniel M Wolpert and Zoubin Ghahramani. Computational principles of movement neuroscience. *Nature neuroscience*, 3(11):1212–1217, 2000.

[67] Jiahu Qin, Wei Xing Zheng, and Huijun Gao. Coordination of multiple agents with double-integrator dynamics under generalized interaction topologies. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(1):44–57, 2011.

[68] Les G Carlton, Mary J Carlton, and KM Newell. Reaction time and response dynamics. *The Quarterly Journal of Experimental Psychology*, 39(2):337–360, 1987.

[69] Anthony N Carlsen, Romeo Chua, J Timothy Inglis, David J Sanderson, and Ian M Franks. Differential effects of startle on reaction time for finger and arm movements. *Journal of neurophysiology*, 101(1):306–314, 2009.

[70] Daniel M Wolpert, Zoubin Ghahramani, and Michael I Jordan. An internal model for sensorimotor integration. *Science*, 269(5232):1880–1882, 1995.

[71] Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M Kitani, Dariu M Gavrila, and Kai O Arras. Human motion trajectory prediction: A survey. *The International Journal of Robotics Research*, 39(8):895–935, 2020.

[72] Christoforos Mavrogiannis, Francesca Baldini, Allan Wang, Dapeng Zhao, Pete Trautman, Aaron Steinfeld, and Jean Oh. Core challenges of social robot navigation: A survey. *ACM Transactions on Human-Robot Interaction*, 12(3):1–39, 2023.

[73] Trenton David Wirth, Gregory C Dachner, Kevin W Rio, and William H Warren. The neighborhood of interaction in human crowds is neither metric nor topological, but visual. *bioRxiv*, pages 2022–08, 2022.

[74] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 961–971, 2016.

[75] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2255–2264, 2018.

[76] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[77] Renato Grasso, Pascal Prévost, Yuri P Ivanenko, and Alain Berthoz. Eye-head coordination for the steering of locomotion in humans: an anticipatory synergy. *Neuroscience Letters*, 253(2):115–118, 1998.

[78] Zhuangzhuang Yang, Chengxin Pang, and Xinhua Zeng. Trajectory forecasting using graph convolutional neural networks based on prior awareness and information fusion. *ISPRS International Journal of Geo-Information*, 12(2):77, 2023.

[79] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[80] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th international conference on computer vision*, pages 261–268. IEEE, 2009.

[81] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. *Computer Graphics Forum*, 26(3):655–664, 2007.

[82] Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian Reid, Hamid Rezatofighi, and Silvio Savarese. Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[83] Pei Lv, Wentong Wang, Yunxin Wang, Yuzhen Zhang, Mingliang Xu, and Changsheng Xu. Ssagcn: social soft attention graph convolution network for pedestrian trajectory prediction. *IEEE transactions on neural networks and learning systems*, 2023.

[84] Wei Ren, Randal W Beard, and Timothy W McLain. Coordination variables and consensus building in multiple vehicle systems. In *Cooperative Control: A Post-Workshop Volume 2003 Block Island Workshop on Cooperative Control*, pages 171–188. Springer, 2005.

[85] Eduardo D Sontag and Yuan Wang. On characterizations of the input-to-state stability property. *Systems & Control Letters*, 24(5):351–359, 1995.

[86] Daniel Liberzon. *Calculus of variations and optimal control theory: a concise introduction*. Princeton university press, 2011.

[87] Suzanne Lenhart and John T Workman. *Optimal control applied to biological models*. Chapman and Hall/CRC, 2007.

[88] Eduardo Casas, Mariano Mateos, and Fredi Tröltzsch. Necessary and sufficient optimality conditions for optimization problems in function spaces and applications to control theory. In *ESAIM: Proceedings*, volume 13, pages 18–30. EDP Sciences, 2003.

[89] Larry Armijo. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of mathematics*, 16(1):1–3, 1966.

[90] Miles Johnson, Navid Aghasadeghi, and Timothy Bretl. Inverse optimal control for deterministic continuous-time nonlinear systems. In *52nd IEEE Conference on Decision and Control*, pages 2906–2913. IEEE, 2013.

[91] Petros A Ioannou and Jing Sun. *Robust adaptive control*, volume 1. PTR Prentice-Hall Upper Saddle River, NJ, 1996.

[92] Daniel Zelazo and Mehran Mesbahi. Edge agreement: Graph-theoretic performance bounds and passivity analysis. *IEEE Transactions on Automatic Control*, 56(3):544–555, 2010.

[93] Zak Costello. *Distributed computation in networked systems*. PhD thesis, Georgia Institute of Technology, 2016.

[94] Shuangzhe Liu, Gotz Trenkler, et al. Hadamard, khatri-rao, kronecker and other matrix products. *Int. J. Inf. Syst. Sci*, 4(1):160–177, 2008.