

Optimization-based Haptic Feedback Synthesis in Human-Multi-Robot Systems

by

Ramisha Anjum

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical And Computer Engineering

Waterloo, Ontario, Canada, 2024

© Ramisha Anjum 2024

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Chapters 1, 2, and 6 were solely authored by Ramisha Anjum. The work presented in these chapters includes formations of the introductory material, literature reviews, methodologies, and result analysis.

Chapters 3, 4 and 5 are based on a collaborative research project conducted with my supervisor Gennaro Notomista. The content in these chapters has been developed from a paper co-authored by Ramisha Anjum and Gennaro Notomista, which is submitted for a possible publication at the 2025 Joint IFAC Symposium on Mechatronic Systems & Robotics. Ramisha Anjum was primarily responsible for the conceptualization, experimentation, result analysis and initial drafting of the paper, while supervisor Gennaro Notomista provided guidance in form of Python, ROS and MATLAB modeling, critical revisions, and editorial support.

Abstract

This thesis presents an optimization-based control framework for multi-robot systems interacting with human. This framework is amenable for both centralized and decentralized control strategies, and addresses human-robot interaction as well as multi-task allocation and execution. The novelty of the framework lies in its ability to leverage optimization algorithms to manage task allocation while ensuring system passivity, which is critical to maintaining stability and safe interaction between robots and human operators. Both single-integrator and double-integrator robot dynamic models are explored for different control scenarios. While the single-integrator scenario employs decentralized control, ensuring scalability and efficient task execution, the double-integrator scenario typically utilizes centralized control. However, if human-robot interaction constraints are distributed between robots, a decentralized approach can be formulated, allowing flexible control structures.

Experimental results demonstrate the system’s ability to dynamically allocate tasks, execute them efficiently, and provide haptic feedback to human operators. The human-robot interaction mechanism combines passivity-based control with feedback synthesis, ensuring that the system can adapt to real-time changes in human input while maintaining stability and safety. Additionally, the thesis identifies the limitations of a purely centralized approach, particularly in larger robot teams, and suggests future improvements, including enhanced scalability, improved feedback mechanisms, and adaptations for dynamic real-world environments.

This research advances the field of human-robot collaboration by offering a robust, adaptable framework capable of handling complex multi-task scenarios. The framework is particularly suited for applications in dynamic and interactive settings, such as telerobotics, industrial automation, and other collaborative operations where human oversight and real-time control play a crucial role.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Dr. Gennaro Notomista, for his invaluable guidance, limitless patience, and tireless efforts in shaping both my understanding and this thesis. His willingness to engage in lengthy, thought-provoking discussions has provided me with profound intellectual insights, of which this thesis represents only a modest fraction.

I am sincerely thankful to Dr. Oliver Schneider and Dr. Yash Vardhan Pant for their meticulous feedback and thoughtful critiques. Their effort in carefully reading through this work has been instrumental in refining its quality.

My heartfelt gratitude and deepest respect go to my dear younger brother, MD. Salahuddin Fahad Rashik, for his unwavering support, encouragement, and companionship, even across significant time zones. To my beloved parents and grandparents (especially my grandfather MD. Abdul Latif), I owe immense thanks for their constant love, sacrifices, and the strong values they instilled in me, shaping the person I am today.

To the love of my life, I extend my profound appreciation for his steadfast support, unwavering belief in me, and encouragement throughout this journey. Your presence has been a pillar of strength during this challenging, yet rewarding endeavor.

I am deeply grateful to my labmates, Justine Shaw, Andrew Adel Shenouda Nasif, Sheikh Abrar Tahmid, Longchen Niu (Mason), Ziyi Wang, visiting student of our lab and good friend Takumi Ito, and visiting scholar of our lab, Dr. Satoshi Nakano, for their invaluable feedback, guidance, and camaraderie throughout my academic and research journey at Waterloo. Your support has been an integral part of this experience.

I would like to express my heartfelt gratitude to my best friends, Kazi Tushita Tahsin, Syeda Lamia Tabassum, and Mushfiqul Islam, for their unwavering support and encouragement. Despite the challenges of time differences, your unconditional friendship has been a source of immense strength and comfort throughout this journey.

To my dear sisters, Najratun Nayeem Pinky and Humayra Tasnim, thank you for your unwavering kindness, encouragement, and companionship, which have been a source of strength and inspiration. I also would like to extend my heartfelt thanks to my friends, Shovon Sudan Saha, Ahmed Shahriar Sakib, and Nazia Tabassum Shuvechha, whose support and friendship have been invaluable throughout this process.

I want to acknowledge Haruichi Furudate for his manga Haikyuu!!, which has been both a source of inspiration and a much-needed escape during this journey. His work reminded me of the importance of persistence and finding joy in even the most challenging moments.

With this thesis, I feel I have successfully completed a significant chapter in my academic journey, a milestone that has brought me closer to my goals.

Finally, I would like to thank Allah for providing me with the strength and abilities to complete this work.

Dedication

This thesis is dedicated to my beloved grandparents (especially my grandfather MD. Abdul Latif), my beloved parents, my dear younger brother, and most importantly the love of my life.

Table of Contents

Author's Declaration	ii
Statement of Contributions	iii
Abstract	iv
Acknowledgements	v
Dedication	vii
List of Figures	xi
1 Introduction	1
2 Related Work	8
2.1 Optimization Based Task Execution	8
2.2 Human-Multi-Robot Teleoperation	12
3 Problem Formulation	18
3.1 System Model and Mathematical Background	18
3.2 Passive Teleoperation	21

4	Haptic Feedback Synthesis Framework	23
4.1	Single-Integrator Dynamic System	23
4.1.1	Model of the Tasks	23
4.1.2	Optimization-based Decentralized Control Framework	28
4.2	Double-Integrator Dynamic System	30
4.2.1	Model of the Tasks	30
4.2.2	Optimization-based Centralized Control Framework	37
5	Simulations and Experimental Results	39
5.1	Single-Integrator Dynamic System Simulation	39
5.1.1	Multi-Robot Multi-Task Execution without Human Feedback Synthesis	40
5.1.2	Multi-Robot Multi-Task Execution with Human Feedback Synthesis	44
5.2	Double-Integrator Dynamic System Simulation and Experiment	46
5.2.1	Centralized Multi-Robot Multi-Task Execution with Passivity and Human Feedback	47
5.2.2	Centralized Multi-Robot Multi-Task Execution without Passivity and with Human Feedback	56
5.3	Discussions and Limitations	64
5.3.1	Discussions	65
5.3.2	Limitations	65
6	Conclusions	68
	References	70
	APPENDICES	77
A	Matlab Simulation of Different Tasks	78
A.1	Multi-Robot Multi-Task Execution with Only Go-to-Goal Task and Cover- age Control	78

B Python Simulation of Different Tasks	82
B.1 Multi-Robot Multi-Task Execution with Only Go-to-Goal Task	82

List of Figures

1.1	Passive optimization-based approach for the haptic feedback synthesis for multi-robot multi-task systems. The robots receive the input u_H from a human operator through a haptic device, such as the Novint Falcon, which is used in this work as an example implementation. The passivity-based haptic synthesis algorithm takes the human input, u_H and state of the robots, x , to produce the optimal control input for the robots, u^* , and the haptic feedback for the human, Δu_H^* .	3
1.2	Passive optimization-based approach for the haptic feedback synthesis for multi-robot multi-task systems for the real world application. The team of DJI Robomastes receive the input u_H from a human operator through a haptic device (Novint Falcon). The passivity-based haptic synthesis algorithm takes the human input, u_H and state of the robots, x , to produce the optimal control input for the robots, u^* , and the haptic feedback for the human, Δu_H^* .	4
1.3	optimization-based approach for the haptic feedback synthesis for multi-robot multi-task systems. The robots receive the input u_H from a human operator through a haptic device. The optimization algorithm takes the human input, u_H and state of the robots, x , to produce the optimal control input for the robots, u^* , and the haptic feedback for the human, Δu_H^* .	5

4.1	Schematic representation of the Human-Robot Interaction framework in a double-integrator dynamic system. The human operator provides velocity commands (u_H) to a team of robots. These commands are typically derived from the position or velocity inputs of the haptic device, depending on the specific implementation. The robots send feedback (Δu_H) through the haptic device, which is mapped to forces or adjustments in velocity to communicate discrepancies between the intended and actual movements, ensuring alignment and stability in the interaction.	34
5.1	Multi-robot multi-task execution without human input in Python. The four robots are represented by circles of four colors (red, green, blue, and brown, respectively). The green cross mark depicts a designated goal location. The robots assemble a rectangular shape using formation control (highest priority task for robots 1, 3, and 4) as robot 2 eventually reaches the goal (its highest priority task).	41
5.2	Snapshots of the simulations performed in Webot implemented using ROS2 humble. The pictures show four DJI Mavic 2 Pro quadcopters starting from their initial positions (Fig. 5.2a), moving in the environment towards the goal location (gray square in the lower right corner) and assembling a rectangular formation (Figures 5.2b, 5.2c, 5.2d, 5.2e), and reaching their final configuration (Fig. 5.2f).	42
5.3	Control input vectors of the four robots with respect to time. As can be seen, the control signals produced by the task allocation optimization algorithm are smooth throughout the simulation.	42
5.4	Slack variables of the four robots with respect to time. The smaller the value of $\delta_{i,j}$, the higher the priority of task j for robot i	43
5.5	Multi-robot multi-task execution with human feedback synthesis. Circles in the colors red, green, blue, and brown, respectively, stand in for the four robots. The green cross mark depicts the desired goal for task T_1 , while brown cross represents the goal where a human operator drives robot 4. Due to a human input u_H to robot 4, robots are unable to form a rectangle. Robot 4 is prevented from reaching the brown cross by an opposing repulsive force, symbolized by a brown arrow.	44

5.6	Snapshots of the simulations performed in Webots and implemented using ROS2 humble. The scenario is similar to the one presented in the previous section, except for the presence of a human input that controls robot 4 to move towards the gray square in the upper right corner. The quadcopters attempt to assemble the desired rectangular shape, while one of them reaches the desired goal (gray square at the bottom right), as prescribed by its task priorities. The task execution is traded off with the human input, and the mismatch between desired and actual robot behavior is synthesized in the optimal haptic feedback proportional to Δu_H^*	45
5.7	Human feedback with respect to time. Δu_H does not converge to 0, as a result there will be a non-zero haptic feedback.	45
5.8	Snapshots of the simulations of centralized multi-robot multi-task execution with passivity and human feedback implemented using Python. The pictures show four robots starting from their initial positions (Fig. 5.8a), moving in the environment towards the goal location (black cross (p_G) in the upper left corner) and assembling a square formation (Figures 5.8b, 5.8c, 5.8d, 5.8e), and reaching their final configuration (Fig. 5.8f).	48
5.9	Snapshots of the experiments of centralized multi-robot multi-task execution with passivity and human feedback performed in ROS2 humble using Webot as the simulator. The pictures show four DJI Mavic 2 Pro quadcopters starting from their initial positions (Fig. 5.9a) represented by four gray squares in the bottom left corner, moving in the environment towards the goal location (gray square in the upper right) and assembling a square formation (Figures 5.9b, 5.9c, 5.9d, 5.9e), and reaching their final configuration (Fig. 5.9f).	49
5.10	Snapshots of the experiments of centralized multi-robot multi-task execution with passivity and human feedback performed in the Robohub at the University of Waterloo. The pictures show four DJI Robomaster EP mobile platforms starting from their initial positions (Fig. 5.10a), moving in the environment towards the goal location and assembling a square formation (Figures 5.10b, 5.10c, 5.10d, 5.10e), and reaching their final configuration (Fig. 5.10f).	50

5.11	Control input vectors of the four robots with respect to time. As can be seen, the control signals produced by the optimization algorithm demonstrates the effectiveness of the passivity-based control system in managing the robots' responses under real-world experimental conditions. It confirms that the control inputs, despite fluctuations, are constrained within safe limits, ensuring smooth and stable operations for all robots.	51
5.12	Human feedback with respect to time. Δu_H does not converge to 0, as a result there will be a non-zero haptic feedback.	51
5.13	Formation graph of the four robots with respect to time. As can be seen, the smooth oscillations in T_1 indicate that the system dynamically adjusts the velocities of the robots to maintain the desired formation.	52
5.14	Passivity graph of the four robots with respect to time. As can be seen, the system's velocity closely follows the human input, but without significant deviation, which suggests that the control strategy is successfully enforcing the passivity condition.	52
5.15	x_G and p_G with respect to time. As can be seen, the robot system, while adhering to passivity constraints, adjusts smoothly toward the goal.	52
5.16	u_H and $u_H + \Delta u_H$ with respect to time. As can be seen, The difference between the u_H and $u_H + \Delta u_H$ lines indicates the magnitude of the adjustments made by the system. If the adjustments (Δu_H) are small, it suggests that the system is only making minor corrections to maintain stability. Larger deviations might indicate more significant interventions by the system to correct potential issues or instabilities.	53
5.17	The evolution of the velocity components x_2 for each of the four robots over time. As can be seen, the velocities of all the robots start at certain initial values and evolve over time. The general trend of these lines indicates how each robot's velocity changes in response to the control inputs, including any adjustments made by the system to maintain formation and stability.	53
5.18	Snapshots of the simulations of centralized multi-robot multi-task execution without passivity and with human feedback implemented using Python. The scenario is similar to the one presented in the previous section. The robots attempt to assemble the desired square shape and move towards the goal (black cross (p_G) at the upper left) but the shape of the formation is not a proper square like previous case and finally the center of the formed square reaches the desired goal (black cross (p_G) at the upper left).	57

5.19	Snapshots of the experiments of centralized multi-robot multi-task execution without passivity and with human feedback performed in ROS2 humble using Webot as the simulator. The scenario is similar to the one presented in the previous section. The DJI Mavic 2 Pro quadcopters attempt to assemble the desired square shape and move towards the goal and but cannot maintain the square formation and end up crashing into each other and fall on the ground.	58
5.20	Snapshots of the experiments of centralized multi-robot multi-task execution without passivity and with human feedback performed in the Robohub at the University of Waterloo. The scenario is similar to the one presented in the previous section. The DJI Robomaster EP mobile platforms attempt to assemble the desired square shape and move towards the goal but cannot maintain the square formation and bump into each other.	59
5.21	Control input vectors of the four robots with respect to time. As can be seen, the control inputs exhibits significantly larger and more erratic fluctuations, showing that the system becomes less stable and more prone to instability when passivity control is removed.	60
5.22	Human feedback with respect to time. Δu_H does not converge to 0, as a result there will be a non-zero haptic feedback.	60
5.23	Formation graph of the four robots with respect to time. As can be seen, the smooth oscillations in T_1 indicate that the system dynamically adjusts the velocities of the robots to maintain the desired formation.	60
5.24	Passivity graph of the four robots with respect to time in case of absence of passivity. Without passivity control, the system may exhibit less predictable or more aggressive responses to human input, as indicated by the larger velocity fluctuations. This could imply that the system is more prone to instability or oscillations, especially if the human input is not smooth or consistent.	61
5.25	x_G and p_G with respect to time. As can be seen, the robot system, while adhering to passivity constraints, adjusts smoothly toward the goal.	61
5.26	u_H and $u_H + \Delta u_H$ with respect to time. As can be seen, The difference between the u_H and $u_H + \Delta u_H$ lines indicates the magnitude of the adjustments made by the system. If the adjustments (Δu_H) are small, it suggests that the system is only making minor corrections to maintain stability. Larger deviations might indicate more significant interventions by the system to correct potential issues or instabilities.	62

5.27	The evolution of the velocity components x_2 for each of the four robots over time. As can be seen, the velocities of all the robots start at certain initial values and evolve over time. The general trend of these lines indicates how each robot's velocity changes in response to the control inputs, including any adjustments made by the system to maintain formation and stability. .	62
A.1	Simulation of robot 1 (R_1) going to its own voronoi cell's centroid G_1 (T_4) and robot 2 (R_2) going to goal P_2 (T_1).	79
A.2	Simulation of robot 1 (R_1) after reaching its own voronoi cell's centroid G_1 (T_4) and robot 2 (R_2) after reaching goal P_2 (T_1).	80
B.1	Multi-robot multi-task execution in Python. The four robots (Robot 1, Robot 2, Robot 3, and Robot 4) are represented by circles of four colors (red, green, blue, and brown, respectively). The red, green, blue, and brown cross marks depict four designated goal locations respectively. The robots move towards its own predefined goals (highest priority task for robots 1, 2, 3, and 4).	83
B.2	Snapshots of the simulations performed in Webot implemented using ROS2 humble. The pictures show four DJI Mavic 2 Pro quadcopters starting from their initial positions represented by the four gray square in the middle (Fig. B.2a), moving in the environment towards their own predefined goal locations represented by gray square in the upper left corner, gray square in the upper right corner, gray square in the bottom left corner, and gray square in the bottom right corner respectively (Figures B.2b, B.2c), and reaching their goal finally (Fig. B.2d).	83

Chapter 1

Introduction

Human-robot interaction has become an essential area of research as robots are increasingly integrated into complex, real-world environments. The ability of robots to work alongside humans, adapting to their commands and responding to dynamic changes, is critical for applications ranging from industrial automation to healthcare and rescue missions. In these settings, robots must not only operate autonomously but also collaborate effectively with human operators to achieve common goals. The success of such interactions hinges on developing systems that can seamlessly integrate human input, process feedback, and maintain stability and safety during collaborative tasks.

Human operators, through their cognitive capabilities, can provide high-level oversight and guidance, significantly improving the robots' decision-making processes, as demonstrated in applications such as robotic surgery, where surgeons guide autonomous tools for precision tasks [1], or search and rescue operations, where human intervention helps prioritize tasks in dynamic and uncertain environments [2]. A structured framework that allows humans to interface with multi-robot teams can leverage this advantage, ensuring that robots dynamically adjust their behaviors based on real-time instructions and mission demands. By integrating human decision-making into the control loop, these systems can perform more effectively in unpredictable and dynamic environments, as evidenced by examples such as robotic-assisted disaster response, where human operators guide robots in navigating debris and identifying survivors [3], and collaborative industrial tasks, where human workers provide real-time corrections to robotic actions for tasks requiring precision and adaptability [4].

Multi-robot systems are capable of executing multiple task concurrently. Therefore, effective multi-robot task allocation techniques are required to coordinate several tasks

among robots working in a team [5]. The efficient coordination and task allocation of multiple robots have been the focus of extensive research in the field of multi-robot systems. These systems find applications in various domains, such as search and rescue operations in disaster response [6], environmental monitoring [7], [8], exploration, and reconnaissance [9], where multiple robots are required to collaborate and achieve common objectives. One crucial challenge in such scenarios is to allocate tasks to individual robots effectively, considering their capabilities, priorities, and the overall mission objectives [10].

A critical area of research is the efficient coordination and control of multi-robot systems in environments with dynamic tasks and human interaction. Conventional methods, like formation control and task distribution, have demonstrated their effectiveness in applications such as environmental monitoring, where robots coordinate to cover large areas [11], and search and rescue missions, where task distribution enables efficient exploration and victim localization [12]. But the integration of passivity-based control into multi-robot systems represents a significant advancement, as it guarantees robustness and stability in the presence of unknowns and external inputs. This contribution is particularly impactful in scenarios involving human-robot interaction, where maintaining passivity ensures that the system remains stable and safe despite unpredictable human inputs. In this thesis, we extend this concept to the multi-robot context, enabling both centralized and decentralized frameworks to incorporate passive control mechanisms, which is a key contribution of this work.

Passivity, a property derived from energy-based control theory, guarantees that the system does not generate energy, thereby ensuring stability. This concept is particularly important in human-robot interaction, where the human operator's input can vary unpredictably. The implementation of passivity within the control framework allows the system to maintain stability while still responding to human inputs in a controlled and predictable manner.

Human-robot interaction introduces another level of complexity, as the robots must not only execute their tasks efficiently but also adapt to real-time inputs from human operators. This real-time adaptability becomes critical in applications such as search and rescue missions, where human operators can offer high-level guidance while robots perform individual tasks. In such contexts, it is necessary to maintain stability, safety, and passivity in the system, ensuring that robot behavior remains predictable and safe despite human involvement [13], [14]. This thesis focuses on integrating human input into a multi-robot system through an optimization-based teleoperation framework that guarantees these key properties.

This research stems from the increasing need for multi-robot systems in real-world

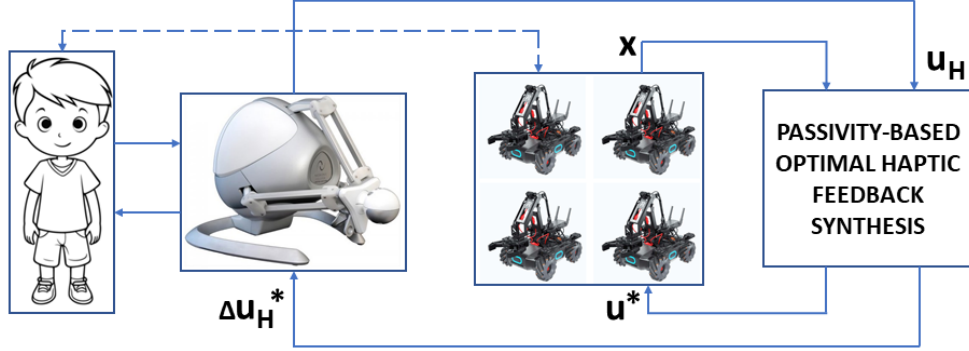


Figure 1.1: Passive optimization-based approach for the haptic feedback synthesis for multi-robot multi-task systems. The robots receive the input u_H from a human operator through a haptic device, such as the Novint Falcon, which is used in this work as an example implementation. The passivity-based haptic synthesis algorithm takes the human input, u_H and state of the robots, x , to produce the optimal control input for the robots, u^* , and the haptic feedback for the human, Δu_H^* .

applications where robot teams have to interact with humans, make decisions online, and give feedback based on the evolution of their mission in real time. Human operators, thanks to their high level cognitive capabilities, have the potential of enhancing the efficiency of task execution. In order to fully take advantage of the presence of human operators, a systematic way of interfacing a human with a team of robots is required. Figure 1.1 depicts the centralized control framework proposed in this thesis, which integrates passivity, formation control, and human-robot interaction into a unified system. This framework represents a novel contribution, as it is among the first to combine these elements cohesively for multi-robot systems, enabling safe and efficient operation in dynamic environments with human input. It consists of an optimization-based approach for the haptic feedback synthesis for multi-robot multi-task systems. The robot state, x , is used together with the human input u_H in order to synthesize the optimal robot control input u^* and haptic feedback for the human operator Δu_H^* . This centralized method allows for a more cohesive and integrated control strategy, ensuring that the collective behavior of the robots adheres to the desired objectives. The passivity-based approach used in this framework ensures that the interaction between the human operator and the robots remains stable, even as the robots work together to maintain a formation and accomplish tasks.

Figure 1.2 depicts the real world scenario of the centralized control framework proposed in this thesis, which integrates passivity, formation control, and human-robot interaction into a unified system that clarifies the 2-dimensional formulation of the problem.

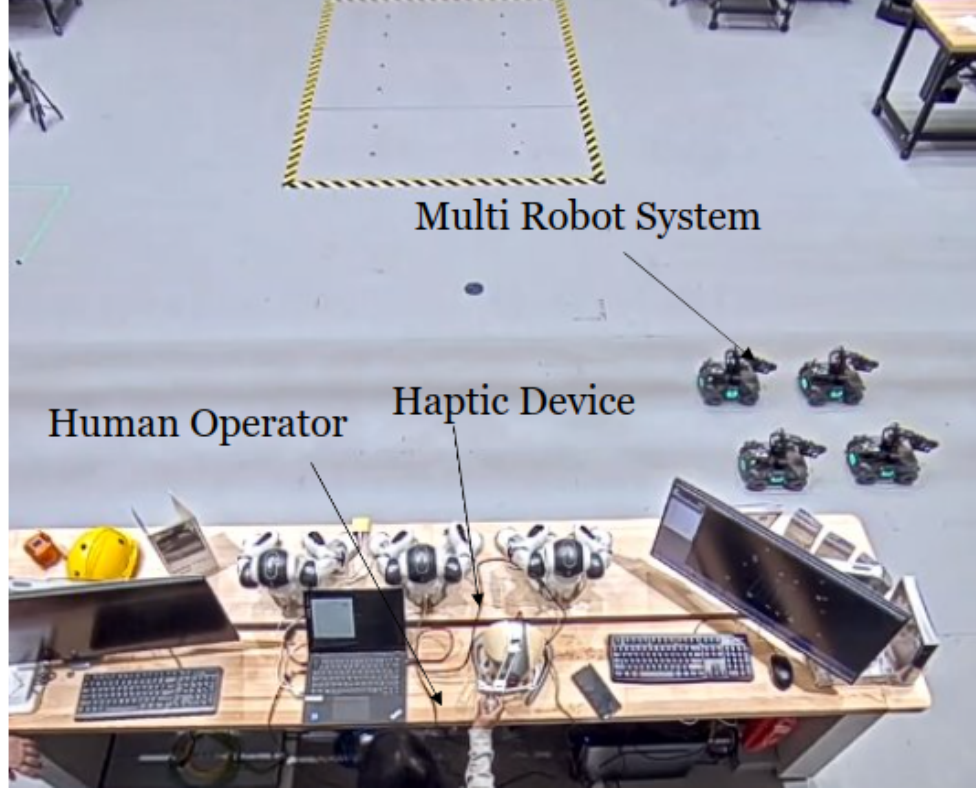


Figure 1.2: Passive optimization-based approach for the haptic feedback synthesis for multi-robot multi-task systems for the real world application. The team of DJI Robomasters receive the input u_H from a human operator through a haptic device (Novint Falcon). The passivity-based haptic synthesis algorithm takes the human input, u_H and state of the robots, x , to produce the optimal control input for the robots, u^* , and the haptic feedback for the human, Δu_H^* .

