HUMAN-IN-THE-LOOP CONTROL FOR COOPERATIVE HUMAN-ROBOT TASKS

A Thesis
Presented to
The Academic Faculty

by

Rahul Chipalkatty

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in Robotics in the
School of Mechanical Engineering

Georgia Institute of Technology
May 2012

HUMAN-IN-THE-LOOP CONTROL FOR COOPERATIVE HUMAN-ROBOT TASKS

Approved by:

Magnus Egerstedt, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Wayne Book, Co-Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Ayanna Howard
School of Electrical and Computer Engineering
Georgia Institute of Technology

Jun Ueda
School of Mechanical Engineering
Georgia Institute of Technology

Nader Sadegh
School of Mechanical Engineering
Georgia Institute of Technology

Date Approved: March 12, 2012

To my mother, father, and sister for their unwavering love and support.

ACKNOWLEDGEMENTS

I would like the acknowledge my advisor, Dr. Magnus Egerstedt, who was the perfect advisor to guide me through this degree. His enthusiasm and vision for our research constantly pushed my work and what I expected of it. I'd also like to thank my co-advisor, Dr. Wayne Book, who guided me in applying my research to robotics problems and generously included me in his group. I would also like to thank the members of my dissertation committee: Dr. Ayanna Howard, Dr. Jun Ueda, and Dr. Nader Sadegh. I thank Dr. Howard for helping me navigate the murky waters of this new Robotics PhD program.

I would also like to thank Dr. Amir Rahmani for always being there when I needed advice regarding research and things unrelated to research. Also, I owe a large thanks to Jean-Pierre De la Croix for answering all my questions (large or small) regarding software implementations. I would like to acknowledge Hannes Daepp for staying long hours with me in the lab in order to get the rescue robot code working with my research. I am also grateful for my conversations with Dr. Brian Levinthal and Tiffany Chen who patiently helped me learn what it takes to run and analyze a human operator study. I'd like to thank Dr. Axel Schild and Dr. Dennis Ding for insights on model predictive control. I also want to thank Dr. Patrick Martin, Dr. Musad Haque, Philip Twu, and Peter Kingston (who was always the human operator in my videos) for always taking the time to listen to my ideas when I needed someone to bounce them off of. I would also like to thank Greg Droge, Amy Laviers, Hassan Jaleel, Waseem Abbas, Smriti Chopra and the rest of the GRITS lab. The IMDL lab was also instrumental in furthering my research with great discussions on the more mechanical aspects of robotics. In particular, I'd like to thank Dr. Aaron Enes, Brian Post, Ryder Wync, Heather Humphries, and JD Huggins.

Finally, I'd like to thank my family, who has supported me in every way throughout my life, but especially during my time as a PhD student. I also am thankful for all my friends from childhood, college, UIUC, and all the fantastic people I have met here in Atlanta

and Georgia Tech. Without your encouragement and inspiration, this would not have been possible.

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

Even with the advance of autonomous robotics and automation, many automated tasks still require human intervention or guidance to mediate uncertainties in the environment or to execute the complexities of a task that autonomous robots are not yet equipped to handle. As such, robot controllers are needed that utilize the strengths of both autonomous agents, adept at handling lower level control tasks, and humans, superior at handling higher-level cognitive tasks.

To address this need, we develop a control theoretic framework that seeks to incorporate user commands such that user intention is preserved while an automated task is carried out by the controller. This is a novel approach in that system theoretic tools allow for analytic guarantees of feasibility and convergence to goal states which naturally lead to varying levels of autonomy. We develop a model predictive controller that takes human input, infers human intent, then applies a control that minimizes deviations from the intended human control while ensuring that the lower-level automated task is being completed.

This control framework is then evaluated in a human operator study involving a shared control task with human guidance of a mobile robot for navigation. These theoretical and experimental results lay the foundation for applying this control method for human-robot cooperative control to actual human-robot tasks. Specifically, the control is applied to a Urban Search and Rescue robot task where the shared control of a quadruped rescue robot is needed to ensure static stability during human-guided leg placements in uneven terrain. This control framework is also extended to a multiple user and multiple agent system where the human operators control multiple agents such that the agents maintain a formation while allowing the human operators to manipulate the shape of the formation. User studies are also conducted to evaluate the control in multiple operator scenarios.

CHAPTER I

INTRODUCTION

Despite advances in autonomous robotics and automation, some tasks still require human intervention or guidance to mediate uncertainties in the environment or to manage the complexities of a task that autonomous robots are not yet equipped to handle. Therefore, it is desirable to design robot controllers that utilize the strengths