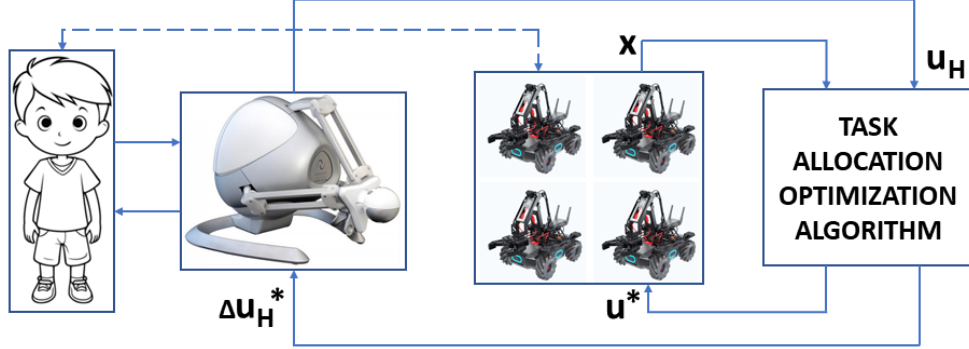


Figure 1.3: optimization-based approach for the haptic feedback synthesis for multi-robot multi-task systems. The robots receive the input u_H from a human operator through a haptic device. The optimization algorithm takes the human input, u_H and state of the robots, x , to produce the optimal control input for the robots, u^* , and the haptic feedback for the human, Δu_H^* .

Figure 1.3 depicts the decentralized control framework we propose in this thesis. It consists of a haptic feedback synthesis for multi-robot multi-task systems. The robot state, x , is used together with the human input u_H in order to synthesize the optimal robot control input u^* and haptic feedback for the human operator Δu_H^* .

This thesis focuses on common applications where teams of robots have to execute multiple tasks. Task allocation in multi-robot systems is a well-studied problem that has garnered considerable attention from researchers in recent years. The challenge lies in finding effective strategies to distribute tasks among the robots efficiently, while considering various factors such as distance travelled, workload balance, energy consumption, communication overhead, and individual robot capabilities. Additionally, the task allocation process should be robust to uncertainties, dynamically adapting to changes in the environment or in the team composition.

However, the control strategies introduced in these works are primarily focused on controlling robot behavior in relatively simple, single-task or single-degree-of-freedom settings. The challenge increases significantly in multi-robot, multi-task environments where human inputs must interact with a variety of control objectives. The goal of this thesis is to extend these control methodologies to multi-task, multi-robot systems, where each robot may have to execute several tasks simultaneously, often with differing priorities. Furthermore, the incorporation of haptic feedback synthesis will enable human operators to dynamically adjust their inputs based on real-time feedback from the robot system.

This work builds on previous research [10] that focuses on task allocation and control in multi-robot systems, by extending the framework to include human teleoperation and haptic feedback mechanisms. The novelty lies in integrating human-in-the-loop control with passivity-based optimization frameworks, ensuring stability and adaptability during human-robot interaction. This extension is valuable as it enables the safe and efficient operation of multi-robot systems in dynamic environments. In this thesis, we have proposed a novel centralized and decentralized control framework for multi-robot, multi-task systems with human-robot interaction, including haptic feedback synthesis. In the single-integrator scenario, a decentralized framework ensures scalability, allowing robots to execute tasks based on local information. In contrast, the double-integrator model allows for centralized control with feedback mechanisms, ensuring smooth performance even in scenarios where human inputs dynamically alter the robots' objectives.

The primary objective of this thesis is to propose a systematic approach to input human instructions to specific robots and get feedback about how this input affects the system. This is achieved while allocating specific tasks to the robots based on their capabilities via predefined priorities. The framework is applicable to a wide variety of tasks [15]. As an example, in the Simulations and Experimental Results chapter of this thesis, we consider the following tasks, which have significantly different nature: Single-robot navigation, multi-robot formation control, passivity-based control, coverage control, and human-robot interaction. These tasks will be implemented taking into account each robot's priorities. A convex optimization formulation allows the task allocation algorithm to run in polynomial time.

The aim has been to optimize factors such as overall energy efficiency and communication overhead by designing decentralized framework among the robots. The main contribution of this thesis to the state of the art is the introduction of a systematic approach to synthesize haptic feedback in human-multi-robot systems and introduction of a passivity-based control mechanism that integrates with formation control and human-robot interaction to ensure stability across the entire multi-robot system. Moreover, we present the results of extensive simulations performed using dynamic simulators and experiments for the multi-robot system to assess the performance of the proposed approach under various scenarios and conditions.

The rest of the thesis is organized as follows: Chapter 2 provides a survey on the literature related to passivity-based control, formation control, human-multi-robot interaction and task allocation. Chapter 3 presents the detailed description of the problem formulations, system model and mathematical background. Chapter 4 presents the detailed description of the Haptic Feedback Synthesis Framework. Chapter 5 presents the results of the performed simulations and experiments and also discusses the performance of the

proposed approach. Chapter 6 concludes the thesis.

A part of this research is submitted for possible publication in the IFAC Mechatronics [16].

Chapter 2

Related Work

The pertinent literature on human-robot interaction, multi-robot systems, task allocation and passivity is covered in this chapter.

2.1 Optimization Based Task Execution

Prior studies on multi-robot systems have investigated various task allocation strategies, such as market-based approaches, auction mechanisms, swarm intelligence algorithms, and task-driven learning techniques [5]. The topic of task allocation has been substantially explored in the literature on multi-robot systems. The framework presented by the authors of [17] begins with a comprehensive task representation that includes requirements, dependencies, priorities, and task characteristics. One of the main contributions of the thesis is the task allocation algorithm, which considers various factors such as task requirements, robot capabilities, energy consumption rates, and the dynamic nature of the environment. The proposed framework also stands out for having an energy awareness. Because the task allocation algorithm considers the energy levels of individual robots, it avoids overwork or the depletion of any one robot's energy. The authors of [10] emphasize how crucial it is to consider task assignment considering robot capability heterogeneity. They emphasize that assigning tasks solely based on robot availability or proximity may lead to sub-optimal performance and under utilization of resources. Instead, the proposed strategy considers the individual capabilities of robots and the requirements of tasks, aiming to maximize the overall system efficiency. So, in our thesis, we present an approach for haptic feedback synthesis of multi-robot multi-task allocation and prioritization system that considers hu-

man instruction and provides feedback based on output, building on our previous work, presented in [17], [10], [18], [19].

Furthermore, the concept of approximation algorithms is explored in [20] as a workable solution to the multi-robot task allocation problem. The goal of approximation algorithms is to identify computationally efficient sub-optimal solutions that strike a good compromise between computational complexity and solution quality. The researchers talk about various approximation methods that have been used in multi-robot task allocation, including randomly assigned algorithms, mathematical relaxations, and greedy heuristics. The authors of [21] suggest a decentralized task allocation algorithm that enables individual robots to decide on task assignments on their own. To ensure effective task allocation, the algorithm considers variables like robot payload capacities, robot proximity to task locations, and task deadlines. Additionally, the proposed approach enables robots to exchange information and collaborate in real-time to optimize task distribution and overall performance.

The challenge of dynamically assigning tasks to a group of robots is discussed in [22], with the goal of maximizing overall performance and resource utilisation. The optimal mass transport theory, a potent mathematical framework that identifies the most effective method of moving mass from one distribution to another, serves as the inspiration for the task allocation algorithm that the authors present. When compared to conventional task allocation methods, the outcomes demonstrate higher task completion rates, decreased resource waste, and enhanced system performance overall. In contrast, the authors of [23] suggest a decentralized, self-organizing task allocation mechanism that draws inspiration from the ideas of swarm intelligence. The algorithm allows individual robots to autonomously and cooperatively make task allocation decisions based on local information and communication with neighboring robots. The self-organizing nature of the approach enables the swarm to adapt to dynamic environments and changing task requirements.

The authors of [24] propose a hybrid force/ velocity/ attitude control system for a robot manipulator, where the key contribution is a passivity-based control approach that ensures both performance and safety, supported by Lyapunov-based stability and passivity analysis. The centralized control framework in our work ensures coordinated movement and task execution among all robots. This contrasts with the more localized control in the authors' approach, where the focus is on a single robot's interaction with its immediate environment. Whereas the authors of [25] proposes a framework, termed Passive Model-Predictive Impedance Control (PMPIC), is designed to maintain compliant interaction behavior while ensuring that the robot system remains passive. This approach minimizes the conservativeness typically associated with passive control strategies by reducing the modification required to the original stiffness profiles, thus preserving the desired interac-

tion behavior. In comparison to our centralized approach which might face challenges in ensuring passivity across all robots, especially in scenarios where human input significantly alters the system dynamics, whereas PMPIC guarantees passivity through its optimized control strategy.

Novel distributed algorithms are presented in [26], allowing a group of robots to cooperatively divide up the tasks among themselves and guarantee that each task is finished by the deadline. Based on local data and interactions with nearby robots, the authors’ decentralised method enables individual robots to autonomously and effectively decide which tasks to assign. Because the algorithms are distributed, they can be scaled, adapted, and robust, which makes them appropriate for use in large-scale multi-robot systems in dynamic environments. Additionally, in [27], a potent method called receding horizon optimization—which permits robots to plan their tasks and paths decentralized—is the foundation of the suggested strategy. The receding horizon approach allows robots to efficiently update their plans based on real-time information and adapt to changes in the environment or task requirements. The approach incorporates constraints such as collision avoidance, task assignment, and resource allocation to ensure safe and efficient robot coordination.

The authors of [28], introduces a novel metric based on frame theory, allowing for a quantitative measure of resilience in multi-robot systems, ensuring that robots can react to failures by changing behaviors or objectives. It considers energy consumption in robot systems, incorporating energy-aware constraints to ensure long-duration autonomy and operational efficiency, particularly in unstructured and unpredictable environments. While both our work and this paper focus on task allocation and control in multi-robot systems, the author’s [28] work places a greater emphasis on resilience and energy constraints. In contrast, our work incorporates haptic feedback as an integral part of the control framework, providing real-time instructions to robots and receiving feedback through haptic devices, which ensures more adaptive interaction with humans.

In [29] the authors discuss the challenge of coordinating robots under strict energy constraints, focusing on the problem of energy-aware rendezvous. It introduces an optimization-based control method to ensure that robots meet at a common location before running out of battery. While the paper focuses on optimizing energy usage for coordination, it lacks human-robot interaction or haptic feedback integration, both central aspects of our work. Our approach extends beyond energy-constrained coordination by incorporating haptic feedback and task prioritization. However, the decentralized nature of the author’s work makes it more scalable in scenarios with limited communication compared to our more centralized framework, which may face challenges with scalability.

A decentralized submodular optimization algorithm for task allocation is proposed in [30]. It ensures robots can allocate tasks independently, emphasizing scalability and adaptability. While this work shares similarities in task allocation, our research extends the framework by integrating human-robot interaction and haptic feedback, ensuring real-time adaptability. Our centralized framework offers better coordination in complex scenarios but may face limitations in large-scale, highly dynamic environments compared to the decentralized submodular method that excels in flexibility. While the authors of [31], focuses on task allocation in persistent tasks with an energy-aware framework. It ensures the longevity of robots by optimizing energy usage while allocating tasks dynamically. Compared to this, our work incorporates haptic feedback in multi-robot systems and optimizes both energy and task allocation in real time. Although the energy-aware aspect aligns with our task prioritization, our research adds a layer of complexity by involving human operators and haptic feedback, allowing for more interactive control in dynamic environments. However, the authors’ focus on energy management might offer better longevity in continuous operations.

A framework for multi-robot coalition formation, which involves partitioning robots into coalitions that are dynamically assigned to various tasks is presented in [32]. The work builds on multi-agent systems and aims to bridge the gap between coalition formation in multi-agent and multi-robot systems. The algorithm proposed in this paper deals with communication constraints, fault tolerance, and resource distribution challenges specific to real-world robotic environments. Our centralized task allocation framework provides real-time adaptability with haptic feedback, which is not considered in the authors’ work. Moreover, Our focus on haptic feedback and passivity ensures safer human-robot collaboration. The authors of [33] discusses decentralized receding horizon control (RHC) for large-scale systems, where each subsystem computes its own control inputs based on its states and the states of its neighbors. The decentralized approach ensures that systems can operate independently but still achieve overall system objectives through local interactions. The decentralized RHC approach might be more suitable for large, distributed systems where communication between robots is limited. Our centralized approach might struggle with scalability in comparison, especially in environments with a high degree of robot autonomy.

In [34] the authors present a decentralized approach for maintaining robot formation by relying on local communication between neighboring robots. This enables the system to scale efficiently and adapt to dynamic environments, such as avoiding obstacles, through decentralized decision-making and optimization. Compared to this paper’s distributed approach, our work leverages both centralized and decentralized methods. The decentralized formation control in our work provides the flexibility and scalability seen in this paper,

but our centralized system offers precise coordination in multi-task scenarios with human oversight. However, the paper’s decentralized method excels in large-scale dynamic environments, where our centralized framework may face scalability challenges.

2.2 Human-Multi-Robot Teleoperation

The field of human-multi-robot teleoperation has garnered significant attention due to its potential for enabling human operators to control and interact with multiple robots simultaneously in complex and dynamic environments. Recent advancements in teleoperation frameworks have focused on ensuring stability and passivity in the system, allowing human operators to provide high-level guidance while maintaining safe and predictable robot behavior, even in the presence of time delays or external disturbances. In [35] the authors present a passivity-based control framework to address stability issues in teleoperation systems with time delays, which is critical in environments where robots operate at a distance. This is relevant to our work on human-multi-robot teleoperation as it ensures that haptic feedback remains stable even with time delays. The paper’s emphasis on maintaining stability through passivity aligns with our work, where human operators control multiple robots using real-time haptic feedback. However, our system extends beyond single-Degree Of Freedom control and involves complex multi-task, multi-robot scenarios, which adds more complexity than the single-robot focus of this thesis.

Haptic rendering plays a pivotal role in facilitating realistic and stable interactions between humans and robotic systems. Early foundational work by Zilles and Salisbury introduced the constraint-based ‘god-object’ method, which computes interaction forces in real-time to create convincing haptic experiences with virtual objects [36]. This method established the groundwork for haptic systems capable of accurately simulating surface properties such as compliance and friction. Furthermore, Adams and Hannaford extended these concepts by proposing a virtual coupling network to ensure stability in haptic simulations, even in the presence of complex virtual environments [37]. These contributions highlight the critical importance of stability and performance in haptic feedback systems, which are central to the framework proposed in this thesis for human-multi-robot interaction.

In [38] the authors review various haptic feedback techniques, including force and tactile feedback. This paper is directly relevant to the feedback mechanisms we use in our system to inform human operators of the robot’s state. The paper presents multiple approaches to haptic feedback, including sensory substitution methods (visual and auditory

feedback) when direct haptic feedback is not feasible. These strategies could enhance if implemented in our system, especially in scenarios where full tactile feedback may be difficult to implement. The methods discussed in the paper are primarily focused on single-task, single-robot systems (e.g., surgical robots), whereas our system is designed for multi-task, multi-robot environments. Integrating these approaches might require additional modifications to handle the complexity of our system.

In [13] and [14] the authors introduce optimization-based frameworks to ensure stability, safety, and passivity in teleoperation systems. Both works employ passivity-based control, using filters like the SSP (stability, safety, passivity) filter to minimally alter human inputs and maintain system stability. The approach of [13] filters human inputs to maintain system safety through control barrier functions (CBFs) and passivity with real-time implementation, focusing on experimental results with mobile robots. Both papers' techniques are closely related to our work, as both aim to ensure stability and passivity in human-multi-robot systems. However, our work emphasizes task allocation and prioritization with haptic feedback, incorporating both centralized and decentralized frameworks for different dynamic systems. Unlike the Stability-safety-passivity (SSP) filter, which focuses on stability and safety, our approach integrates haptic feedback and handles multi-task execution with human input. While the SSP framework excels in ensuring safety during teleoperation, our approach offers a more comprehensive task allocation mechanism and direct human-robot interaction, but may face challenges with scalability and safety integration.

The authors of [39] provide a thorough framework for cooperative assembly planning that combines human and robotic capabilities to successfully handle assembly challenges encountered in the real world. The framework is structured into three layers: team-level planning, agent-level task allocation, and real-time execution. The system's ability to generate nominal plans while accounting for unforeseen events and faults is noteworthy and is essential for its practical application in industrial settings. The crucial problem of task allocation in multi-robot systems operating in unpredictable environments is discussed by the authors in [40], with a focus on the field of space exploration. The study, which aims to balance the trade-offs between robot commitment and coordination, presents an empirical investigation into four different task allocation strategies. Notably, the study reveals that no single task allocation strategy consistently outperforms others across all scenarios. Instead, the effectiveness of task allocation strategies is shown to vary depending on the level of noise and uncertainty present in the environment. Overall, the approach in [39] performs better than our approach of human-robot interaction if there are multiple levels of human involvement, but our optimization algorithm and task allocation approach is much faster than the approach described in [40].

This authors [41] introduce a multi-robot teleoperation strategy where a human operator controls multiple robots to perform dexterous and complex tasks in real-time. The system integrates haptic feedback, enabling the operator to sense interaction forces and adjust the task accordingly. Compared to our work, this approach has strong parallels with the use of haptic feedback for real-time control. While the authors of [42], discusses ensuring stability in haptic teleoperation systems, particularly when dealing with nonlinear virtual environments. Like our system, the authors also emphasize passivity to maintain stability in human-robot interactions. The primary distinction is that their system focuses on stabilizing nonlinear virtual environments, whereas our work deals with multiple physical robots performing real tasks. The advantage of our system is its applicability to real-world, multi-robot scenarios, while the authors' work provide more specific solutions for haptic systems in nonlinear virtual settings, which might not be as broadly applicable in real-world robot teleoperation tasks.

The authors of the paper [43] presents a system that uses two human operators (dual-master) controlling a single slave robot to perform bilateral teleoperation. It's designed for scenarios where complex tasks require simultaneous input from two operators. The relevance to our work lies in the use of haptic feedback for control. The disadvantage of this system compared to ours is that it requires multiple operators for complex tasks, whereas our system focuses on a single operator controlling multiple robots. However, this paper provides insight into how multiple operators can be utilized for more complex task coordination. While the authors of [44] presents a comprehensive human-robot interaction control strategy, including force feedback and impedance controllers, allowing the operator to control robots effectively in various scenarios. The teleoperation system incorporates force feedback to improve interaction between the human and the robots, ensuring smooth movement and control. Compared to our work, this paper's contribution is more focused on scenario-specific control. The advantage of our work is its flexibility in handling a broader range of tasks, as well as the incorporation of haptic feedback to optimize task allocation. However, this paper offers a deeper exploration into the control strategies that could further improve the haptic feedback aspect of our system.

A framework for human-in-the-loop control of multi-robot systems, focusing on passivity analysis and using VR interfaces to improve human performance and system stability is presented in [45]. The relevance to our work lies in the use of distributed control strategies to ensure stable multi-robot operation, similar to our focus on task allocation and execution. However, this work heavily emphasizes managing passivity shortages in the human operator, which contrasts with our approach that prioritizes human-robot collaboration and distributed decision-making over strict passivity requirements. While [46], addresses a leader-follower formation control system, where a single human operator manipulates mul-

multiple robots cooperatively, using haptic feedback to enhance interaction. This is relevant to our research on human-multi-robot teleoperation, as it explores how feedback can improve the human’s control over a team of robots. While it offers insights into effective human control, especially through the use of feedback, its focus on object manipulation differs from our application, which deals more with distributed optimization for autonomous task allocation.

The authors of the paper [47] analyze the passivity of the human arm in teleoperation systems, particularly in the position-force domain, to ensure safe interaction in telerobotics. It is relevant to our work in that it explores the critical role of human passivity in teleoperation, aligning with the safety concerns in human-robot interaction. However, the study’s primary focus on single-master/single-slave teleoperation limits its applicability to our multi-robot teleoperation framework, where multiple robots and distributed control are central to system functionality. While the authors of the paper [48] propose a robust control law that ensures both safety and passivity in human-robot interactions by imposing velocity constraints and restoring forces to keep the robot within a safe operating region. Its focus on ensuring safe interaction and passivity in physical human-robot collaborations is closely related to the safety aspect of our research. Though, its emphasis on velocity control and physical constraints is somewhat different from our focus on distributed multi-robot control for task allocation and execution, but the human operator plays a more supervisory role rather than direct physical control.

A decentralized passivity-based control for teleoperation of UAV groups, focusing on maintaining connectivity and obstacle avoidance is presented in [49]. It aligns with our work’s emphasis on safe multi-robot control and interaction through passivity. However, its focus on UAV-specific dynamics and decentralized control differs from our system’s exploration of centralized frameworks under certain conditions. While the authors of the paper [50] explore passivity-based methods for coordinating robot teams across long distances with communication delays. It highlights handling human-swarm interaction and time delays, which is relevant to our focus on robust, scalable multi-robot teleoperation. However, this work emphasizes delay compensation rather than direct haptic feedback or human-in-the-loop control in real-time.

A decentralized control approach that ensures passivity in multi-robot systems under communication delays using control barrier functions is discussed in [51]. It directly relates to our research on maintaining stability and safety in human-robot systems. The key difference is the strong focus on decentralized control, whereas our work also explores centralized strategies for certain dynamic systems. While a passivity-based teleoperation system for heterogeneous robot teams, allowing flexibility in formation and interaction through decentralized control is presented in [52]. It closely relates to our research on

safe human-robot interaction and passivity but differs in focusing more on flexible, time-varying topology and decentralized control, while our system considers centralized control in specific cases.

A novel haptic shared autonomy framework for teleoperating quadrotor UAVs, focusing on safety and stability using control barrier functions and finite L2-gain methods is proposed in [53]. It aligns with our work’s emphasis on passivity and haptic feedback for safe human-robot interaction. However, it primarily addresses UAV teleoperation rather than broader multi-robot systems, which limits its applicability to our generalized framework.

A framework is proposed in the work of Zhang [54] for adaptive scheduling in human-multi-robot collaboration, leveraging real-time performance indices like fatigue to dynamically adjust task allocation. This approach is relevant to human-in-the-loop systems and complements our work by focusing on dynamic adaptability in collaborative tasks. However, their framework is tailored to manufacturing applications and lacks emphasis on haptic feedback or real-time interaction critical for teleoperation systems. My work expands applicability by incorporating haptic feedback and task prioritization, enabling broader real-world applications beyond assembly tasks. While the author of [55], design a modular, multimodal interface for controlling multi-robot teams in hazardous environments, emphasizing safety and adaptability through real-time feedback. This aligns with the focus on robust human-multi-robot interaction but is limited to interface design without optimization-based control or stability considerations. Our framework integrates optimization and passivity principles to ensure both stability and adaptability in teleoperation, extending the utility of human-robot systems to dynamic environments requiring precise control.

Compared to many other task allocation approaches, which assign a single task to each robot at a time, our work investigates a scenario in which individual robots carry out multiple tasks at different priorities simultaneously. This problem is formulated as a quadratic programming, which is resolved at every time step. This formulation computationally efficient as its complexity grows exponentially with the number of tasks and robots [56] than other algorithms that give robots multiple tasks [5]. We propose a multi-robot, multi-task system that uses an optimization algorithm and facilitates human-robot interaction. When making decisions, human-robot interaction can also benefit the system because, in contrast to [20], [21], [22], [23], [26], [27], a human operator can modify instructions based on the best input from the system.

Moreover, task allocation and control in multi-robot systems have increasingly focused on real-time performance and adaptability. The framework in [26] introduces a real-time optimization-based control system that manages multi-robot formations, achieving task

coordination through continuous feedback loops. Similarly, [27] demonstrates a method for managing interactions between robots and human operators through haptic feedback, ensuring smooth coordination during task transitions. Our work builds on these studies by incorporating passivity-based haptic feedback synthesis into the centralized task allocation framework. This allows human operators to interact seamlessly with the multi-robot system, influencing task allocation and robot behavior in real time.

In summary, while previous work has made significant strides in decentralized control, task allocation, and passivity-based human robot interaction, our approach distinguishes itself by combining these elements into a single centralized framework. The integration of haptic feedback, formation control, and passivity-based stability ensures that the multi-robot system operates efficiently and safely, with real-time adaptability to human inputs. This contribution advances the state of the art by addressing the complex requirements of multi-task, multi-robot systems with human interaction, offering a robust solution for applications in dynamic environments.

Chapter 3

Problem Formulation

We consider a group of N robots that are deployed in an environment to perform M tasks. One of the tasks may consist in following the instructions of a human operator. Robot posses features that determine their capabilities of executing different tasks. Based on this capabilities, we assume that a priority stack for the different tasks to execute is assigned to each robot. To perform a prioritized stack of tasks, an optimization algorithm is proposed in this thesis.

3.1 System Model and Mathematical Background

The robots are modelled by a control affine dynamical system

$$\dot{x}_i = f(x_i) + g(x_i)u_i, \quad (3.1)$$

where $x_i \in \mathbb{R}^{n_i}$ and $u_i \in \mathbb{R}^{m_i}$ are the state and the input of robot i . In general, our approach works for control affine dynamical systems, so that the optimization-based controller is convex.

We modelled the robots using single-integrator dynamic system, $\dot{x}_i = u_i$ where $i = 1, \dots, N$, $x_i \in \mathbb{R}^2$ and $u_i \in \mathbb{R}^2$ represents the position of the robots in a two-dimensional workspace. While this formulation assumes a 2D domain, it can be extended to higher-dimensional spaces ($\mathbb{R}^n$) depending on the specific application requirements. Then, we

modelled the robots using double-integrator dynamic system,

$$\dot{x}_i = \overbrace{\begin{bmatrix} 0_{2 \times 2} & I_{2 \times 2} \\ 0_{2 \times 2} & 0_{2 \times 2} \end{bmatrix} x_i + \begin{bmatrix} 0_{2 \times 2} \\ I_{2 \times 2} \end{bmatrix} u_i}^{f(x_i, u_i)}$$

A
 B

where $i = 1, \dots, N$, $u_i \in \mathbb{R}^2$ Acceleration and

$$x_i = \begin{bmatrix} x_{i,1} \\ x_{i,2} \end{bmatrix} \in \mathbb{R}^4$$

Here position $x_{i,1} \in \mathbb{R}^2$ and velocity $x_{i,2} \in \mathbb{R}^2$

The optimization-based control framework builds on the concepts of constraint-driven coordinated control and energy-aware task allocation in multi-robot systems. Notomista and Egerstedt [18] introduced a method where multi-robot tasks are encoded as constraints within an optimization problem. This constraint-driven approach allows for robust and adaptable control strategies, ensuring task execution while maintaining long-term survivability, even under changing environmental conditions. The use of optimization ensures that the robots' actions minimally influence their overall task performance, striking a balance between energy consumption and task execution.

Building on this foundation, Notomista [10] extended the framework to account for heterogeneous robot capabilities, formulating a task allocation strategy that optimizes resource usage while respecting individual robot constraints. By introducing slack variables and prioritization mechanisms, the approach enables robots to handle multiple tasks dynamically and flexibly, even under resource limitations.

These works provide the basis for the optimization program presented in this thesis, which integrates these principles into a unified framework for multi-robot task execution with human interaction and haptic feedback.

In this thesis, we start from the constraint-driven approach for task execution [18], consisting in solving the following minimum-energy optimization program, where the constraint function encodes the task execution [10]:

$$\begin{aligned} \min_{u_i} \quad & \|u_i\|^2 \\ \text{subject to} \quad & L_f T_j(x_i) + L_g T_j(x_i) u_i \leq -\gamma_j T_j(x_i), \\ & j \in \{1, \dots, M\}, \end{aligned} \tag{3.2}$$

where T_j denotes that function encoding task j , $L_f T_j(x_i)$ and $L_g T_j(x_i)$ are the Lie derivatives of T_j in the directions of the vector fields f and g , respectively, and $\gamma_j > 0$. We refer to [18] for an introduction to the constraint-based task execution framework we used in this thesis.

This formulation can be extended to the task allocation and execution for multiple robots that perform multiple tasks. Each task constraint is supplemented with a slack variable and relative constraints on the slack variable are introduced:

$$\begin{aligned} \min_{u_i, \delta_i} \quad & \|u_i\|^2 + K_\delta \|\delta_i\|^2 \\ \text{subject to} \quad & L_f T_1(x_i) + L_g T_1(x_i) u_i \leq -\gamma_1 T_1(x_i) + \delta_{i,1} \\ & \vdots \\ & L_f T_n(x_i) + L_g T_n(x_i) u_i \leq -\gamma_n T_n(x_i) + \delta_{i,n} \\ & \delta_{i,j} \leq \frac{1}{k} \delta_{i,k}, \end{aligned} \tag{3.3}$$

where $\delta_{i,j}$ is the slack variable for the constraint encoding the execution of task j by robot i , and $K_\delta > 0$. The constraint $\delta_{i,j} \leq \frac{1}{k} \delta_{i,k}$, $k \gg 1$, represents the relative priorities of the tasks assigned to robot i . Here, k (also referred to as κ) is a ratio between the slack variables of two tasks. A higher k implies a greater disparity between the corresponding slack variables, signifying a stronger prioritization of one task over the other. This allows the optimization framework to control the relative effectiveness of task execution based on predefined priorities. In this case, the constraint specifies that task j has to be executed with higher priority with respect to task k . This way, the slack variables $\delta = [\delta_1, \dots, \delta_M]^T$ not only make such a program feasible, but also enable individual robots to prioritize tasks.

If a human operator interacts with the robotic system via a human input u_H , our objective is to synthesis the haptic feedback Δu_H^* , which will inform the human about the current performance of the robots. The optimization algorithm (3.3) modifies to:

$$\begin{aligned} \min_{u_i, \delta_i, \Delta u_H} \quad & \|u_i\|^2 + K_\delta \|\delta_i\|^2 + K_\Delta \|\Delta u_H\|^2 \\ \text{subject to} \quad & L_f T_1(x_i) + L_g T_1(x_i) u_i \leq -\gamma_1 T_1(x_i) + \delta_{i,1} \\ & \vdots \\ & L_f T_n(x_i) + L_g T_n(x_i) u_i \leq -\gamma_n T_n(x_i) + \delta_{i,n} \\ & \delta_{i,j} \leq \frac{1}{k} \delta_{i,k} \\ & u_i = f_{H_i}(x, u_H + \Delta u_H), \end{aligned} \tag{3.4}$$

where $u_H \in \mathbb{R}^p$ is human input and $\Delta u_H \in \mathbb{R}^p$ is the feedback from the robots to the human, accounting for the current allocation and execution of tasks. $K_\delta > 0$, $K_\Delta > 0$, and f_{H_i} is a function of x_i , u_H and Δu_H . In the following section, we provide an example of such

a function used to specify the interaction between a human operator and a team of robots. Currently, Δu_H is unconstrained, meaning it can take any value in $\mathbb{R}^p$, allowing flexibility in the system’s response. However, for safety considerations, constraints on Δu_H , such as minimum and maximum bounds, can be incorporated to ensure that the feedback remains within safe limits for the human operator.

Remark 1 Δu_H , the haptic feedback, has the same dimension of u_H , the vector representing the human input. Therefore, the rendering of this feedback to the human operator is seamless. As an example, suppose u_H is a velocity sent by the human to a robot. In this case, the feedback Δu_H is also a velocity of the same dimension, enabling the feedback to be rendered directly in the same unit and modality, such as a force on a haptic device, in a typical teleoperation architecture [57]. Maintaining the same dimension ensures consistency in the feedback mechanism, allowing the haptic device to display feedback in a way that aligns with the operator’s input. If Δu_H were not of the same dimension, the feedback modality would need to change. For instance, visual feedback instead of haptic feedback—resulting in a different user experience and potentially limiting the applicability to certain tasks. Also, representing velocity as a force on a haptic device in this context involves mapping the velocity feedback to an equivalent force output, which provides the operator with an intuitive sense of the robot’s state and the current task dynamics.

3.2 Passive Teleoperation

In human-robot interaction, especially in teleoperation systems, ensuring stability is paramount due to the unpredictable nature of human input. Passivity is a critical concept in control systems, particularly when dealing with dynamic environments and external disturbances. Passivity guarantees that the system does not generate excess energy, preventing instability. By maintaining passivity, the system ensures that it can safely respond to human inputs, adapt to external disturbances, and maintain stable interaction, even in the presence of delays or uncertainties. Passivity also provides a robust framework for ensuring that interactions within the system remain predictable and safe, even when the system is subjected to unknown forces or environmental changes [49], [52]. This makes passivity critical in applications where human operators directly influence the robot’s behavior, such as in teleoperated multi-robot systems.

A system is considered passive if the energy supplied to it does not exceed the energy stored plus the energy dissipated. Mathematically, this can be expressed in terms of the input u and output y as follows:

$$\int_0^T u(t)^T y(t) dt \geq S(x(T)) - S(x(0)) \quad (3.5)$$

where $S(x)$ is the storage function, representing the stored energy in the system, and $x(t)$ is the state of the system at time t . The inequality implies that the energy supplied by the input $u(t)$ is greater than or equal to the increase in the stored energy plus any energy dissipated by the system. Passivity can be further classified into different types, such as:

- **Lossless Passivity:** If all the supplied energy is stored, i.e., no energy is dissipated.
- **Strict Passivity:** If the system dissipates some of the energy, ensuring that the stored energy cannot increase indefinitely [58].

In the context of control systems, especially those involving human-robot interaction, ensuring passivity is critical for maintaining system stability. When a human interacts with a robot, their input can vary unpredictably. A passivity-based control framework ensures that regardless of these variations, the system remains stable by not allowing it to generate excess energy that could lead to instability.

Chapter 4

Haptic Feedback Synthesis Framework

In this chapter we will introduce the formulation of the tasks as constraint of a optimization problem and the control framework for single-integrator dynamic system and double-integrator dynamic system. In case of single-integrator dynamic system, the control framework is decentralized. On the other hand for double-integrator dynamic system, the control framework is centralized.

4.1 Single-Integrator Dynamic System

We modelled the robots using single-integrator dynamic system, $\dot{x}_i = u_i$ where $i = 1, \dots, N$, $x_i \in \mathbb{R}^2$ and $u_i \in \mathbb{R}^2$. The system has direct control over position using velocity as control input.

4.1.1 Model of the Tasks

In this subsection we will introduce the formulation of the tasks as constraint of a optimization problem for single-integrator dynamic system.

Go-to-Goal Task

Driving to specific sites within the domain constitutes Task 1, as detailed in [11]. This is the Go-to-Goal Task. The construction of the task 1 in space can be encoded by the following cost, as has been done in [5], [59], [60].

$$T_1(x) = \sum_{i=1}^N \frac{1}{2} \|x_i - p_i\|^2, \quad (4.1)$$

where $x_i \in \mathbb{R}^2$ represents the state (position) of robot i and $p_i \in \mathbb{R}^2$ represents the target position for robot i . Both x_i and p_i are defined in a two-dimensional workspace. Here, $i = 1, \dots, N$, the total number of robots in the system. The formulation ensures that each robot is assigned a unique target p_i , avoiding ambiguity in the notation. For this example, $N = 4$, indicating a team of four robots operating in a two-dimensional domain.

The following optimization problem determines the execution of task 1 by each robot i :

$$\begin{aligned} \min_{u_i} \quad & \|u_i\|^2 \\ \text{subject to} \quad & 2(x_i - p_i)^T u_i \leq \gamma_1(-\|x_i - p_i\|^2). \end{aligned} \quad (4.2)$$

Objective ($\min_{u_i} \|u_i\|^2$): This term minimizes the control effort u_i applied by robot i . The goal is to ensure that the robot reaches its target position p_i in an energy-efficient manner.

Constraint ($2(x_i - p_i)^T u_i \leq \gamma_1(-\|x_i - p_i\|^2)$): This inequality encodes the task execution constraint. The left-hand side, $2(x_i - p_i)^T u_i$, represents the projection of the control input u_i along the vector pointing from the robot's current position x_i to its target position p_i . This ensures that the control action is directed toward reducing the distance to the goal. The right-hand side, $\gamma_1(-\|x_i - p_i\|^2)$, is a function that determines the allowable rate of distance reduction based on the current distance $\|x_i - p_i\|^2$. The function γ_1 ensures that the robot moves closer to its target while maintaining stability.

This formulation guarantees that the robot minimizes its energy usage while respecting the task constraints that guide it toward its goal. By balancing these components, the optimization problem ensures efficient and stable task execution for each robot.

Formation Constraint

Robots are required to arrange themselves into a predefined shape in formation control scenarios, which are defined by the distances between each robot. We can express this

problem as the minimization of a cost function by representing the formation error as $T_2(x)$:

$$T_2(x) = \sum_{i=1}^n \sum_{j \in \mathcal{N}_i} \frac{1}{2} (\|x_i - x_j\| - d_{ij})^2, \quad (4.3)$$

where $\mathcal{N}_i$ specifies the set of robots with which robot i maintains a predefined distance. In this case, the desired distance between robots i and j is denoted by d_{ij} . These relative distances, d_{ij} , define the desired formation, from which the robot configuration currently operates. $T_2(x)$ quantifies the distance from the desired formation. The robots have precisely formed the desired shape when $T_2(x) = 0$.

The gradient of $T_{2,i}$ with respect to the position of robot i , required to evaluate the Lie derivative of the task function, are calculated as:

$$\frac{\partial T_{2,i}}{\partial x_i} = \sum_{j \in \mathcal{N}_i} \underbrace{\frac{\|x_i - x_j\| - d_{ij}}{\|x_i - x_j\|}}_{w_{ij}} (x_i - x_j)^T. \quad (4.4)$$

The following is an explanation of this equation: When robots i and j are closer than d_{ij} , the weight w_{ij} is negative, which creates a repulsive force between the robots. In contrast, if d_{ij} is smaller than the distance between the robots, then w_{ij} is positive, which creates an attracting force between the robots [18].

The following optimization problem results in the execution of task 2:

$$\begin{aligned} \min_{u_i} \quad & \|u_i\|^2 \\ \text{subject to} \quad & 2 \frac{\partial T_{2,i}}{\partial x_i} u_i \leq -\gamma_2 T_{2,i}. \end{aligned} \quad (4.5)$$

The robots form the desired shape by either attracting or repelling each other based on the distance to their neighbors. If two robots are too close, a repulsive force pushes them apart, and if they are too far, an attractive force pulls them together.

Human-Robot Interaction Constraint

The optimization problem encoding the execution of the human-robot interaction constraint used in the simulations is as follows:

$$\begin{aligned} \min_{u_i, \Delta u_H} \quad & \|u_i\|^2 + K_\Delta \|\Delta u_H\|^2 \\ \text{subject to} \quad & u_i = u_H + \Delta u_H. \end{aligned} \quad (4.6)$$

Recall that $u_H \in \mathbb{R}^2$ is human input and $\Delta u_H \in \mathbb{R}^2$ is feedback from the multi-robot system to the human operator. If we consider a velocity of a given dimension, u_H , then Δu_H is the haptic feedback velocity of that same dimension.

Objective ($\min_{u_i, \Delta u_H} \|u_i\|^2 + K_\Delta \|\Delta u_H\|^2$): This objective has two terms: $\|u_i\|^2$, minimizes the control effort for the robots to execute their tasks efficiently. $K_\Delta \|\Delta u_H\|^2$, regularizes the feedback Δu_H from the robots to the human operator, ensuring that the feedback remains smooth and within practical bounds. The coefficient $K_\Delta > 0$ determines the relative importance of feedback minimization.

Constraint ($u_i = u_H + \Delta u_H$): This constraint couples the robot's control input u_i with the human input u_H and the feedback Δu_H from the robots. The feedback Δu_H serves as a force-like interaction term, guiding the human operator toward the direction in which the robots are executing their tasks, akin to a virtual spring.

Human Operator Control: In this framework, the human operator provides a single input u_H that influences all i robots. Each robot's input u_i is adjusted by adding the feedback Δu_H , which aligns the human's commands with the robots' task execution.

Interpretation: The feedback Δu_H can be interpreted as a force feedback mechanism, where the human operator feels a spring-like force pulling them toward the direction of task execution. This ensures that the human remains aware of the multi-robot system's overall state and performance, improving interaction intuitiveness and task alignment.

This formulation enables a seamless human-robot interaction loop, where the human input drives task execution while feedback from the robots aids the operator in dynamically adjusting their commands for optimal performance.

Via the constraint $u_i = u_H + \Delta u_H$, the multi-robot system receives instructions from a human operator. The latter is informed about the execution of the multiple tasks by the robots through the haptic feedback Δu_h^* . Although there a variety of human input with different dimensions, in the simulations for this thesis, we consider a velocity input, i.e., $u_H \in \mathbb{R}^2$ is the desired velocity of robot i . In particular, the human operator attempts to drive robot 4 in a different direction by providing it with a velocity (u_H) as an input. As a result, the human operator experiences a force as feedback, indicating that the task can be difficult to complete.

The feedback Δu_H reflects the system's response to the human's input, and if the robots face difficulties in tracking the input precisely, the human feels this as feedback via a haptic device. The constraint ensures that this interaction is executed seamlessly, with the robots continually adjusting to follow the input while reporting back any issues via the feedback term.

Coverage Control Constraint

A coverage control algorithm is used to efficiently explore and cover the given environment. Coverage control aims to distribute the robots in to achieve adequate coverage of the environment while minimizing overlap and gaps [61] [62]. The algorithm relies on the concept of Voronoi diagrams to partition the environment into regions, assigning each region to a robot. The Voronoi cells represent areas that are closer to a specific robot than to any other robot. The coverage control technique provided in [11] is implemented in Task T3.

For Task T3, the robots' moving domain is divided into N Voronoi cells that match to the N robots. Only its Voronoi cell V_i is being watched over by robot i . The cost to minimize to carry out this monitoring activity is given by [63], which is presented.

$$T_3(x) = \sum_{i=1}^N \frac{1}{2} \|x_i - G_i(x)\|^2 \quad (4.7)$$

where G_i denotes the centroid of the Voronoi cell V_i . $G_i \in \mathbb{R}^2$, $i = 1, \dots, N$ are the N locations on a plane at a specified altitude and $x_i \in \mathbb{R}^2$, $i = 1, 2$ is the robot state.

The Voronoi diagram of the robot is calculated based on the current positions of the robots. Each robot is associated with a Voronoi cell, which defines the area it is responsible for covering. The centroid of each Voronoi cell is computed. The centroid represents the target position within the cell where the robot should move to ensure better coverage. The computed velocity vectors are used to move the robots over time. It updates the positions of all robots according to their respective velocity vectors.

The environment is represented by a set of boundaries defining the spatial limits in which the robot operates. The boundaries can be defined as a closed rectangular or square path, where each edge represents a wall or obstacle. The environment property stores the coordinates of the boundary vertices. The environment can be either static or dynamic, depending on the application. In a static environment, the boundary does not change over time, and the robots focuses on exploring and covering the fixed area. In contrast, a dynamic environment may involve moving obstacles or changing boundaries, requiring the robots to adapt its coverage strategy in response to the changes.

The following optimization problem determines the execution of task 3 by each robot i :

$$\begin{aligned} \min_{u_i} \quad & \|u_i\|^2 \\ \text{subject to} \quad & 2(x_i - G_i)^T u_i \leq \gamma_3(-\|x_i - G_i\|^2). \end{aligned} \quad (4.8)$$

4.1.2 Optimization-based Decentralized Control Framework

This optimization-based decentralized control framework is based on 3 tasks. They are as follows: 1) The go to goal task; 2) The formation control task; and 3) Human-Robot interaction. We are considering total number of 4 robots for the system. Each of these robots have different set of priorities in terms of executing the tasks. The resulting optimization algorithm is a standard quadratic program:

$$\begin{aligned} \min_{v_i=[u_i^T, \delta_i^T, \Delta u_H^T]^T} \quad & v_i^T H v_i + l^T v_i \\ \text{subject to} \quad & A_{QP} v_i \leq b_{QP}, \\ & C_{QP} v_i = d_{QP}, \end{aligned} \quad (4.9)$$

where $H \in \mathbb{R}^{6 \times 6}$, $A_{QP} \in \mathbb{R}^{3 \times 6}$, and $C_{QP} \in \mathbb{R}^{2 \times 6}$, and $l \in \mathbb{R}^{6 \times 1}$, $b_{QP} \in \mathbb{R}^{3 \times 1}$, $d_{QP} \in \mathbb{R}^{2 \times 1}$, and $v_i \in \mathbb{R}^{1 \times 6}$. For example, in the case of robot 4, have the following expressions:

$$\begin{aligned} H &= \begin{bmatrix} I & 0 & 0 \\ 0 & K_\delta I & 0 \\ 0 & 0 & K_\Delta I \end{bmatrix}, \quad A_{QP} = \begin{bmatrix} \frac{\partial T_2}{\partial x_4} & -1 & 0 & 0 & 0 \\ 2(x_4 - p_4)^T & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & -\frac{1}{K} & 0 & 0 \end{bmatrix}, \\ C_{QP} &= [I \quad 0 \quad -I], \quad l = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad b_{QP} = \begin{bmatrix} -\gamma_1 T_2 \\ \gamma_2(-\|x_4 - p_4\|^2) \\ 0 \end{bmatrix}, \\ d_{QP} &= u_H, \quad v_i = \begin{bmatrix} u_4 \\ \delta_{4,1} \\ \delta_{4,2} \\ \Delta u_H \end{bmatrix}. \end{aligned} \quad (4.10)$$

Inequality Constraint ($A_{QP} v_i \leq b_{QP}$) : Ensures task feasibility for each robot.

$$\begin{aligned} A_{QP} v_i &\leq b_{QP} \\ \begin{bmatrix} \frac{\partial T_2}{\partial x_4} & -1 & 0 & 0 & 0 \\ 2(x_4 - p_4)^T & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & -\frac{1}{K} & 0 & 0 \end{bmatrix} \begin{bmatrix} u_4 \\ \delta_{4,1} \\ \delta_{4,2} \end{bmatrix} &\leq \begin{bmatrix} -\gamma_1 T_2 \\ \gamma_2(-\|x_4 - p_4\|^2) \\ 0 \end{bmatrix} \end{aligned} \quad (4.11)$$

Equality Constraint ($C_{QP}v_i = d_{QP}$) : Enforces equality for task execution.

$$C_{QP}v_i = d_{QP}$$

$$\begin{bmatrix} I & 0 & -I \end{bmatrix} \begin{bmatrix} u_i \\ \delta_{i,1} \\ \delta_{i,2} \\ \Delta u_H \end{bmatrix} = u_H \quad (4.12)$$

Objective Function ($v_i^T H v_i + l^T v_i$): The control input u_i is minimized to reduce the energy expenditure of the robot. This term is quadratic, reflecting the minimization of $\|u_i\|^2$. Slack variables δ_i allow task constraints to remain feasible, and Δu_H regularizes the feedback from the robots to the human operator. These quadratic and linear terms form the basis of the quadratic program's objective function, where H is a positive definite matrix encoding weights for the control effort and feedback terms.

Inequality Constraints ($A_{QP}v_i \leq b_{QP}$): These constraints encode task requirements, such as maintaining inter-robot distances (formation control) or ensuring progress toward a goal (go-to-goal task). For example, constraints derived from control barrier functions (CBFs) for safety or passivity can be expressed as linear inequalities.

Equality Constraints ($C_{QP}v_i = d_{QP}$): These constraints enforce consistency in the human-robot interaction. For instance, they couple the control inputs u_i and the human input u_H via the feedback Δu_H . Equality constraints also ensure task priorities are distributed correctly among the robots.

By combining these components, the optimization problem naturally takes the form of a standard QP, which is computationally efficient to solve and scales well for decentralized multi-robot systems. This framework enables each robot to dynamically balance task execution while adapting to human inputs and maintaining system safety.

Another optimization-based decentralized control framework is based on 2 tasks. They are as follows: 1) The go to goal task; and 2) The coverage control task. We are considering 2 robots for the system. Each of these robots have different set of priorities in terms of executing the tasks. The resulting optimization algorithm is a standard quadratic program:

$$\begin{aligned} \min_{v_i=[u_i^T, \delta_i^T, \Delta u_H^T]^T} \quad & v_i^T H v_i + l^T v_i \\ \text{subject to} \quad & A v_i \leq b, \end{aligned} \quad (4.13)$$

where $H \in \mathbb{R}^{4 \times 4}$, $A_{QP} \in \mathbb{R}^{3 \times 4}$, and $l \in \mathbb{R}^{4 \times 1}$, $b_{QP} \in \mathbb{R}^{3 \times 1}$, and $v_i \in \mathbb{R}^{1 \times 2}$. For example, in

the case of robot 1, have the following expressions:

$$\begin{aligned}
H &= \begin{bmatrix} I & 0 \\ 0 & K_\delta I \end{bmatrix}, \\
A_{\text{QP}} &= \begin{bmatrix} 2(x_1 - G_1)^T & -1 & 0 \\ 2(x_1 - p_1)^T & 0 & -1 \\ 0 & 0 & 1 & -\frac{1}{K} \end{bmatrix}, \\
l &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad b_{\text{QP}} = \begin{bmatrix} \gamma_3(-\|x_1 - G_1\|^2) \\ \gamma_1(-\|x_1 - p_1\|^2) \\ 0 \end{bmatrix}, \\
v &= [u_4].
\end{aligned} \tag{4.14}$$

4.2 Double-Integrator Dynamic System

we modelled the robots using double-integrator dynamic system,

$$\dot{x}_i = \overbrace{\underbrace{\begin{bmatrix} 0_{2 \times 2} & I_{2 \times 2} \\ 0_{2 \times 2} & 0_{2 \times 2} \end{bmatrix}}_A x_i + \underbrace{\begin{bmatrix} 0_{2 \times 2} \\ I_{2 \times 2} \end{bmatrix}}_B u_i}_{f(x_i, u_i)}$$

where $i = 1, \dots, N$, $u_i \in \mathbb{R}^2$ Acceleration and

$$x_i = \begin{bmatrix} x_{i,1} \\ x_{i,2} \end{bmatrix} \in \mathbb{R}^4$$

Here position $x_{i,1} \in \mathbb{R}^2$ and velocity $x_{i,2} \in \mathbb{R}^2$. The system has control over both position and velocity using acceleration as control input.

4.2.1 Model of the Tasks

In this subsection we will introduce the formulation of the tasks as constraint of a optimization problem for double-integrator dynamic system.

Formation Constraint

Robots are required to arrange themselves into a predefined shape in formation control scenarios, which are defined by the sum of the distances between each robot and the square of the velocity of the robot. We can express this problem as the minimization of a cost function by representing the formation error as $T_1(x_i)$:

$$T_1(x_i) = \sum_{i=1}^n \sum_{j \in \mathcal{N}_i} \frac{1}{2} (\|x_{i,1} - x_{j,1}\| - d_{(i,1)(j,1)})^2 + \sum_{i=1}^n \frac{1}{2} \alpha \|x_{i,2}\|^2, \quad (4.15)$$

where $\mathcal{N}_i$ specifies the set of robots with which robot i maintains a predefined distance. In this case, the desired distance between robots $(i, 1)$ and $(j, 1)$ is denoted by $d_{(i,1)(j,1)}$. These relative distances, $d_{(i,1)(j,1)}$, define the desired formation, from which the robot configuration currently operates. α is a constant parameter. $T_1(x_i)$ quantifies the distance from the desired formation. The robots have precisely formed the desired shape when $T_1(x_i) = 0$.

$$J_i(x_{i,1}) = \sum_{i=1}^n \sum_{j \in \mathcal{N}_i} \frac{1}{2} (\|x_{i,1} - x_{j,1}\| - d_{(i,1)(j,1)})^2$$

The gradient of $J_i(x_{i,1})$ with respect to the position of robot i , required to evaluate the Lie derivative of the task function, are calculated as:

$$\frac{\partial J_i(x_{i,1})}{\partial x_{i,1}} = \sum_{j \in \mathcal{N}_i} \frac{\|x_{i,1} - x_{j,1}\| - d_{(i,1)(j,1)}}{\|x_{i,1} - x_{j,1}\|} (x_{i,1} - x_{j,1})^T. \quad (4.16)$$

The following is an explanation of this equation: When robots $(i, 1)$ and $(j, 1)$ are closer than $d_{(i,1)(j,1)}$, the weight $w_{(i,1)(j,1)}$ is negative, which creates a repulsive force between the robots. In contrast, if d_{ij} is smaller than the distance between the robots, then w_{ij} is positive, which creates an attracting force between the robots [18].

The following optimization problem results in the execution of task 1:

$$\begin{aligned} \min_{u_i} \quad & \|u_i\|^2 \\ \text{subject to} \quad & \frac{\partial T_1(x_i)}{\partial x_i} f(x_i, u_i) \leq -\gamma_2 T_1(x_i) + \delta_i \end{aligned} \quad (4.17)$$

$$\begin{aligned}
& \underbrace{\begin{bmatrix} \frac{\partial T_1(x_i)}{\partial x_{i,1}} & \frac{\partial T_1(x_i)}{\partial x_{i,2}} \\ \frac{\partial J_i(x_{i,1})}{\partial x_{i,1}} & \alpha x_{i,2}^T \end{bmatrix}}_{\frac{\partial T_1(x_i)}{\partial x_i}} (Ax_i + Bu_i) \leq -\gamma T_1(x_i) + \delta_i \\
& \frac{\partial T_1(x_i)}{\partial x_i} Ax_i + \frac{\partial T_1(x_i)}{\partial x_i} Bu_i \leq -\gamma T_1(x_i) + \delta_i \\
& \begin{bmatrix} \frac{\partial T_1(x_i)}{\partial x_i} B & -1 \end{bmatrix} \begin{bmatrix} u_i \\ \delta_i \end{bmatrix} \leq -\gamma T_1(x_i) - \frac{\partial T_1(x_i)}{\partial x_i} Ax_i.
\end{aligned} \tag{4.18}$$

In addition to the attractive and repulsive forces between robots based on their positions, there is an additional force that influences the velocity of the robots. This ensures that the robots do not move too fast or too slow while forming the desired shape. Thus, the system regulates both the spatial arrangement and the velocity distribution across the robot team.

If we compare to previous section's formation constraint, we can see that in single-integrator formation constraint, the focus is on achieving the desired positions, making it simpler and quicker, but potentially less stable in terms of velocity. On the other hand, in double-integrator formation constraint, both position and velocity are regulated, resulting in smoother, more stable formation behavior. The double-integrator model is more suited for real-world applications where smooth and coordinated motion is critical for task execution.

Human-Robot Interaction Constraint

The human-robot interaction task in this context refers to the human operator controlling the collective movement of a team of robots. Specifically, the human input governs the speed of the centroid of the robot formation. The centroid speed is given by the average of the velocities of all robots in the team, represented as $\frac{1}{N} \sum x_{i,2}$, where $x_{i,2}$ is the velocity of robot i and N is the number of robots. The centre of the square formation is represented

by x_G and p_G is predefined goal.

$$\begin{aligned}
x_G &= \frac{1}{N} \sum_{i=1}^N x_{i,1} \\
v_G = \dot{x}_G &= \frac{1}{N} \sum_{i=1}^N x_{i,2} \\
a_G = \ddot{x}_G &= \frac{1}{N} \sum_{i=1}^N u_i \\
u_H &= k_p(p_G - x_G)
\end{aligned} \tag{4.19}$$

The human operator sends a reference speed u_H , which serves as the desired velocity for the entire robot team. The system tracks this reference speed using a proportional controller, which compares the human-provided reference speed u_H with the actual average velocity of the robots. The controller works to minimize the error between the reference speed and the actual average speed by adjusting the velocities of individual robots.

This tracking behavior can be represented by the following control law:

$$\frac{1}{N} \sum_{i=1}^N u_i = K_v \left(u_H - \frac{1}{N} \sum_{i=1}^N x_{i,2} \right). \tag{4.20}$$

where u_i is the control input for robot i , u_H is the reference velocity provided by the human, and K_v is the proportional gain that determines the responsiveness of the system to the human input. The term $\frac{1}{N} \sum x_{i,2}$ represents the current average velocity of the robots. The equation ensures that the robot team adjusts their velocities collectively to match the human's desired reference speed. This human-robot interaction constraint ensures that the human operator can effectively control the overall speed of the robot formation, while the robots work collaboratively to achieve the target velocity by dynamically adjusting their individual speeds.

The human operator is informed about the execution of the multiple tasks by the robots through the feedback Δu_H which offsets the reference speed, in the following fashion:

$$\frac{1}{N} \sum_{i=1}^N u_i = K_v(u_H + \Delta u_H - \frac{1}{N} \sum x_{i,2}). \tag{4.21}$$

This feedback is turned into a feedback force experienced by the human operator via a haptic device.

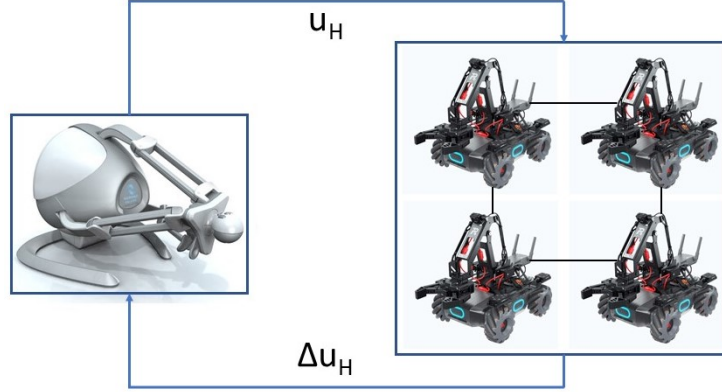


Figure 4.1: Schematic representation of the Human-Robot Interaction framework in a double-integrator dynamic system. The human operator provides velocity commands (u_H) to a team of robots. These commands are typically derived from the position or velocity inputs of the haptic device, depending on the specific implementation. The robots send feedback (Δu_H) through the haptic device, which is mapped to forces or adjustments in velocity to communicate discrepancies between the intended and actual movements, ensuring alignment and stability in the interaction.

Remark 2 Recall that $u_H \in \mathbb{R}^2$ is human input and $\Delta u_H \in \mathbb{R}^2$ is feedback from the multi-robot system to the human operator. If we consider a velocity of a given dimension, u_H , then Δu_H is the haptic feedback velocity of that same dimension.

Figure 4.1 illustrates the Human-Robot Interaction constraint in a double-integrator dynamic system for a multi-robot teleoperation scenario. The setup shows a human operator interfacing with a haptic device (on the left side of the image), through which they send commands to control the speed and trajectory of a team of robots (on the right side of the image). The human input, denoted by u_H , is the desired velocity that the human operator communicates to the robot team. As the robots receive the human's command, they attempt to execute the desired task, but deviations may occur due to system dynamics, communication delays, or unforeseen obstacles. To compensate for this, the robots send a feedback signal, denoted by Δu_H , back to the human operator via the haptic device. This feedback signal informs the human operator of the discrepancies between the intended and actual robot performance, allowing the operator to adjust their inputs accordingly. The haptic feedback provides a tactile sense of the task execution, allowing for more intuitive control and precise adjustment of the robot's behavior in real-time. The closed-loop interaction between u_H (human input) and Δu_H (robot feedback) ensures that

the human operator remains in control while also being informed of any deviations from the desired trajectory. This feedback loop is central to ensuring stability and efficiency in multi-robot teleoperation tasks, especially when managing the formation, movement, and task execution of multiple robots simultaneously.

The optimization problem encoding the execution of the human-robot interaction constraint used in the simulations is as follows:

$$\begin{aligned} & \min_{u_i, \Delta u_H} \|u_i\|^2 + K_\Delta \|\Delta u_H\|^2 \\ & \text{subject to } \frac{1}{N} \sum_{i=1}^N u_i = K_v(u_H + \Delta u_H) - \frac{1}{N} \sum x_{i,2} \end{aligned} \quad (4.22)$$

$$\begin{bmatrix} \frac{1}{N} I_{2 \times 2} \cdots \frac{1}{N} I_{2 \times 2} & -I_{2 \times 2} \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_N \\ \Delta u_H \end{bmatrix} = -K_v \frac{1}{N} \sum_{i=1}^N x_{i,2} + K_v u_H. \quad (4.23)$$

The human receives real-time feedback about the performance of the robot team through Δu_H . If the robots struggle to track the centroid's desired velocity, the human operator feels this through a haptic device, similar to the single-integrator scenario but with the added dimension of controlling the system's acceleration as well as velocity.

If we compare to previous section's human-robot interaction constraint, we can see that in the single-integrator system, the human directly controls the velocity of individual robots, whereas in the double-integrator system, the human controls the centroid of the formation, which reflects a more collective control approach. In both cases, feedback Δu_H informs the human operator about the system's performance. However, in the double-integrator case, this feedback includes information about both velocity and acceleration, giving the human operator a more comprehensive sense of the system's behavior and responsiveness. The double-integrator system introduces additional complexity by involving acceleration control, which leads to a more nuanced interaction between the human and the robots. The feedback loop in this scenario helps ensure smoother transitions and greater system responsiveness, especially when the robots need to move collectively as a formation.

Passivity Constraint

In control theory, passivity is a property that ensures a system does not generate energy, which is crucial for maintaining stability, especially in systems involving human-robot in-

teraction. Passivity can be expressed mathematically by defining the relationship between the input u_H and the output y of a system. Given the input u_H and output y , the passivity condition can be expressed as:

$$u_H^T y \geq \dot{S} \quad (4.24)$$

where $\dot{S}$ represents the rate of change of the storage function S that is $S = \frac{1}{2}x^T \pi x$, which is a measure of the stored energy in the system. Here, π is $\begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix}$.

In the context of human-robot interaction, this passivity condition can be extended to include the interaction between the human input u_H and the system's response Δu_H :

$$u_H^T \Delta u_H \geq \dot{S} \quad (4.25)$$

Further expanding, the relationship between the input and the system's state x can be described by:

$$u_H^T \Delta u_H \geq x_i^T \pi \dot{x}_i \quad (4.26)$$

where π is a matrix that characterizes the energy distribution within the system. Considering the system dynamics, the expression can be simplified to:

$$\begin{aligned} u_H^T \Delta u_H &\geq x_i^T \pi \left(\begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} x_i + \begin{bmatrix} 0 \\ I \end{bmatrix} u_i \right) \\ u_H^T \Delta u_H &\geq x_i^T \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} x_i + x_i^T \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 \\ I \end{bmatrix} u_i \\ u_H^T \Delta u_H &\geq 0 + x_i^T \begin{bmatrix} 0 \\ I \end{bmatrix} u_i \\ u_H^T \Delta u_H &\geq x_{i,2}^T u \\ \begin{bmatrix} x_{i,2}^T & -u_H^T \end{bmatrix} \begin{bmatrix} u \\ \Delta u_H \end{bmatrix} &\leq 0 \end{aligned} \quad (4.27)$$

For systems with multiple agents (such as multiple robots), the passivity condition can be distributed among N agents, leading to:

$$\begin{bmatrix} x_{i,2}^T & \frac{-u_H^T}{N} \end{bmatrix} \begin{bmatrix} u_i \\ \Delta u_H \end{bmatrix} \leq 0 \quad (4.28)$$

Passivity is only required for double-integrator. Because the single-integrator does not accumulate energy.

4.2.2 Optimization-based Centralized Control Framework

The resulting optimization algorithm is a standard quadratic program:

$$\begin{aligned} \min_{v=[u_1^T \dots u_4^T \delta^T \Delta u_H^T]^T} \quad & v^T H v \\ \text{subject to} \quad & A_{\text{QP}} v \leq b_{\text{QP}}, \\ & C_{\text{QP}} v = d_{\text{QP}}. \end{aligned} \quad (4.29)$$

where $H \in \mathbb{R}^{14 \times 14}$, $A_{\text{QP}} \in \mathbb{R}^{8 \times 14}$, $C_{\text{QP}} \in \mathbb{R}^{2 \times 14}$, $b_{\text{QP}} \in \mathbb{R}^{8 \times 1}$, $d_{\text{QP}} \in \mathbb{R}^{2 \times 1}$, and $v \in \mathbb{R}^{14 \times 1}$.

$$\begin{aligned} H &= \begin{bmatrix} I_{8 \times 8} & 0_{8 \times 4} & 0_{8 \times 2} \\ 0_{4 \times 8} & K_\delta I_{4 \times 4} & 0_{4 \times 2} \\ 0_{2 \times 8} & 0_{2 \times 4} & K_\Delta I_{2 \times 2} \end{bmatrix}, \quad A_{\text{QP}} = \begin{bmatrix} A_{\text{QP},11} & -I_{4 \times 4} & 0_{4 \times 2} \\ A_{\text{QP},21} & 0_{4 \times 4} & -\frac{u_H^T}{N} \otimes \mathbb{I}_{4 \times 1} \end{bmatrix} \\ A_{\text{QP},11} &= \begin{bmatrix} \frac{\partial T_1}{\partial x_1} & 0 & 0 & 0 \\ 0 & \frac{\partial T_1}{\partial x_2} & 0 & 0 \\ 0 & 0 & \frac{\partial T_1}{\partial x_3} & 0 \\ 0 & 0 & 0 & \frac{\partial T_1}{\partial x_4} \end{bmatrix} \otimes B, \quad A_{\text{QP},21} = \begin{bmatrix} x_{1,2}^T & 0 & 0 & 0 \\ 0 & x_{2,2}^T & 0 & 0 \\ 0 & 0 & x_{3,2}^T & 0 \\ 0 & 0 & 0 & x_{4,2}^T \end{bmatrix}, \\ b_{\text{QP}} &= \begin{bmatrix} -\gamma T_1 - \frac{\partial T_1}{\partial x_1} A x_1 \\ -\gamma T_1 - \frac{\partial T_1}{\partial x_2} A x_2 \\ -\gamma T_1 - \frac{\partial T_1}{\partial x_3} A x_3 \\ -\gamma T_1 - \frac{\partial T_1}{\partial x_4} A x_4 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad v = \begin{bmatrix} [u_i]_{8 \times 1} \\ [\delta]_{4 \times 1} \\ [\Delta u_H]_{2 \times 1} \end{bmatrix} \\ C_{\text{QP}} &= \begin{bmatrix} \frac{1}{N} I & \frac{1}{N} I & \frac{1}{N} I & \frac{1}{N} I & 0 & -K_v I \end{bmatrix}, \quad d_{\text{QP}} = K_v u_H - K_v \frac{1}{N} \sum_{i=1}^N x_{i,2}. \end{aligned} \quad (4.30)$$

Inequality Constraint ($A_{\text{QP}} v \leq b_{\text{QP}}$) : Ensures task feasibility for each robot based on

formation control.

$$A_{QP}v \leq b_{QP}$$

$$\begin{bmatrix} A_{QP,11} & -I_{4 \times 4} & 0_{4 \times 2} \\ A_{QP,21} & 0_{4 \times 4} & -\frac{u_H^T}{N} \otimes \mathbb{1}_{4 \times 1} \end{bmatrix} \begin{bmatrix} [u_i]_{8 \times 1} \\ [\delta]_{4 \times 1} \\ [\Delta u_H]_{2 \times 1} \end{bmatrix} \leq \begin{bmatrix} -\gamma T_1 - \frac{\partial T_1}{\partial x_1} Ax_1 \\ -\gamma T_1 - \frac{\partial T_1}{\partial x_2} Ax_2 \\ -\gamma T_1 - \frac{\partial T_1}{\partial x_3} Ax_3 \\ -\gamma T_1 - \frac{\partial T_1}{\partial x_4} Ax_4 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (4.31)$$

Equality Constraint ($C_{QP}v = d_{QP}$) : Governs the velocity centroid alignment with the human-provided reference velocity (u_H).

$$C_{QP}v = d_{QP}$$

$$\begin{bmatrix} \frac{1}{N}I & \frac{1}{N}I & \frac{1}{N}I & \frac{1}{N}I & 0 & -K_v I \end{bmatrix} \begin{bmatrix} [u_i]_{8 \times 1} \\ [\delta]_{4 \times 1} \\ [\Delta u_H]_{2 \times 1} \end{bmatrix} = K_v u_H - K_v \frac{1}{N} \sum_{i=1}^N x_{i,2} \quad (4.32)$$

Remark 3 (Convexity of the Passivity-Based haptic feedback Synthesis Framework)

The convex nature of the optimization framework ensures that it can be solved efficiently, even under real-time constraints. This efficiency is critical for practical applications involving human-robot systems, where dynamic environments and human inputs require immediate responses. By leveraging the convexity of the problem, the framework achieves computational feasibility, making it suitable for online implementation and scenarios with stringent timing requirements.

Remark 4 (Decentralized optimal haptic feedback synthesis) *In decentralizing the equality constraint, the framework allows individual control over separate robots. However, in this thesis, the optimization framework for the double-integrator dynamic system is specifically designed so that the human operator controls the entire team of robots as a cohesive unit, rather than managing each robot individually as in the single-integrator case. This necessitated the choice of a centralized framework, ensuring coordinated control and smoother task execution across the robot team. In general if the human-robot interaction modality can be decentralized then the whole optimal haptic feedback synthesis control framework is decentralized.*

Chapter 5

Simulations and Experimental Results

In this chapter we will discuss the simulation of the human feedback synthesis in a multi-robot multi-task system for single-integrator dynamic system as well as the simulation and experiment of the human feedback synthesis in a multi-robot multi-task system for double-integrator dynamic system.

5.1 Single-Integrator Dynamic System Simulation

The simulation of the human feedback synthesis in a multi-robot multi-task system have been implemented using ROS2 Humble. Webots served as a dynamic simulator. The system with human input and the system without human input are the two scenarios covered by the simulation. The experiment utilized DJI Mavic 2 Pro quadcopters as the robotic platform. These quadcopters have a fixed altitude. Since we do not control the altitude in the simulator, the simulation represents a 2D version of the system.

Webot: Webots is a professional mobile robot simulation software developed by Cyberbotics Ltd. It provides a comprehensive and flexible environment for modeling, programming, and simulating mobile robots. One of its standout features is its ability to prototype robots quickly and efficiently, enabling researchers to create, simulate, and transfer control programs to real robots. It includes libraries of sensors and actuators and allows users to define properties such as shape, color, and texture for each robot, facilitating the creation of complex and varied robot environments. Webots supports simulations of multi-agent

systems, including wheeled, legged, and flying robots, and can handle various sensors and actuators. Importantly, Webots enables the transfer of simulated controllers to actual robots, allowing for seamless integration from simulation to real-world application. It supports programming languages such as C, C++, Python and Java and can interface with third-party software like MatLab or LabView. In terms of functionality, Webots allows the simulation to run faster than real-time, offering detailed step-by-step analysis. It also incorporates physics-based simulations through the Open Dynamics Engine (ODE) for accurate robot behavior. Additionally, Webots can manage large environments and optimize them for faster simulations [64].

In the context of my work, Webots provides a robust platform for simulating and prototyping systems, allowing me to test multi-robot coordination, task allocation, and human-robot interaction before transferring these concepts to real-world implementations. The ability to simulate multi-agent systems and transfer results to actual robots aligns with my research in optimizing human-multi-robot task execution.

Although the proposed multi-robot multi-task system is effective for a large variety of tasks [15], in this section we consider three prototypical robotic tasks. They are as follows: 1) The go to goal task; 2) The formation control task; and 3) Human-Robot interaction. We are considering 4 robots for the system. Each of these robots have different set of priorities in terms of executing the tasks and these priorities will be specified in the following subsection, together with the task functions.

5.1.1 Multi-Robot Multi-Task Execution without Human Feedback Synthesis

In the scenario of multi-robot multi-task execution, there are four robots: the highest priority for robots 1, 3, and 4 is to assemble a rectangular formation, while for robot 2 the go-to-goal task has highest priority. In summary, the task constraints for the robots are:

1. Proceeding in the direction of the predefined goal (task T_1)
2. Keeping the rectangle-shaped formation (task T_2)

whereas their priorities are ($T_i \prec T_j$ means that task i has higher priority than task j):

1. Robots 1, 3, 4: $T_2 \prec T_1$
2. Robot 2: $T_1 \prec T_2$

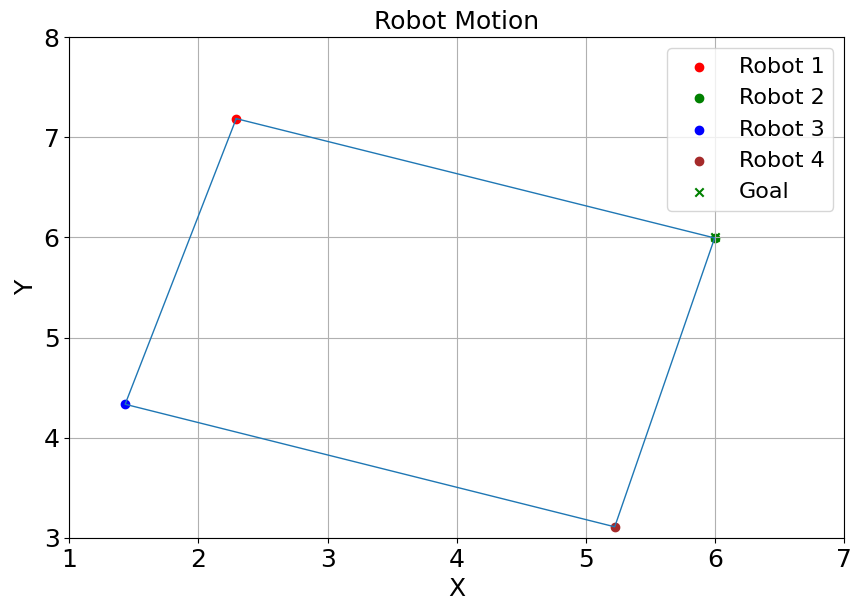


Figure 5.1: Multi-robot multi-task execution without human input in Python. The four robots are represented by circles of four colors (red, green, blue, and brown, respectively). The green cross mark depicts a designated goal location. The robots assemble a rectangular shape using formation control (highest priority task for robots 1, 3, and 4) as robot 2 eventually reaches the goal (its highest priority task).

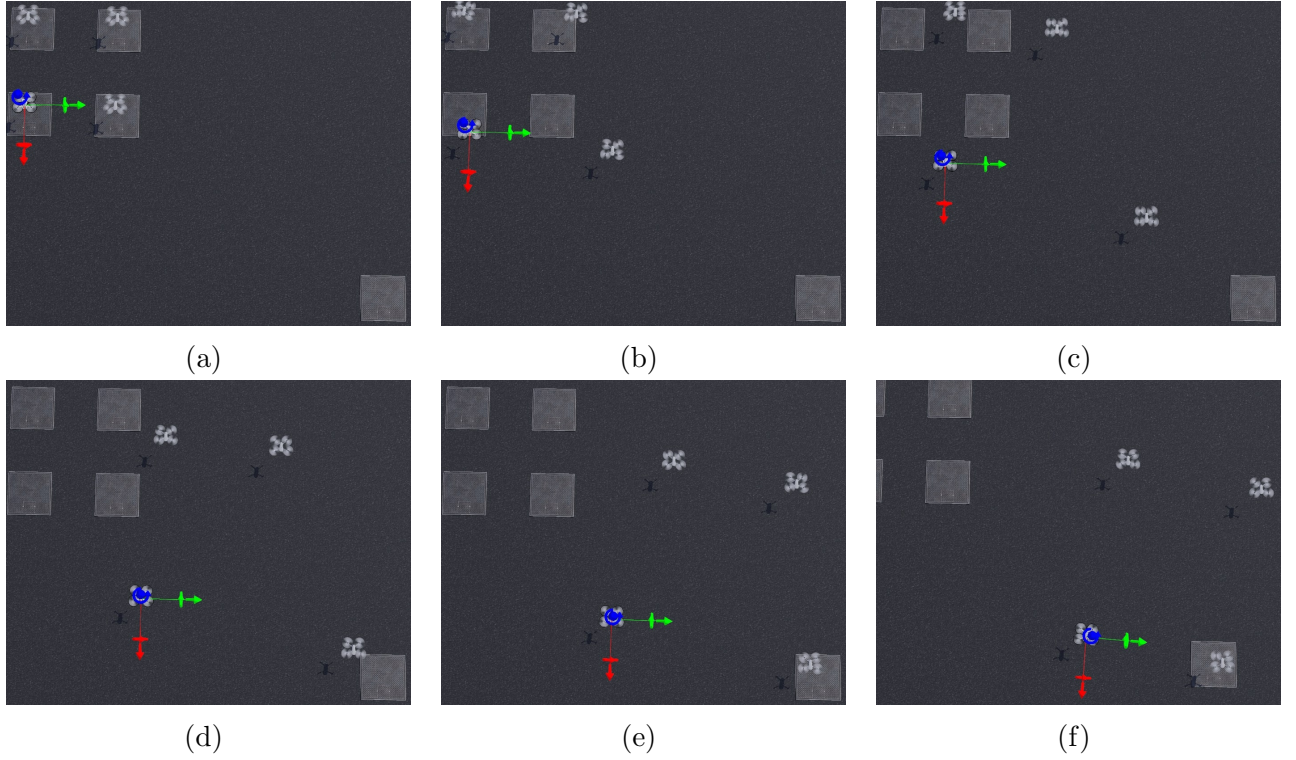


Figure 5.2: Snapshots of the simulations performed in Webot implemented using ROS2 humble. The pictures show four DJI Mavic 2 Pro quadcopters starting from their initial positions (Fig. 5.2a), moving in the environment towards the goal location (gray square in the lower right corner) and assembling a rectangular formation (Figures 5.2b, 5.2c, 5.2d, 5.2e), and reaching their final configuration (Fig. 5.2f).

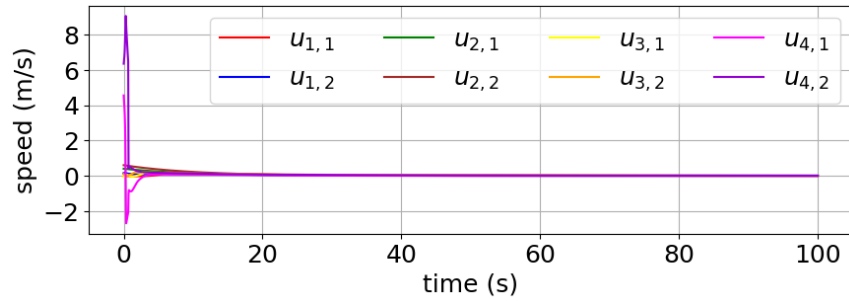


Figure 5.3: Control input vectors of the four robots with respect to time. As can be seen, the control signals produced by the task allocation optimization algorithm are smooth throughout the simulation.

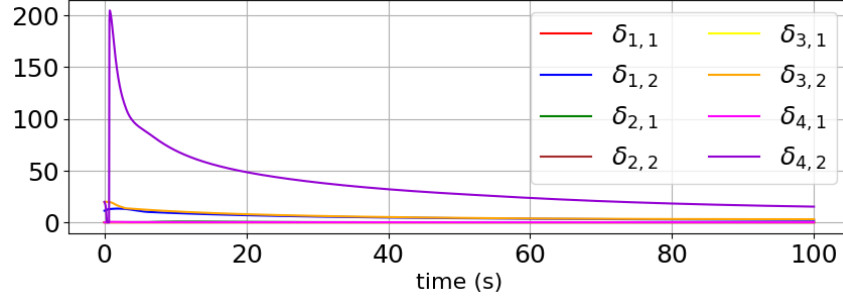


Figure 5.4: Slack variables of the four robots with respect to time. The smaller the value of $\delta_{i,j}$, the higher the priority of task j for robot i .

Figure 5.1 displays the multi-robot multi-task execution without human input. Robots 1, 2, 3, and 4 are represented by red, green, blue, and brown circles, respectively. The predefined goal is represented by the green cross. In this figure, robots 1, 2, 3, and 4 form a rectangle; in addition, robot 2 reaches the goal location. The following controller parameters have been chosen for this simulation: $k = 1000$, $k_\delta = 0.001$, $\gamma_1 = 0.1$, and $\gamma_2 = 0.1$.

Figure 5.2 displays the different state of multi-robot multi-task execution without human input performed using the Webot simulator and implemented in ROS2 humble, where four DJI Mavic 2 Pro quadcopters are simulated. The predetermined goal is represented by a gray square in the lower right corner, and the initial positions of the four quadcopters are represented by the four gray squares in the upper left corner. The initial position of the four quadcopters are shown in Fig. 5.2a. Then, Figures 5.2b–5.2e show the positions of the quadcopters over the course of the experiment, moving towards the goal and assembling the desired shape; finally, in Fig. 5.2f, the four quadcopters form a rectangular shape as robot 2 eventually reaches the goal.

In Fig. 5.3, we report the control inputs, u_i , of the four robots with respect to time. The two components of the control inputs of robot i is represented by $u_{i,1}(t)$ and $u_{i,2}(t)$. From the figure, we can appreciate the continuity and smoothness of the control signals produced by the multi-task allocation optimization-based controller. Figure 5.4 displays the slack variables, δ_i , of the four robots with respect to time. Similarly to the case of the control input, the slack variables of robot i are denoted as $\delta_{i,1}(t)$ and $\delta_{i,2}(t)$. As can be seen, the values of $\delta_{i,j}$ corresponding to tasks with higher priorities are smaller.

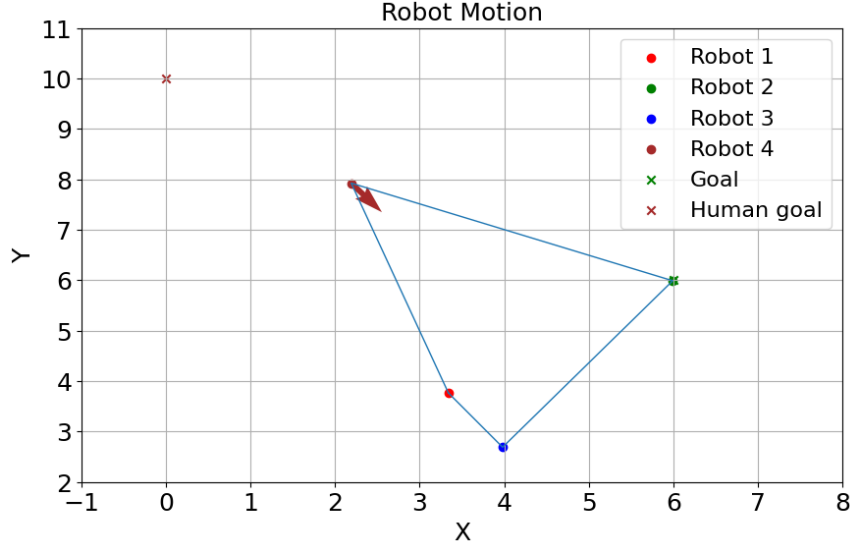


Figure 5.5: Multi-robot multi-task execution with human feedback synthesis. Circles in the colors red, green, blue, and brown, respectively, stand in for the four robots. The green cross mark depicts the desired goal for task T_1 , while brown cross represents the goal where a human operator drives robot 4. Due to a human input u_H to robot 4, robots are unable to form a rectangle. Robot 4 is prevented from reaching the brown cross by an opposing repulsive force, symbolized by a brown arrow.

5.1.2 Multi-Robot Multi-Task Execution with Human Feedback Synthesis

In this section, we present the results of simulations obtained introducing an artificial human input, u_H , that is able to specify the desired velocity of robot 4. All the constraints and priorities are the same as in the previous subsection, except for the additional equality constraint related to the human-robot interaction for robot 4. The controller parameters for this scenario have been set to: $k = 500$, $k_\delta = 0.1$, $k_\Delta = 1000$, $\gamma_1 = 0.0005$, and $\gamma_2 = 0.1$.

Figure 5.5 shows the four robots, the goal location (green cross) used for task T_2 , and the location where the artificial human input is trying to drive robot 4 (brown cross). As can be seen, compared to the previous case, the robots attempting to form a rectangular shape are disturbed by the human input u_H trying to drive robot 4 to the brown cross. In the end, a feedback force (depicted as a brown arrow) informs the human about the direction towards which the team of robots would like to move in order to execute their tasks with their respective priorities.

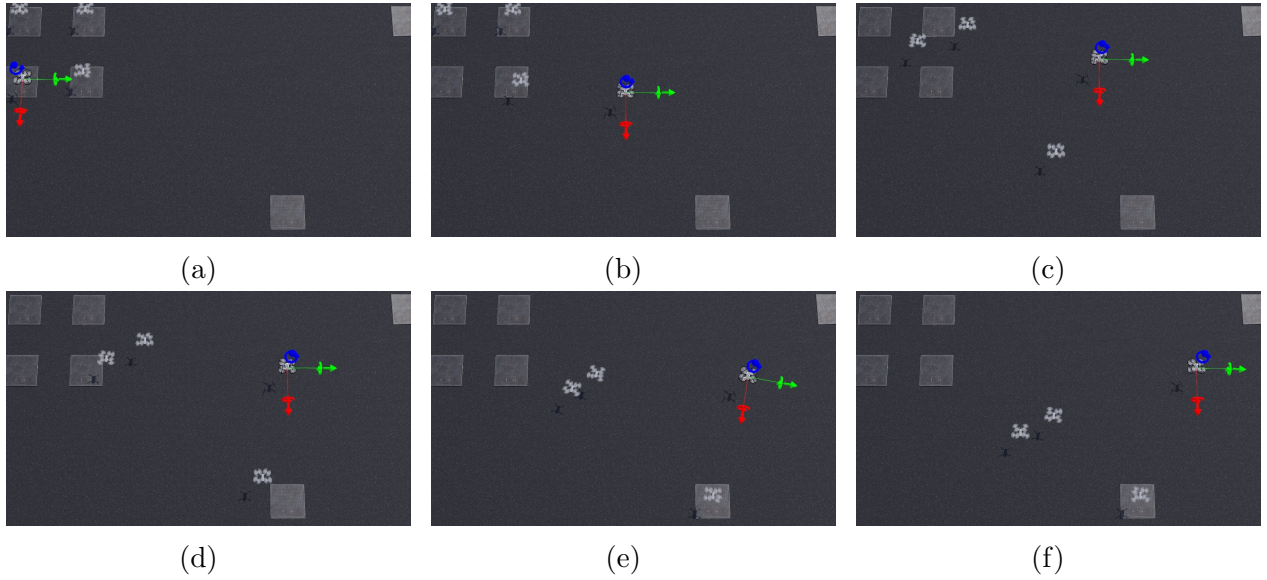


Figure 5.6: Snapshots of the simulations performed in Webots and implemented using ROS2 humble. The scenario is similar to the one presented in the previous section, except for the presence of a human input that controls robot 4 to move towards the gray square in the upper right corner. The quadcopters attempt to assemble the desired rectangular shape, while one of them reaches the desired goal (gray square at the bottom right), as prescribed by its task priorities. The task execution is traded off with the human input, and the mismatch between desired and actual robot behavior is synthesized in the optimal haptic feedback proportional to Δu_H^* .

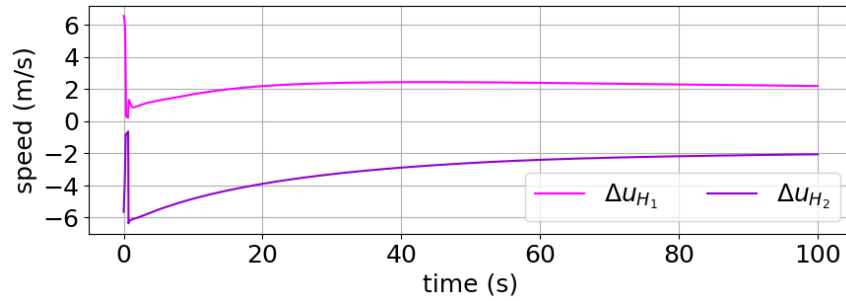


Figure 5.7: Human feedback with respect to time. Δu_H does not converge to 0, as a result there will be a non-zero haptic feedback.

Figures 5.6a–5.6f display the different state of multi-robot multi-task execution with human feedback synthesis simulated using Webot and implemented in ROS2 humble. The setup is similar to the previous case except for the fact that robot 4 is pulled towards the gray square in the upper right corner by an artificial human input. The initial position of the four quadcopters is shown in Fig. 5.6a. The quadcopters move in the attempt of reaching the desired goal, assembling the rectangular shape, and following the human input (Figures 5.6b–5.6e), until they settle in their final configuration shown in Fig. 5.6f.

Finally, Fig. 5.7 reports the human feedback, Δu_H , with respect to time. The two components of the human feedback is represented by $\Delta u_{H_1}(t)$ and $\Delta u_{H_2}(t)$. As can be seen, Δu_H does not converge to 0: this means that the human input is not executed exactly. In fact, the haptic feedback proportional to Δu_H will be sent to the human operator to inform them about this mismatch between desired and actual robot behavior.

5.2 Double-Integrator Dynamic System Simulation and Experiment

The simulation of the Passivity-based human feedback synthesis in a multi-robot multi-task system have been implemented using ROS2 Humble. Webots served as a dynamic simulator. 1) The system with passivity and human input and 2) the system without passivity and with human input are the two scenarios covered by the simulation. The simulation utilized DJI Mavic 2 Pro quadcopters as the robotic platform. These quadcopters have a fixed altitude. Since we do not control the altitude in the simulator, the simulation represents a 2D version of the system.

While the experiment of the Passivity-based human feedback synthesis in a multi-robot multi-task system have been implemented using four DJI Robomaster EP in the Robohub at the University of Waterloo. We used Novint Falcon game controller as a haptic device to send instructions to the robots and receive feedback also. 1) The system with passivity and human input and 2) the system without passivity and with human input are the two scenarios covered by the experiment. The experiment utilized DJI Robomaster EP as the mobile robotic platform. These DJI Robomaster EPs move on a 2D plane. So, the simulation represents a 2D version of the system.

Novint Falcon Game Controller: The Novint Falcon Game Controller is a relatively inexpensive 3-DOF (degrees of freedom) haptic device originally developed for the gaming industry. It uses a form similar to the delta-robot configuration, which restricts its end-effector to translational degrees of freedom only. This device is commonly applied in

haptic feedback systems due to its high stiffness and low inertia compared to serial kinematic configurations. In terms of haptic feedback for robotics, the Novint Falcon provides tactile sensations to the human operator based on the forces the robot experiences in real time. The Falcon’s system operates at around 800-1000 Hz, where forces and position updates are communicated between the device and a controlling computer, creating smooth and responsive feedback for the user. Through its force feedback capabilities, it can be integrated into teleoperation systems where the user controls the robots remotely, and the device generates a haptic response to inform the operator about obstacles or task difficulties encountered by the robot. The controller allows for precise control and feedback during teleoperation, making it an ideal tool for human-multi-robot interaction tasks, such as the one in your research. The Novint Falcon’s ability to simulate real-world forces and motions enhances the human operator’s capacity to manage and interact with a team of robots in real-time [65].

In the context of my work, this device facilitates control over the robot team and ensures real-time interaction through haptic feedback, allowing the operator to perceive forces acting on the robots and make corrections based on this sensory input. This aligns well with my system’s goals of optimizing task allocation while incorporating human feedback during robot interactions.

Although the proposed multi-robot multi-task system is effective for a large variety of tasks [15], in this section we consider three prototypical robotic tasks. They are as follows: 1) The formation control task; 2) Human-Robot interaction; and 3) The passivity-based control task. We are considering 4 robots for the system. Each of these robots have same set of priorities in terms of executing the tasks and these priorities will be specified in the following subsection, together with the task functions.

5.2.1 Centralized Multi-Robot Multi-Task Execution with Passivity and Human Feedback

In the scenario of multi-robot multi-task execution with passivity and human input, there are four robots: the priority for robots 1, 2, 3, and 4 is to assemble a square formation and maintain passivity as well as follow the human instructions. In summary, the task constraints for the robots are:

1. Keeping the square-shaped formation (task T_1)
2. Maintaining the passivity

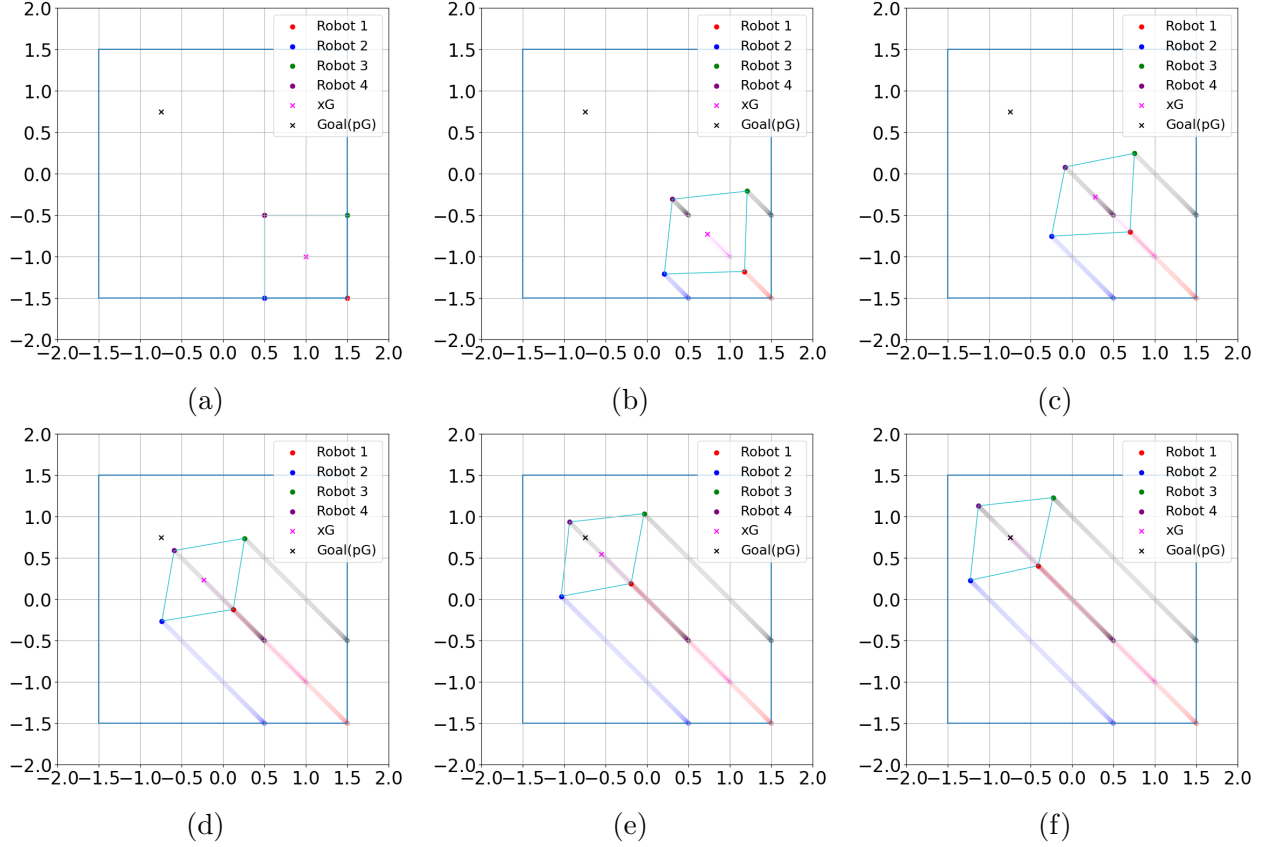


Figure 5.8: Snapshots of the simulations of centralized multi-robot multi-task execution with passivity and human feedback implemented using Python. The pictures show four robots starting from their initial positions (Fig. 5.8a), moving in the environment towards the goal location (black cross (p_G) in the upper left corner) and assembling a square formation (Figures 5.8b, 5.8c, 5.8d, 5.8e), and reaching their final configuration (Fig. 5.8f).

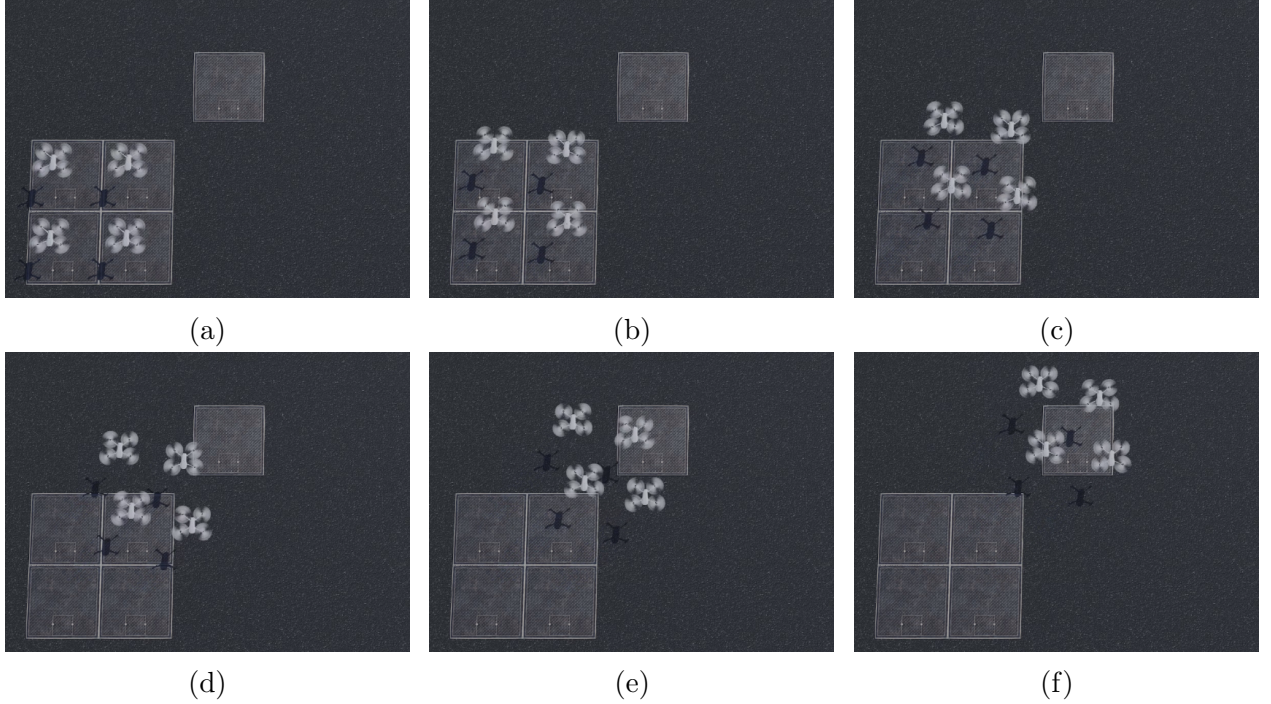


Figure 5.9: Snapshots of the experiments of centralized multi-robot multi-task execution with passivity and human feedback performed in ROS2 humble using Webot as the simulator. The pictures show four DJI Mavic 2 Pro quadcopters starting from their initial positions (Fig. 5.9a) represented by four gray squares in the bottom left corner, moving in the environment towards the goal location (gray square in the upper right) and assembling a square formation (Figures 5.9b, 5.9c, 5.9d, 5.9e), and reaching their final configuration (Fig. 5.9f).

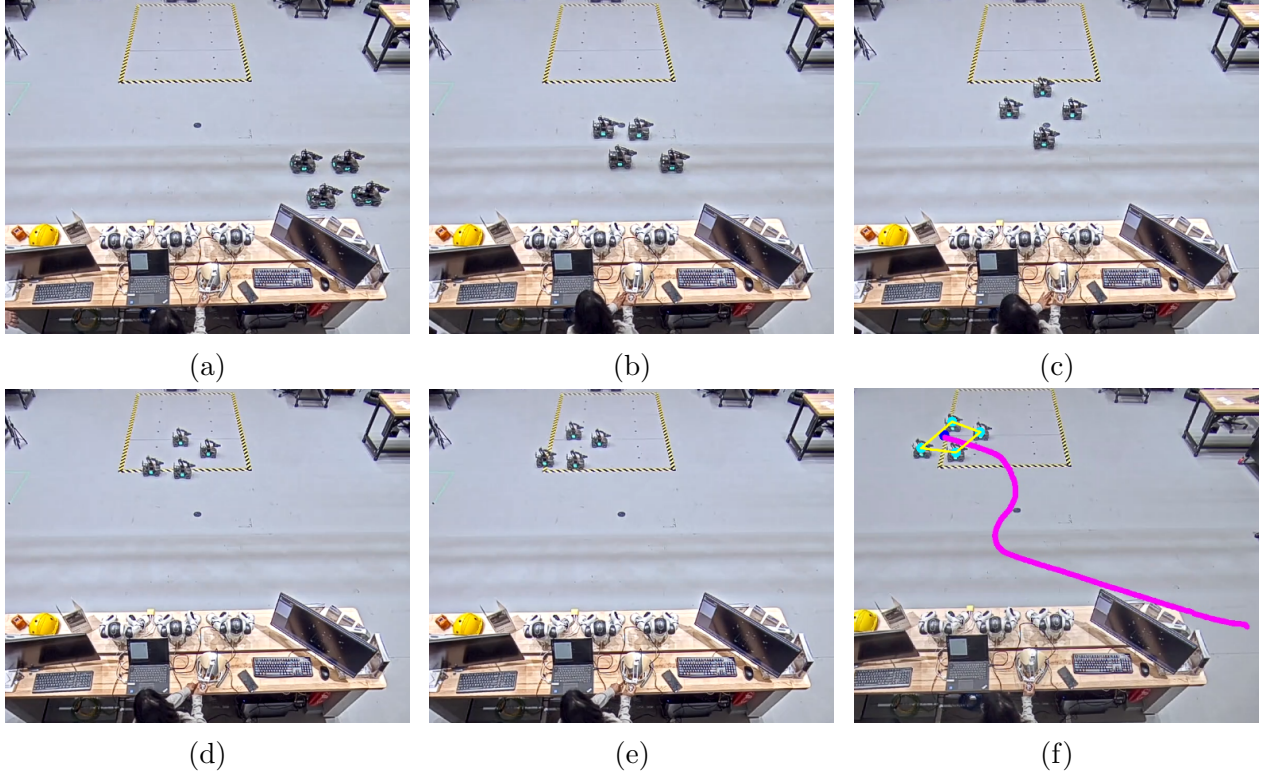


Figure 5.10: Snapshots of the experiments of centralized multi-robot multi-task execution with passivity and human feedback performed in the Robohub at the University of Waterloo. The pictures show four DJI Robomaster EP mobile platforms starting from their initial positions (Fig. 5.10a), moving in the environment towards the goal location and assembling a square formation (Figures 5.10b, 5.10c, 5.10d, 5.10e), and reaching their final configuration (Fig. 5.10f).

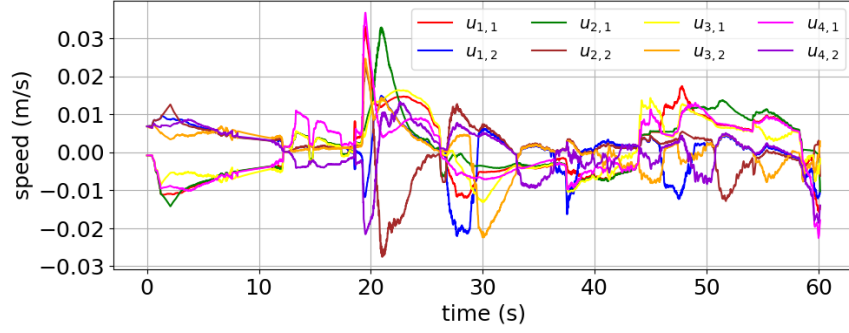


Figure 5.11: Control input vectors of the four robots with respect to time. As can be seen, the control signals produced by the optimization algorithm demonstrates the effectiveness of the passivity-based control system in managing the robots' responses under real-world experimental conditions. It confirms that the control inputs, despite fluctuations, are constrained within safe limits, ensuring smooth and stable operations for all robots.

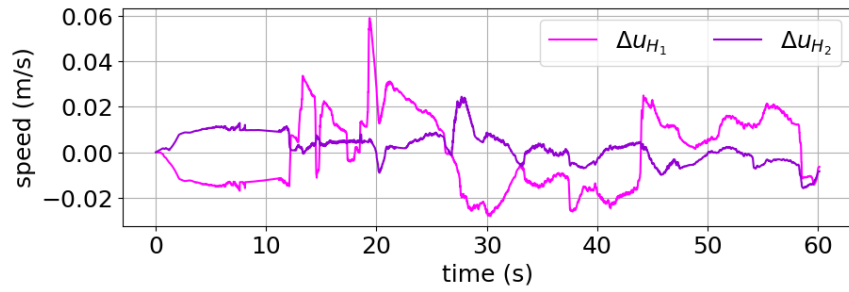


Figure 5.12: Human feedback with respect to time. Δu_H does not converge to 0, as a result there will be a non-zero haptic feedback.

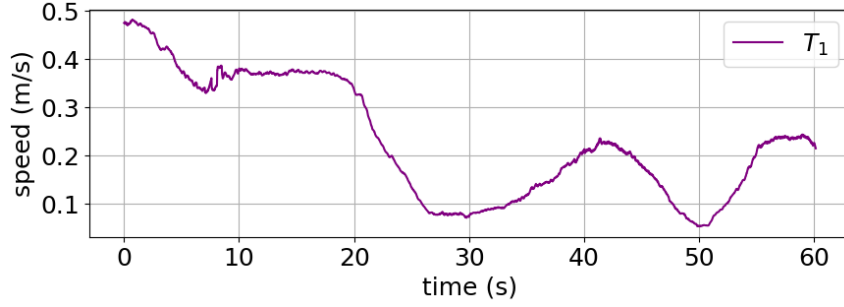


Figure 5.13: Formation graph of the four robots with respect to time. As can be seen, the smooth oscillations in T_1 indicate that the system dynamically adjusts the velocities of the robots to maintain the desired formation.

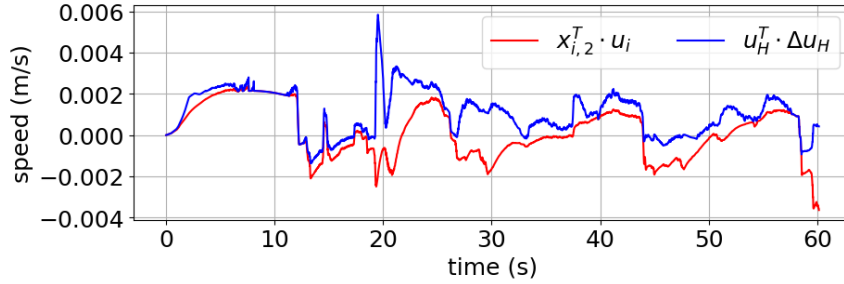


Figure 5.14: Passivity graph of the four robots with respect to time. As can be seen, the system's velocity closely follows the human input, but without significant deviation, which suggests that the control strategy is successfully enforcing the passivity condition.

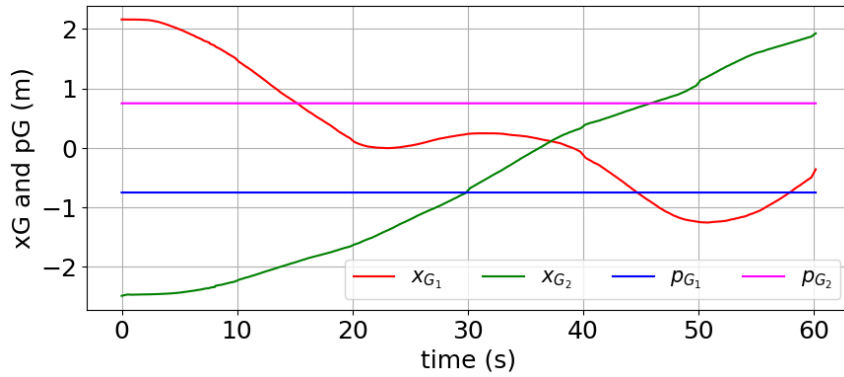


Figure 5.15: x_G and p_G with respect to time. As can be seen, the robot system, while adhering to passivity constraints, adjusts smoothly toward the goal.

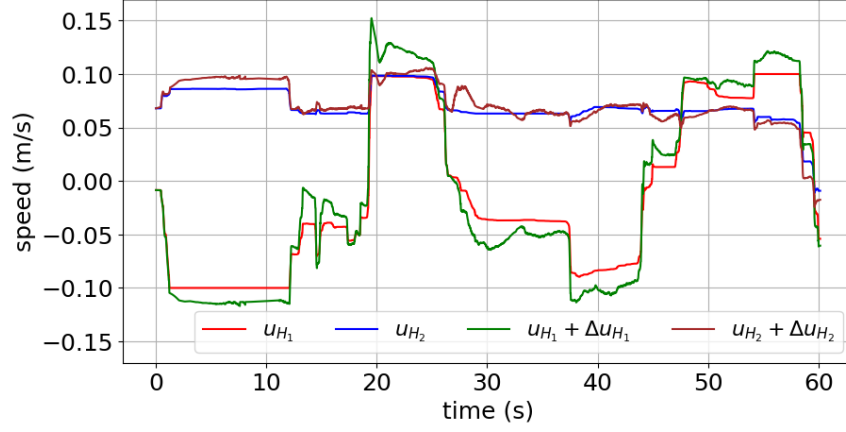


Figure 5.16: u_H and $u_H + \Delta u_H$ with respect to time. As can be seen, The difference between the u_H and $u_H + \Delta u_H$ lines indicates the magnitude of the adjustments made by the system. If the adjustments (Δu_H) are small, it suggests that the system is only making minor corrections to maintain stability. Larger deviations might indicate more significant interventions by the system to correct potential issues or instabilities.

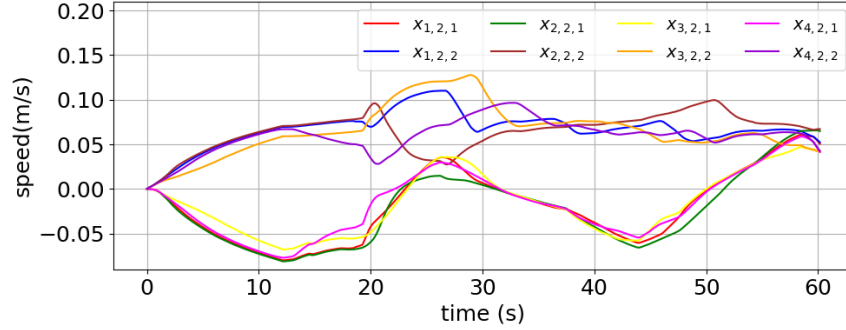


Figure 5.17: The evolution of the velocity components x_2 for each of the four robots over time. As can be seen, the velocities of all the robots start at certain initial values and evolve over time. The general trend of these lines indicates how each robot's velocity changes in response to the control inputs, including any adjustments made by the system to maintain formation and stability.

3. Following the human instruction of driving the centre of the square formation towards the direction of the predefined goal

Figure 5.8a–5.8f displays the the different state of the simulations of multi-robot multi-task execution with passivity and human input in Python. Robots 1, 2, 3, and 4 are represented by red, blue, green, and purple circles, respectively. The predefined goal (p_G) is represented by the black cross. In this figure, robots 1, 2, 3, and 4 form a square; in addition, the centre (x_G) of the square formed by the four robots reaches the goal location (p_G).

The following controller parameters have been chosen for this simulation: $k = 35$, $k_h = 1$, $k_v = 0.1$, $k_p = 0.1$, $\gamma = 0.3$, and $\alpha = 2$.

Figure 5.9 displays the different state of the simulations of multi-robot multi-task execution with passivity and human input performed in ROS2 humble using Webot as the simulator, where four DJI Mavic 2 Pro quadcopters are served as robots. The predetermined goal is in the upper right corner represented by a gray square, and the initial positions of the four DJI Mavic 2 Pro quadcopters are represented by the four gray squares in the bottom left corner. The initial position of the four DJI Mavic 2 Pro quadcopters are shown in Fig. 5.9a. Then, Figures 5.9b–5.9e show the positions of the four DJI Mavic 2 Pro quadcopters over the course of the experiment, moving towards the goal (gray square in the upper right) and assembling the desired shape; finally, in Fig. 5.9f, the four DJI Mavic 2 Pro quadcopters form a square shape as the centre of the square formation formed by the robots eventually reaches the goal.

Figure 5.10 displays the different states of the experiment of multi-robot multi-task execution with passivity and human input performed in the Robohub at the University of Waterloo, where four DJI Robomaster EP mobile platforms are served as robots. The predetermined goal is in the upper left corner, and the initial positions of the four DJI Robomaster EP are in the lower right corner. The initial position of the four DJI Robomaster EP are shown in Fig. 5.10a. Then, Figures 5.10b–5.10e show the positions of the four DJI Robomaster EP over the course of the experiment, moving towards the goal and assembling the desired shape; finally, in Fig. 5.10f, the four DJI Robomaster EP form a square shape as the centre of the square formation formed by the robots eventually reaches the goal.

In Fig. 5.11, we report the control inputs, u_i , of the four robots with respect to time. The two components of the control inputs of robot i is represented by $u_{i,1}(t)$ and $u_{i,2}(t)$. From the figure, we can see the control inputs fluctuate within a relatively small range, both positively and negatively. These fluctuations reflect real-time corrections made by the

control system to maintain stable robot performance in response to external commands, ensuring the passivity constraint is upheld. This is essential for avoiding instability or oscillatory behaviors in robot movement. Although each robot's control inputs exhibit their own unique fluctuations, the general trend shows a synchronized pattern, which indicates that the control system successfully coordinates the robots' movements in a centralized manner. Figure 5.12 reports the human feedback, Δu_H , with respect to time. The two components of the human feedback is represented by $\Delta u_{H_1}(t)$ and $\Delta u_{H_2}(t)$. As can be seen, Δu_H does not converge to 0, this means that the human input is not executed exactly. In fact, the haptic feedback proportional to Δu_H will be sent to the human operator to inform them about this mismatch between desired and actual robot behavior.

The plot in Figure 5.13 shows that at the beginning of the experiment, the speed starts high, around 0.4 m/s, and gradually decreases until around 20 seconds, where T_1 stabilizes at a lower value before rising and falling again in a smooth pattern. The smooth oscillations in the speed indicate that the system dynamically adjusts the velocities of the robots to maintain the formation. The controlled rise and fall demonstrate that the passivity-based control ensures that the energy in the system is regulated, preventing instability while allowing the formation to react smoothly to human input and external forces. Figure 5.14 illustrates the passivity-based control of the system over time. The two key components are $x_{i,2}^T u_i$ (red line), representing the system's velocity control, and $u_H^T \Delta u_H$ (blue line), which reflects the human input's influence on the system. Both values fluctuate as the control system adjusts to maintain stability and ensure passivity is preserved during the interaction. As observed, the fluctuations are relatively small, indicating that the control system is successfully ensuring that the robots respond to human inputs without generating excess energy, which could destabilize the system. The alignment between the two curves indicates that the human input is consistently adjusted by the system to balance between task execution and passivity constraints, thus ensuring a smooth interaction. This graph validates the effectiveness of the passivity-based control system, showcasing the consistent behavior of the robot system under human input while maintaining the required passivity condition.

Figure 5.15 shows x_{G_1} and x_{G_2} fluctuate as the robots adjust their movement to track the desired target positions p_{G_1} and p_{G_2} , which remain constant throughout the experiment. The graph demonstrates that the robot system, while adhering to passivity constraints, adjusts smoothly toward the goal. The fluctuations indicate the system's efforts to maintain stability and balance between following the human input and preserving passivity, leading to coordinated movement. Figure 5.16 provides insight that throughout the experiment, the feedback signals (Δu_H) make necessary corrections to the original human inputs to ensure passivity is maintained. The green and brown lines deviate from the red and blue

lines, showing how the feedback adjusts the control inputs. These deviations are more pronounced when the human inputs (red and blue lines) cause larger changes in the robot system, indicating that the feedback compensates for those changes to maintain stability. Figure 5.17 provides a detailed look at how the velocity states of the robots evolve under the centralized control framework. The trends observed in the graph can be used to assess the effectiveness of the control strategy in managing the velocities of the robots, ensuring they perform their tasks in a coordinated and stable manner. This graph is crucial for verifying that the system is achieving the desired dynamic behavior and that the velocities of the robots are being controlled appropriately to maintain system stability and task performance.

5.2.2 Centralized Multi-Robot Multi-Task Execution without Passivity and with Human Feedback

In this section, we present the results of simulations and experiments obtained in the optimization algorithm without passivity and with human input, u_H . All the constraints and priorities are the same as in the previous subsection, except the absence of the second task which is passivity constraint. The controller parameters for this scenario have been set to: $k = 35$, $k_h = 1$, $k_v = 0.1$, $k_p = 0.1$, $\gamma = 0.3$, and $\alpha = 2$.

Figure 5.18a–5.18f shows the multi-robot multi-task execution without passivity and with human input. Robots 1, 2, 3, and 4 are represented by red, blue, green, and purple circles, respectively. The predefined goal (p_G) is represented by the black cross. In this figure, robots 1, 2, 3, and 4 form a square; in addition, the centre (x_G) of the square formed by the four robots reaches the goal location (p_G). As can be seen, compared to the previous case, the robots attempting to form a square shape but the shape of the formation is not a proper square like previous case since absence of passivity interrupts the formation control and human instructions.

Figures 5.19a–5.19f display the different state of the experiment of multi-robot multi-task execution with human feedback synthesis in ROS2 humble using Webot as the simulator. The setup is similar to the previous case except for the fact that this experiment does not have the passivity based control task so the square shape of the formation is a bit different from the previous one. The initial position (four gray squares in the middle) of the four DJI Mavic 2 Pro quadcopters are shown in Fig. 5.19a. The four DJI Mavic 2 Pro quadcopters move in the attempt of reaching the desired goal represented by a gray square in the upper right corner, try to assemble the square shape, and following the human input (Figures 5.19b–5.19e), but cannot maintain the square formation and end up crashing

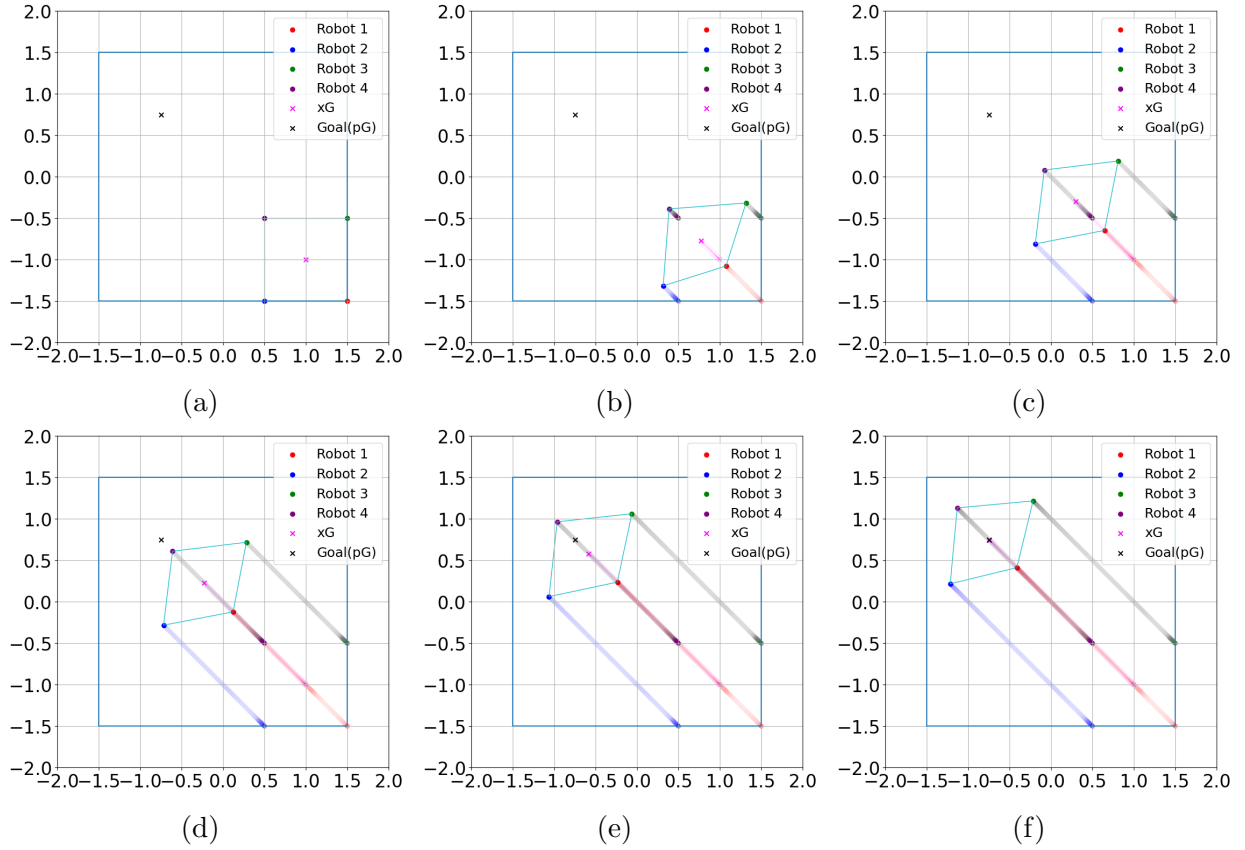


Figure 5.18: Snapshots of the simulations of centralized multi-robot multi-task execution without passivity and with human feedback implemented using Python. The scenario is similar to the one presented in the previous section. The robots attempt to assemble the desired square shape and move towards the goal (black cross (pG) at the upper left) but the shape of the formation is not a proper square like previous case and finally the center of the formed square reaches the desired goal (black cross (pG) at the upper left).

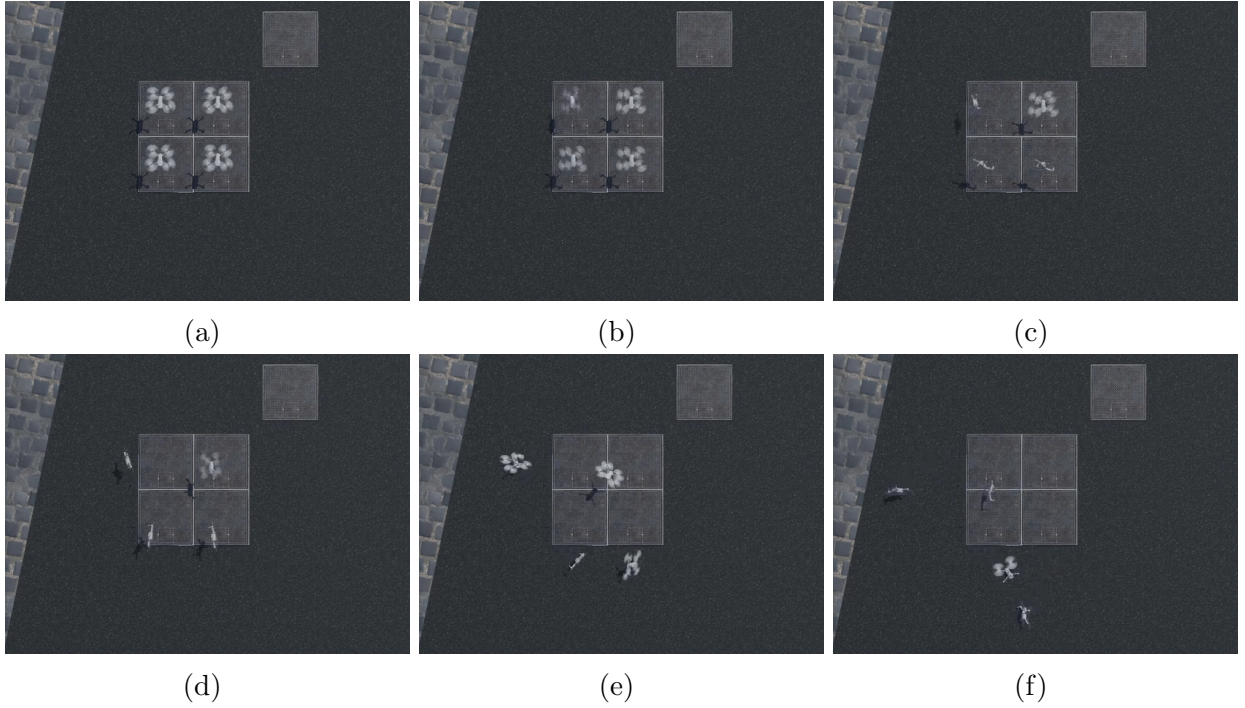


Figure 5.19: Snapshots of the experiments of centralized multi-robot multi-task execution without passivity and with human feedback performed in ROS2 humble using Webot as the simulator. The scenario is similar to the one presented in the previous section. The DJI Mavic 2 Pro quadcopters attempt to assemble the desired square shape and move towards the goal and but cannot maintain the square formation and end up crashing into each other and fall on the ground.

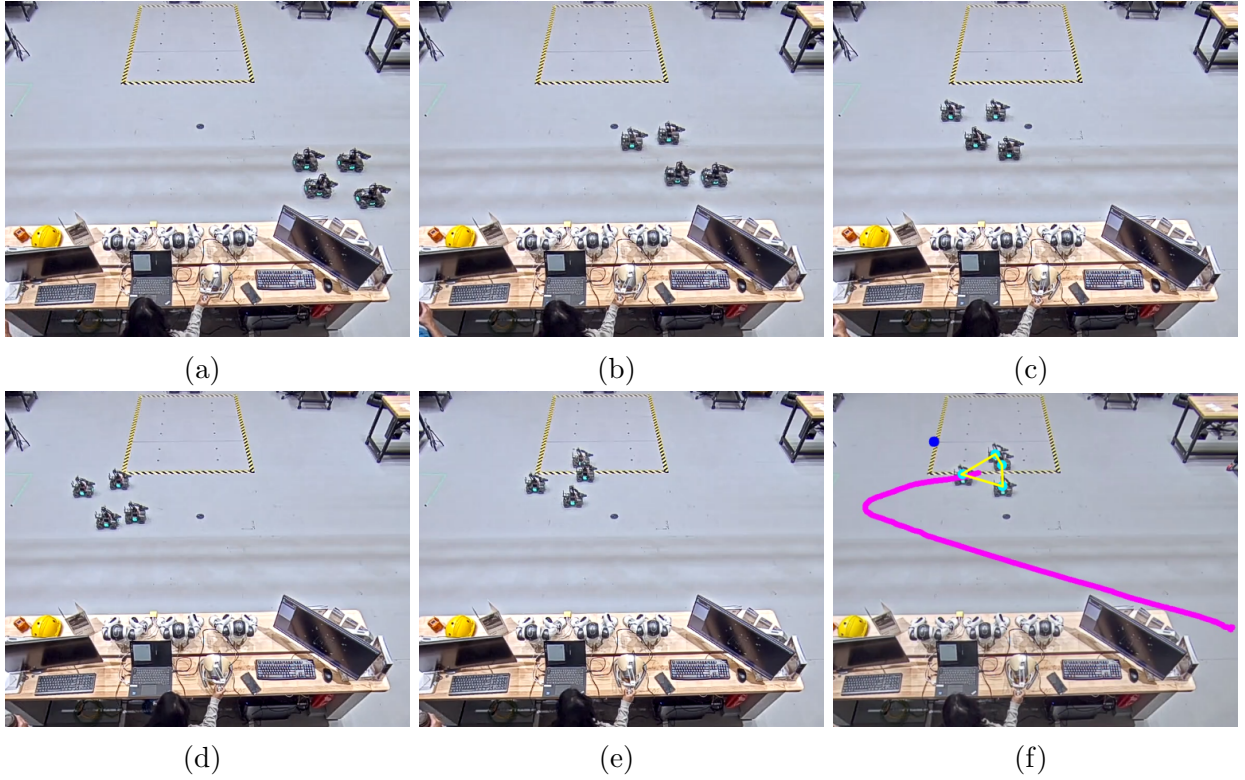


Figure 5.20: Snapshots of the experiments of centralized multi-robot multi-task execution without passivity and with human feedback performed in the Robohub at the University of Waterloo. The scenario is similar to the one presented in the previous section. The DJI Robomaster EP mobile platforms attempt to assemble the desired square shape and move towards the goal but cannot maintain the square formation and bump into each other.

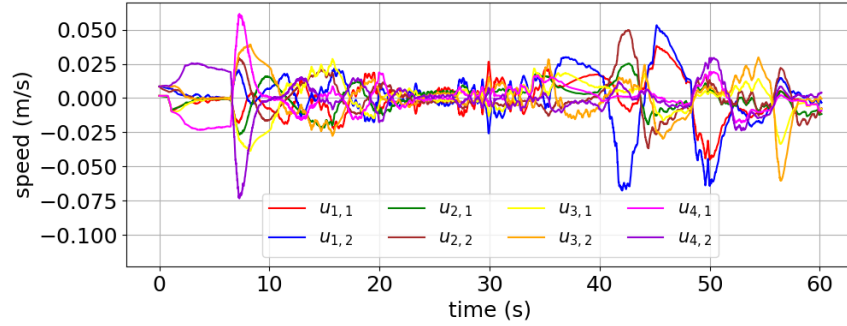


Figure 5.21: Control input vectors of the four robots with respect to time. As can be seen, the control inputs exhibits significantly larger and more erratic fluctuations, showing that the system becomes less stable and more prone to instability when passivity control is removed.

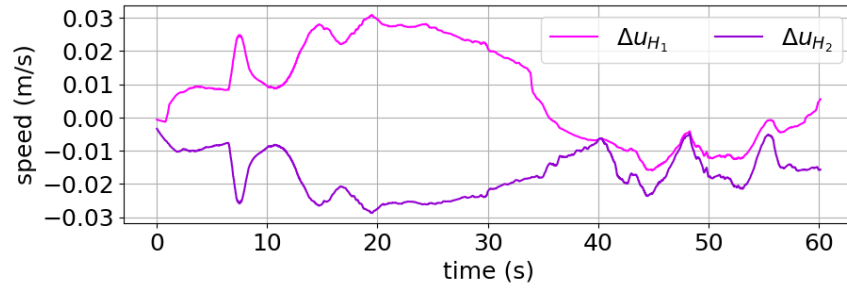


Figure 5.22: Human feedback with respect to time. Δu_H does not converge to 0, as a result there will be a non-zero haptic feedback.

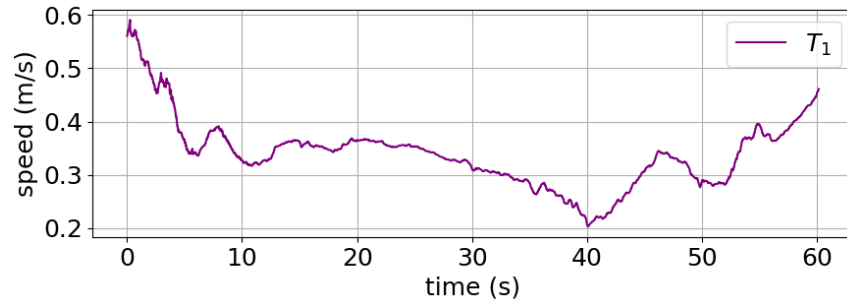


Figure 5.23: Formation graph of the four robots with respect to time. As can be seen, the smooth oscillations in T_1 indicate that the system dynamically adjusts the velocities of the robots to maintain the desired formation.

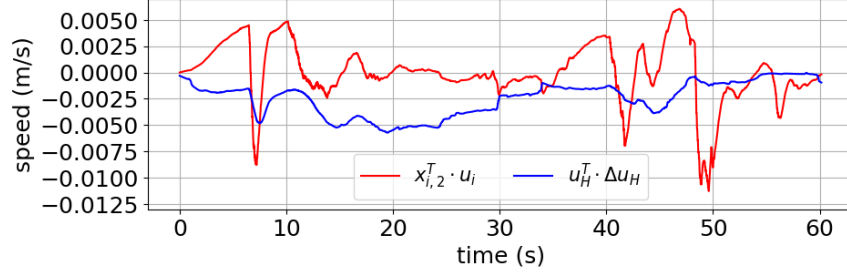


Figure 5.24: Passivity graph of the four robots with respect to time in case of absence of passivity. Without passivity control, the system may exhibit less predictable or more aggressive responses to human input, as indicated by the larger velocity fluctuations. This could imply that the system is more prone to instability or oscillations, especially if the human input is not smooth or consistent.

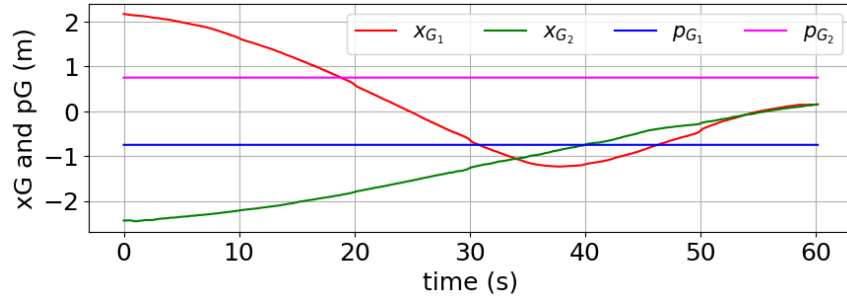


Figure 5.25: x_G and p_G with respect to time. As can be seen, the robot system, while adhering to passivity constraints, adjusts smoothly toward the goal.

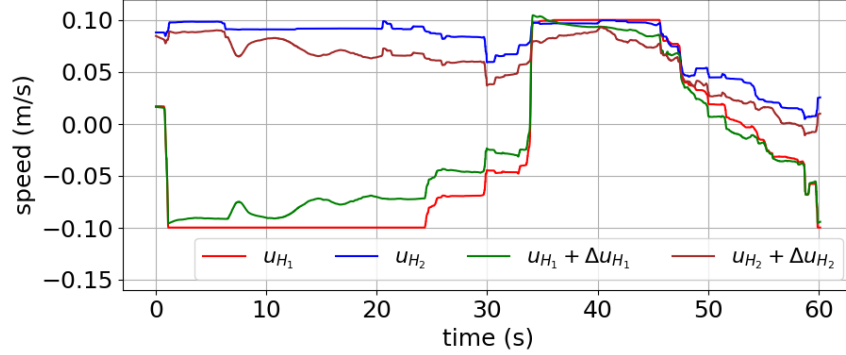


Figure 5.26: u_H and $u_H + \Delta u_H$ with respect to time. As can be seen, The difference between the u_H and $u_H + \Delta u_H$ lines indicates the magnitude of the adjustments made by the system. If the adjustments (Δu_H) are small, it suggests that the system is only making minor corrections to maintain stability. Larger deviations might indicate more significant interventions by the system to correct potential issues or instabilities.

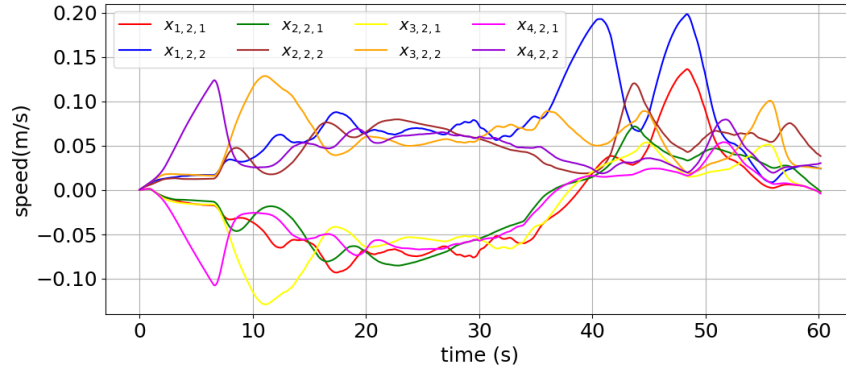


Figure 5.27: The evolution of the velocity components x_2 for each of the four robots over time. As can be seen, the velocities of all the robots start at certain initial values and evolve over time. The general trend of these lines indicates how each robot's velocity changes in response to the control inputs, including any adjustments made by the system to maintain formation and stability.

into each other (Figures 5.19c–5.19e), eventually the same failure to maintain the formation continues and the crashed quadcopters fall on the ground in their final configuration shown in Fig. 5.19f.

Figures 5.20a–5.20f display the different state of the experiment of multi-robot multi-task execution with human feedback synthesis in the Robohub at the University of Waterloo. The setup is similar to the previous case except for the fact that this experiment does not have the passivity based control task so the square shape of the formation is a bit different from the previous one. The initial position of the four DJI Robomaster EP are shown in Fig. 5.20a. The four DJI Robomaster EP move in the attempt of reaching the desired goal, try to assemble the square shape, and following the human input (Figures 5.20b–5.20c) but cannot maintain the square formation and bump into each other (Figures 5.20d–5.20e), eventually the same failure to maintain the formation continues in their final configuration shown in Fig. 5.20f.

In Fig. 5.21, we report the control inputs, u_i , of the four robots with respect to time. The two components of the control inputs of robot i is represented by $u_{i,1}(t)$ and $u_{i,2}(t)$. From the figure, we can see unlike the control inputs of passive scenario, here the control inputs exhibits significantly larger and more erratic fluctuations, showing that the system becomes less stable and more prone to instability when passivity control is removed. The larger deviations indicate a lack of energy regulation, which could lead to unsafe or unpredictable behavior in the robots. Figure 5.22 reports the human feedback, Δu_H , with respect to time. The two components of the human feedback is represented by $\Delta u_{H_1}(t)$ and $\Delta u_{H_2}(t)$. As can be seen, the feedback signals exhibit significantly larger and more erratic fluctuations, especially during the initial stages of the experiment. The larger magnitude of Δu_H suggests that the system is struggling to compensate for the mismatch between human input and robot behavior. Without passivity to regulate the interaction, the system experiences greater instability, as evidenced by the spikes in feedback around the 10 to 20-second mark. This erratic feedback not only makes it more difficult for the human operator to adjust their control but also leads to more frequent and severe corrections, which could cause frustration and reduce the effectiveness of the human-robot interaction.

The plot in Figure 5.23 demonstrates that without passivity control, the system experiences significant fluctuations in formation speed, particularly at the beginning and end of the experiment. The inability to regulate energy effectively results in more pronounced oscillations, making it challenging for the robot team to maintain a stable and coordinated formation. These fluctuations are less controlled than in the passivity-controlled scenario, leading to a less predictable and stable system overall.

Finally, Fig. 5.24 highlights the differences in system behavior when passivity control

is not enforced. The larger range of velocity fluctuations suggests that the system is more responsive but potentially less stable, especially under varying human input conditions. It implies that the system is not adequately regulating energy, which could cause the robot behaviors to deviate from the desired trajectories or become unstable under certain conditions. This graph serves as a key comparison point to understand the benefits of enforcing passivity in terms of system stability and control smoothness.

The positioning of the blue line above the red line in the passivity-controlled scenario reflects the system’s ability to effectively prioritize human input, ensuring smooth and stable operation with proper energy regulation. In contrast, the non-passivity scenario, where the red line is above the blue line, shows a system that is more reactive to its own internal control dynamics, resulting in larger fluctuations and potential instability. This comparison highlights the critical role of passivity-based control in maintaining a balance between human input and robot behavior, ensuring that the system operates stably while accurately following the human operator’s commands. The absence of passivity, however, leads to more aggressive system behavior, reducing the effectiveness of human input and potentially compromising the stability and safety of the multi-robot system.

Figure 5.25 demonstrates the challenges of maintaining precise and stable control in a system without passivity. The centroid positions x_{G_1} and x_{G_2} struggle to match the human input targets p_{G_1} and p_{G_2} . Figure 5.26 shows that the significant deviations between u_H and $u_H + \Delta u_H$ reflect the inability of the system to regulate energy and interaction effectively, making it more difficult for the human operator to control the robots. Figure 5.27 demonstrates that in the non-passivity scenario, the system struggles to maintain stable and synchronized robot velocities. The lack of energy regulation results in persistent and large fluctuations, which could lead to difficulties in performing tasks that require coordinated movement among the robots.

5.3 Discussions and Limitations

In this thesis, we have proposed a novel centralized and decentralized control framework for multi-robot, multi-task systems with human-robot interaction, including haptic feedback synthesis. The simulation and experimental results demonstrate the framework’s effectiveness in task allocation, formation control, and human feedback processing.

5.3.1 Discussions

One of the primary contributions of this thesis is the introduction of passivity-based control integrated with human-robot interaction. This mechanism ensures stability in scenarios where the robots receive unpredictable inputs from human operators. As evidenced in both the single-integrator and double-integrator dynamic systems, the robots were able to maintain formation and execute tasks while responding dynamically to human inputs.

In particular, the experimental results with double-integrator dynamics highlight how incorporating both position and velocity control leads to smoother robot trajectories and more stable interactions with the human operator. The human feedback, modeled through haptic feedback devices, provides real-time information about task performance, allowing the human operator to adjust their commands accordingly. This interaction ensures that the robots can respond more accurately and safely in dynamic environments.

Moreover, the simulations for both the single-integrator and double-integrator systems underscore the scalability of the proposed framework. Despite the inherent differences in complexity between the two systems, the optimization-based task allocation method allowed for efficient execution of tasks in both frameworks. Specifically, the single-integrator simulations demonstrated faster convergence in simpler task settings, while the double-integrator simulations provided smoother control in more complex environments involving human inputs and velocity regulation.

The inclusion of human feedback into the centralized control system highlights an advantage of adaptability in real-time scenarios. The system adjusts to unpredictable human instructions while ensuring the stability of the overall robot team. The feedback mechanism, particularly with passivity control, significantly improves the interaction by notifying the human operator about mismatches between desired and actual robot behavior. This feedback, while improving control, also reduces the likelihood of task execution failures.

5.3.2 Limitations

While the proposed framework has demonstrated significant advantages, there are limitations that merit consideration:

1. Scalability Issues in Centralized Control: One of the key challenges in implementing a centralized control approach is the scalability of the framework, particularly in larger, more complex multi-robot systems. As the number of robots increases, the computational complexity of the task allocation algorithm grows significantly.

While the centralized framework allows for precise coordination, it may become less practical in real-world scenarios involving hundreds of robots, where decentralized approaches are typically more efficient.

2. **Communication Overhead:** In both centralized and decentralized control frameworks, the communication overhead between robots and the human operator could present challenges in environments where communication bandwidth is limited or unstable. In particular, the decentralized system may struggle with real-time coordination if communication delays or packet losses occur, leading to suboptimal task execution.
3. **Limited Task Diversity:** The simulations primarily focused on formation control, go-to-goal tasks, and human-robot interaction constraints. While these tasks are essential, the proposed framework lacks more diverse tasks, including those requiring complex interactions between robots, such as object manipulation or cooperative transportation tasks.
4. **Feedback Mechanism Precision:** While the feedback mechanism provides valuable information about the mismatch between the robot’s behavior and human input, there is a limitation in the precision of this feedback. For more complex tasks that require precise coordination, the current feedback may not offer the level of detail necessary for the human operator to make nuanced adjustments in real time. Improvements in haptic feedback devices and feedback algorithms could help address this limitation.
5. **Experimental Environment Limitations:** The experimental validation was conducted using a small number of robots in a controlled environment. While this is sufficient for proof-of-concept demonstrations, the framework has not been tested in large-scale, dynamic environments with real-world uncertainties, such as outdoor search and rescue missions or exploration in unknown terrain. It lacks the validation of the framework in more realistic and challenging environments.
6. **Energy Constraints Not Considered:** While task allocation efficiency was a focus of this work, energy constraints for individual robots were not directly addressed. In long-term operations or energy-limited environments, managing the energy consumption of robots is crucial for task sustainability. This framework lacks incorporation of energy-awareness mechanisms to ensure that robots are not depleted of power during critical tasks.
7. **Human Input Variability:** The system’s performance depends significantly on the smoothness and accuracy of the human operator’s input. While passivity-based control mitigates stability concerns, unexpected or highly erratic human input can still

lead to suboptimal performance. Human operators, by nature, can and often will provide inputs that deviate from expectations, especially in dynamic and high-stress environments. A user study could provide valuable insights into the robustness and usability of the proposed control scheme. Metrics such as task completion time, operator workload, and perceived ease of use could be used to evaluate system performance under various levels of input variability. As a potential future direction, incorporating real-world user feedback through such a study could guide refinements to the control framework. During preliminary experiments, operating the system with varying input patterns highlighted the intuitive feedback provided by the haptic device, although erratic inputs occasionally necessitated additional operator corrections. This reinforces the need for a systematic user study to fully assess the system's adaptability and usability in diverse scenarios.

Chapter 6

Conclusions

This thesis has explored the integration of decentralized and centralized control frameworks for multi-robot, multi-task systems with human interaction. The proposed framework utilizes both single-integrator and double-integrator dynamic models, emphasizing the role of human feedback in guiding the robot formation and task execution. Through simulations and real-world experiments, the results demonstrated the effectiveness of the optimization-based task allocation framework that ensures stability, passivity, and real-time responsiveness to human input.

The integration of passivity-based control was a significant contribution, allowing the system to maintain stability while interacting with human operators through haptic feedback. The experiments confirmed that both the decentralized and centralized frameworks successfully handled multi-task execution with human feedback, while the use of slack variables facilitated task prioritization.

The framework proved particularly useful for tasks like formation control, human-robot interaction, and coverage control. The decentralized approach ensured scalability in single-integrator dynamic systems, while the centralized control was more effective in managing complex double-integrator systems with acceleration feedback.

Several directions can be pursued to extend the current work:

1. Improving Human-Robot Interaction: Future research could focus on refining the feedback mechanisms provided to human operators, including more intuitive haptic feedback systems and adaptive learning algorithms to predict and adjust to human input more effectively.

2. **Dynamic Environments:** Extending the current work to dynamically changing environments where robots must adapt to moving obstacles, variable goals, and uncertain conditions in real-time is an essential area for future investigation.
3. **Scalability and Distributed Control:** A promising area of research is the development of hybrid control frameworks that combine centralized and decentralized approaches, enabling the system to handle larger and more distributed teams of robots while reducing computational overhead.
4. **Safety and Fault Tolerance:** Incorporating more advanced safety mechanisms, such as control barrier functions (CBFs) and fault-tolerant algorithms, could make the system more robust against unexpected failures or communication breakdowns during human-robot interaction.
5. **Application in Real-World Scenarios:** Finally, validating the framework in more real-world applications, such as autonomous warehouse management, search and rescue, or industrial automation, could provide deeper insights into its practical utility and performance under real-time constraints.

This research opens up several possibilities for future development in the field of multi-robot systems and human-robot collaboration, paving the way for more effective and resilient robot teams capable of working in dynamic and unpredictable environments.

References

- [1] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Force control*. Springer, 2009.
- [2] Robin R Murphy and Jennifer L Burke. The safe human-robot ratio. In *Human-robot interactions in future military operations*, pages 31–50. CRC Press, 2016.
- [3] Robin R Murphy. Human-robot interaction in rescue robotics. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 34(2):138–153, 2004.
- [4] J Edward Colgate, Witaya Wannasuphoprasit, and Michael A Peshkin. Cobots: Robots for collaboration with human operators. In *ASME international mechanical engineering congress and exposition*, volume 15281, pages 433–439. American Society of Mechanical Engineers, 1996.
- [5] Brian P Gerkey and Maja J Matarić. A formal analysis and taxonomy of task allocation in multi-robot systems. *The International journal of robotics research*, 23(9):939–954, 2004.
- [6] Keiji Nagatani, Yoshito Okada, Naoki Tokunaga, Seiga Kiribayashi, Kazuya Yoshida, Kazunori Ohno, Eijiro Takeuchi, Satoshi Tadokoro, Hidehisa Akiyama, Itsuki Noda, et al. Multirobot exploration for search and rescue missions: A report on map building in robocuprescue 2009. *Journal of Field Robotics*, 28(3):373–387, 2011.
- [7] Maria Valera Espina, Raphael Grech, Deon De Jager, Paolo Remagnino, Luca Iocchi, Luca Marchetti, Daniele Nardi, Dorothy Monekosso, Mircea Nicolescu, and Christopher King. Multi-robot teams for environmental monitoring. *Innovations in Defence Support Systems-3: Intelligent Paradigms in Security*, pages 183–209, 2011.

- [8] Leighton Collins, Payam Ghassemi, Ehsan T Esfahani, David Doermann, Karthik Dantu, and Souma Chowdhury. Scalable coverage path planning of multi-robot teams for monitoring non-convex areas. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7393–7399. IEEE, 2021.
- [9] Edwin Olson, Johannes Strom, Ryan Morton, Andrew Richardson, Pradeep Ranganathan, Robert Goeddel, Mihai Bulic, Jacob Crossman, and Bob Marinier. Progress toward multi-robot reconnaissance and the magic 2010 competition. *Journal of Field Robotics*, 29(5):762–792, 2012.
- [10] Gennaro Notomista, Siddharth Mayya, Seth Hutchinson, and Magnus Egerstedt. An optimal task allocation strategy for heterogeneous multi-robot systems. In *2019 18th European Control Conference (ECC)*, pages 2071–2076. IEEE, 2019.
- [11] Jorge Cortes, Sonia Martinez, Timur Karatas, and Francesco Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on robotics and Automation*, 20(2):243–255, 2004.
- [12] Nathan Michael, Jonathan Fink, and Vijay Kumar. Cooperative manipulation and transportation with aerial robots. *Autonomous Robots*, 30:73–86, 2011.
- [13] Gennaro Notomista. Stable, safe, and passive teleoperation of multi-robot systems. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11540–11546. IEEE, 2024.
- [14] Gennaro Notomista and Xiaoyi Cai. A safety and passivity filter for robot teleoperation systems. In *Human-Friendly Robotics 2020: 13th International Workshop*, pages 101–115. Springer, 2021.
- [15] Gennaro Notomista. A constrained-optimization approach to the execution of prioritized stacks of learned multi-robot tasks. In *International Symposium on Distributed Autonomous Robotic Systems*, pages 479–493. Springer, 2022.
- [16] Ramisha Anjum and Gennaro Notomista. Optimization-based haptic feedback synthesis for passive human-multi-robot systems. *Submitted to IFAC Mechatronics*, 2025.
- [17] Gennaro Notomista, Siddharth Mayya, Yousef Emam, Christopher Kroninger, Addison Bohannon, Seth Hutchinson, and Magnus Egerstedt. A resilient and energy-aware task allocation framework for heterogeneous multirobot systems. *IEEE Transactions on Robotics*, 38(1):159–179, 2021.

- [18] Gennaro Notomista and Magnus Egerstedt. Constraint-driven coordinated control of multi-robot systems. In *2019 American control conference (ACC)*, pages 1990–1996. IEEE, 2019.
- [19] Yousef Emam, Siddharth Mayya, Gennaro Notomista, Addison Bohannon, and Magnus Egerstedt. Adaptive task allocation for heterogeneous multi-robot teams with evolving and unknown robot capabilities. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7719–7725. IEEE, 2020.
- [20] Haris Aziz, Hau Chan, Ágnes Cseh, Bo Li, Fahimeh Ramezani, and Chenhao Wang. Multi-robot task allocation—complexity and approximation. *arXiv preprint arXiv:2103.12370*, 2021.
- [21] Payam Ghassemi and Souma Chowdhury. Multi-robot task allocation in disaster response: Addressing dynamic tasks with deadlines and robots with range and payload constraints. *Robotics and Autonomous Systems*, 147:103905, 2022.
- [22] Qiuzhen Wang and Xinjun Mao. Dynamic task allocation method of swarm robots based on optimal mass transport theory. *Symmetry*, 12(10):1682, 2020.
- [23] Arne Brutschy, Giovanni Pini, Carlo Pinciroli, Mauro Birattari, and Marco Dorigo. Self-organized task allocation to sequentially interdependent tasks in swarm robotics. *Autonomous agents and multi-agent systems*, 28:101–125, 2014.
- [24] Zhangchi Ding, Masoud Baghbahari, and Aman Behal. A passivity-based framework for safe physical human–robot interaction. *Robotics*, 12(4):116, 2023.
- [25] Ran Cao, Long Cheng, and Houcheng Li. Passive model-predictive impedance control for safe physical human–robot interaction. *IEEE Transactions on Cognitive and Developmental Systems*, 16(2):426–435, 2023.
- [26] Lingzhi Luo, Nilanjan Chakraborty, and Katia Sycara. Distributed algorithms for multirobot task assignment with task deadline constraints. *IEEE Transactions on Automation Science and Engineering*, 12(3):876–888, 2015.
- [27] Yuxiao Chen, Ugo Rosolia, and Aaron D Ames. Decentralized task and path planning for multi-robot systems. *IEEE Robotics and Automation Letters*, 6(3):4337–4344, 2021.
- [28] Gennaro Notomista. Resilience and energy-awareness in constraint-driven-controlled multi-robot systems. In *2022 American Control Conference (ACC)*, pages 3682–3687. IEEE, 2022.

- [29] Tina Setter and Magnus Egerstedt. Energy-constrained coordination of multi-robot teams. *IEEE Transactions on Control Systems Technology*, 25(4):1257–1263, 2016.
- [30] Pau Segui-Gasco, Hyo-Sang Shin, Antonios Tsourdos, and VJ Segui. Decentralised submodular multi-robot task allocation. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2829–2834. IEEE, 2015.
- [31] Ehsan Latif, Yikang Gui, Aiman Munir, and Ramviyas Parasuraman. Energy-aware multi-robot task allocation in persistent tasks. *arXiv preprint arXiv:2112.15282*, 2021.
- [32] Lovekesh Vig and Julie A Adams. Multi-robot coalition formation. *IEEE transactions on robotics*, 22(4):637–649, 2006.
- [33] Tamás Keviczky, Francesco Borrelli, and Gary J Balas. Decentralized receding horizon control for large scale dynamically decoupled systems. *Automatica*, 42(12):2105–2115, 2006.
- [34] Javier Alonso-Mora, Eduardo Montijano, Tobias Nägele, Otmar Hilliges, Mac Schwager, and Daniela Rus. Distributed multi-robot formation control in dynamic environments. *Autonomous Robots*, 43:1079–1100, 2019.
- [35] Robert J Anderson and Mark W Spong. Bilateral control of teleoperators with time delay. In *Proceedings of the 1988 IEEE International Conference on Systems, Man, and Cybernetics*, volume 1, pages 131–138. IEEE, 1988.
- [36] Craig B Zilles and J Kenneth Salisbury. A constraint-based god-object method for haptic display. In *Proceedings 1995 ieee/rsj international conference on intelligent robots and systems. Human robot interaction and cooperative robots*, volume 3, pages 146–151. IEEE, 1995.
- [37] Richard J Adams and Blake Hannaford. Stable haptic interaction with virtual environments. *IEEE Transactions on robotics and Automation*, 15(3):465–474, 1999.
- [38] Allison M Okamura. Methods for haptic feedback in teleoperated robot-assisted surgery. *Industrial Robot: An International Journal*, 31(6):499–508, 2004.
- [39] Lars Johannessmeier and Sami Haddadin. A hierarchical human-robot interaction-planning framework for task allocation in collaborative industrial assembly processes. *IEEE Robotics and Automation Letters*, 2(1):41–48, 2016.
- [40] Maja J Matarić, Gaurav S Sukhatme, and Esben H Østergaard. Multi-robot task allocation in uncertain environments. *Autonomous Robots*, 14:255–263, 2003.

- [41] Albert Hernansanz, Alicia Casals, and Josep Amat. A multi-robot cooperation strategy for dexterous task oriented teleoperation. *Robotics and Autonomous Systems*, 68:156–172, 2015.
- [42] Brian E Miller, J Edward Colgate, and Randy A Freeman. Guaranteed stability of haptic systems with nonlinear virtual environments. *IEEE Transactions on Robotics and Automation*, 16(6):712–719, 2000.
- [43] Jiayu Li, Bo You, Liang Ding, Xiaoyang Yu, Weihua Li, Tianyong Zhang, and Haibo Gao. Dual-master/single-slave haptic teleoperation system for semiautonomous bilateral control of hexapod robot subject to deformable rough terrain. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(4):2435–2449, 2021.
- [44] Zhengyu Wang, Mingxin Hai, Xuchang Liu, Zongkun Pei, Sen Qian, and Daoming Wang. A human–robot interaction control strategy for teleoperation robot system under multi-scenario applications. *International Journal of Intelligent Robotics and Applications*, pages 1–21, 2024.
- [45] Takeshi Hatanaka, Takahiro Mochizuki, Takumi Sumino, José M Maestre, and Nikhil Chopra. Human modeling and passivity analysis for semi-autonomous multi-robot navigation in three dimensions. *IEEE Open Journal of Control Systems*, 2023.
- [46] Dominik Sieber, Selma Musić, and Sandra Hirche. Multi-robot manipulation controlled by a human with haptic feedback. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2440–2446. IEEE, 2015.
- [47] Mahya Shahbazi, Seyed Farokh Atashzar, Mahdi Tavakoli, and Rajni V Patel. Position-force domain passivity of the human arm in telerobotic systems. *IEEE/ASME Transactions on Mechatronics*, 23(2):552–562, 2018.
- [48] Wenceslao Shaw Cortez, Christos K Verginis, and Dimos V Dimarogonas. Safe, passive control for mechanical systems with application to physical human-robot interactions. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3836–3842. IEEE, 2021.
- [49] Paolo Robuffo Giordano, Antonio Franchi, Cristian Secchi, and Heinrich H Bühlhoff. Bilateral teleoperation of groups of uavs with decentralized connectivity maintenance. In *Robotics: Science and Systems*, 2011.
- [50] Riku Funada, Xiaoyi Cai, Gennaro Notomista, Made Widhi Surya Atman, Junya Yamauchi, Masayuki Fujita, and Magnus Egerstedt. Coordination of robot teams over

long distances: From georgia tech to tokyo tech and back-an 11,000-km multirobot experiment. *IEEE Control Systems Magazine*, 40(4):53–79, 2020.

- [51] Gennaro Notomista, Xiaoyi Cai, Junya Yamauchi, and Magnus Egerstedt. Passivity-based decentralized control of multi-robot systems with delays using control barrier functions. In *2019 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, pages 231–237. IEEE, 2019.
- [52] Antonio Franchi, Cristian Secchi, Hyoungh Il Son, Heinrich H Bulthoff, and Paolo Robuffo Giordano. Bilateral teleoperation of groups of mobile robots with time-varying topology. *IEEE Transactions on Robotics*, 28(5):1019–1033, 2012.
- [53] Dawei Zhang and Roberto Tron. Safe and stable teleoperation of quadrotor uavs under haptic shared autonomy. *arXiv preprint arXiv:2403.15335*, 2024.
- [54] Shaobo Zhang, Yi Chen, Jun Zhang, and Yunyi Jia. Real-time adaptive assembly scheduling in human-multi-robot collaboration according to human capability. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3860–3866. IEEE, 2020.
- [55] Giacomo Lunghi, Raul Marin, Mario Di Castro, Alessandro Masi, and Pedro J Sanz. Multimodal human-robot interface for accessible remote robotic interventions in hazardous environments. *IEEE Access*, 7:127290–127319, 2019.
- [56] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [57] Dale A Lawrence. Stability and transparency in bilateral teleoperation. *IEEE transactions on robotics and automation*, 9(5):624–637, 1993.
- [58] H.K. Khalil. *Nonlinear Systems*. Pearson Education. Prentice Hall, 2002.
- [59] M Bernardine Dias, Robert Zlot, Nidhi Kalra, and Anthony Stentz. Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE*, 94(7):1257–1270, 2006.
- [60] Danfeng Wu, Guangping Zeng, Lingguo Meng, Weijian Zhou, and Linmin Li. Gini coefficient-based task allocation for multi-robot systems with limited energy resources. *IEEE/CAA Journal of Automatica Sinica*, 5(1):155–168, 2017.

- [61] José Manuel Palacios-Gasós, Zeynab Talebpour, Eduardo Montijano, Carlos Sagüés, and Alcherio Martinoli. Optimal path planning and coverage control for multi-robot persistent coverage in environments with obstacles. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1321–1327. IEEE, 2017.
- [62] Walker Gosrich, Siddharth Mayya, Rebecca Li, James Paulos, Mark Yim, Alejandro Ribeiro, and Vijay Kumar. Coverage control in multi-robot systems via graph neural networks. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8787–8793. IEEE, 2022.
- [63] 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.
- [64] Olivier Michel. Cyberbotics ltd. webotsTM: professional mobile robot simulation. *International Journal of Advanced Robotic Systems*, 1(1):5, 2004.
- [65] Steven Martin and Nick Hillier. Characterisation of the novint falcon haptic device for application as a robot manipulator. In *Australasian Conference on Robotics and Automation (ACRA)*, pages 291–292. Citeseer, 2009.

APPENDICES

Appendix A

Matlab Simulation of Different Tasks

A.1 Multi-Robot Multi-Task Execution with Only Go-to-Goal Task and Coverage Control

In this scenario of multi-robot multi-task execution, there are two robots: the highest priority for robots 1 is to move in the direction of its own voronoi cell's centroid, while for robot 2 the go-to-goal task has highest priority. MATLAB simulation has been shown for this scenario. In summary, the task constraints for the robots are:

1. Proceeding in the direction of the predefined goal (task T_1)
2. Proceeding in the direction of its own voronoi cell's centroid (task T_4)

whereas their priorities are ($T_i \prec T_j$ means that task i has higher priority than task j):

1. Robots 1: $T_4 \prec T_1$
2. Robot 2: $T_1 \prec T_4$

In Fig. [A.1](#), Robot 1 (R_1) is travelling in the direction of its own Voronoi cell's Centroid (G_1), whereas Robot 2 (R_2) is moving in the direction of a predetermined goal (P_1). Here, the optimization algorithm decides beforehand which robot will carry out which task. Robots are prioritizing that duty as a result. Here, robot 1 (R_1) is working on task (T_4), and robot 2 (R_2) is working on task (T_1).

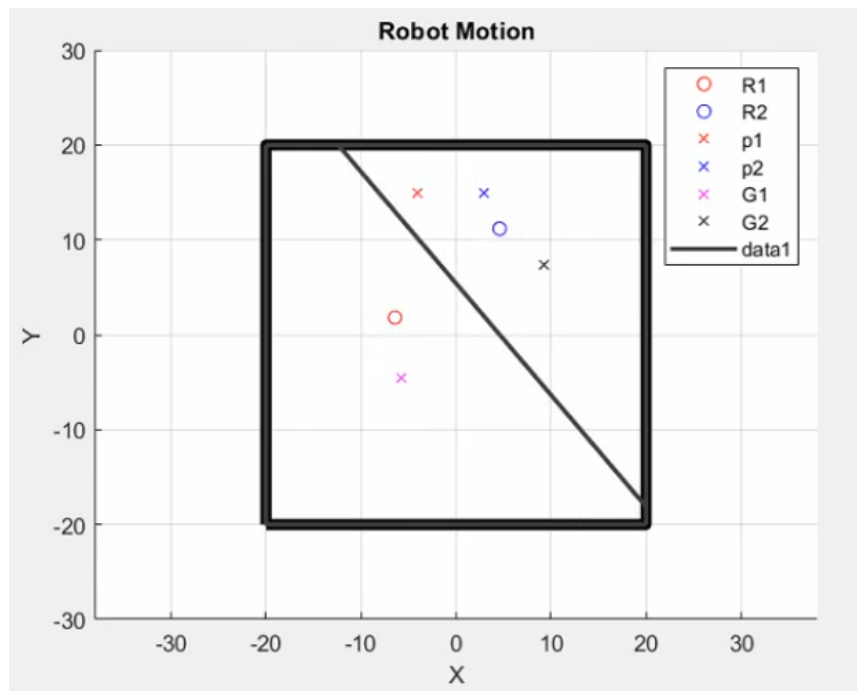


Figure A.1: Simulation of robot 1 (R_1) going to its own voronoi cell's centroid G_1 (T_4) and robot 2 (R_2) going to goal P_2 (T_1).

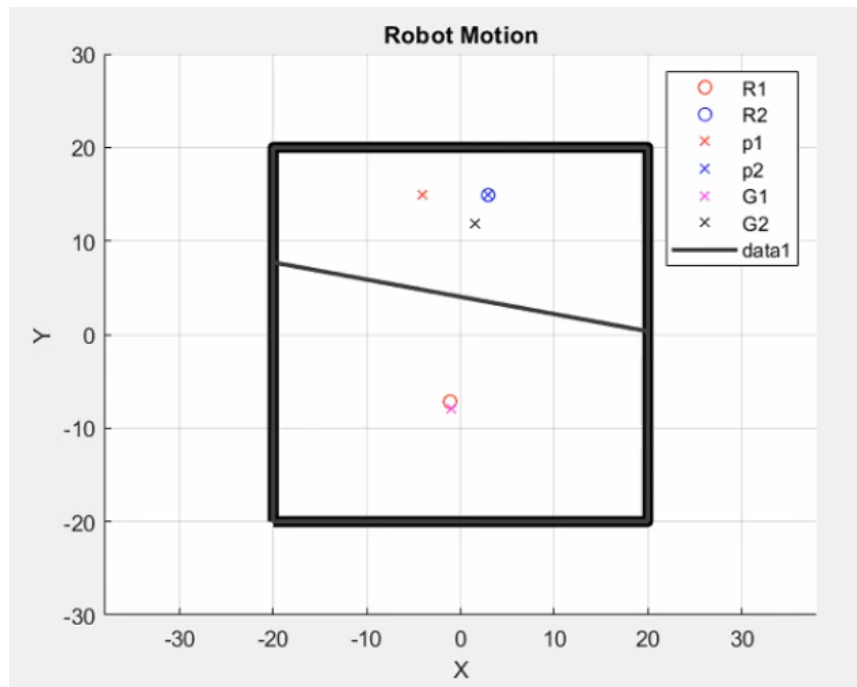


Figure A.2: Simulation of robot 1 (R_1) after reaching its own voronoi cell's centroid G_1 (T_4) and robot 2 (R_2) after reaching goal P_2 (T_1).

Robot 1 (R_1) reached its own voronoi cell's centroid (G_1) in Fig. A.2, and robot 2 (R_2) attained the predetermined goal (P_2). If their predefined priorities are set differently, they can both do the same task or exchange tasks in a different scenario.

Appendix B

Python Simulation of Different Tasks

B.1 Multi-Robot Multi-Task Execution with Only Go-to-Goal Task

In this scenario of multi-robot multi-task execution, there are four robots but they have four different goals: the highest priority for robots 1, 2, 3, and 4 is the go-to-goal task but the goal for the four robots are different and the priority depends on which goal has been set to which robot. Robot 1 only goes towards its own predefined goal and other three goals are not a highest priority to robot 1 and similar case is true for the remaining three robots. Python simulation has been shown for this scenario. In summary, the task constraints for the robots are:

1. Proceeding in the direction of the predefined goal (task T_1)

Figure [B.1](#) displays the multi-robot multi-task execution. Robots 1, 2, 3, and 4 are represented by red, green, blue, and brown circles, respectively. The predefined goals are represented by the red, green, blue, and brown crosses respectively. In this figure, robots 1, 2, 3, and 4 move towards its own predefined goals (P_1 , P_2 , P_3 , and P_4) respectively. The following controller parameters have been chosen for this simulation: $k = 0.001$, $K = 1000$, $\gamma_1 = 10$, $\gamma_2 = 10$, $\gamma_3 = 10$, and $\gamma_4 = 10$.

Figure [B.2](#) displays the different states of multi-robot multi-task execution performed using the Webot simulator, where four DJI Mavic 2 Pro quadcopters are simulated. The predetermined goals are represented by four gray squares in the upper left, upper right,

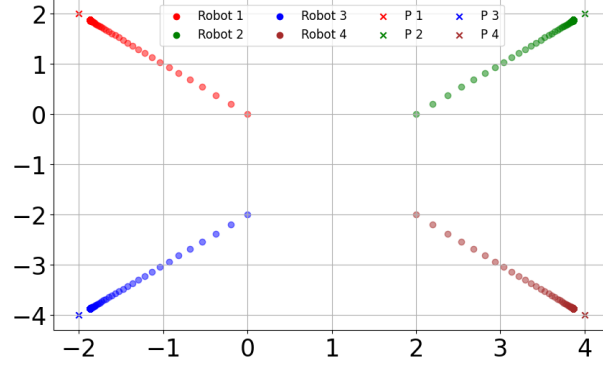


Figure B.1: Multi-robot multi-task execution in Python. The four robots (Robot 1, Robot 2, Robot 3, and Robot 4) are represented by circles of four colors (red, green, blue, and brown, respectively). The red, green, blue, and brown cross marks depict four designated goal locations respectively. The robots move towards its own predefined goals (highest priority task for robots 1, 2, 3, and 4).

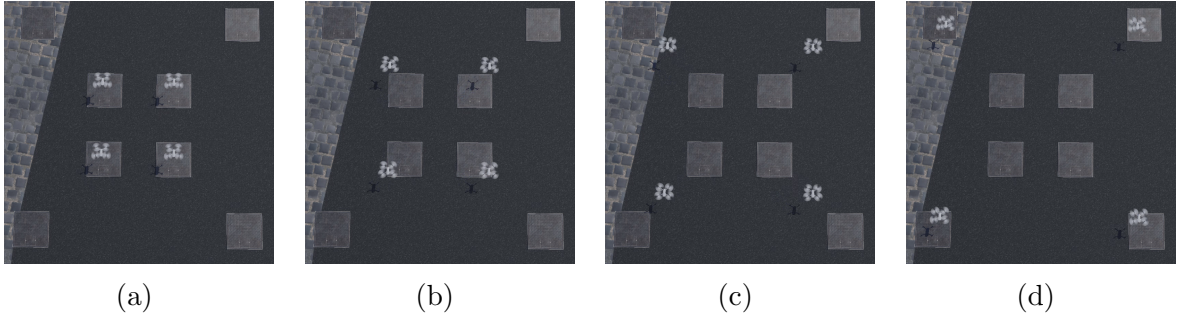


Figure B.2: Snapshots of the simulations performed in Webot implemented using ROS2 humble. The pictures show four DJI Mavic 2 Pro quadcopters starting from their initial positions represented by the four gray square in the middle (Fig. B.2a), moving in the environment towards their own predefined goal locations represented by gray square in the upper left corner, gray square in the upper right corner, gray square in the bottom left corner, and gray square in the bottom right corner respectively (Figures B.2b, B.2c), and reaching their goal finally (Fig. B.2d).

bottom left, and bottom right corner respectively, and the initial positions of the four quadcopters are represented by the four gray squares in the middle. The initial position of the four quadcopters are shown in Fig. [B.2a](#). Then, Figures [B.2b](#)–[B.2c](#) show the positions of the drones over the course of the experiment, moving towards their own goals; finally, in Fig. [B.2d](#), the four quadcopters reach their own goals.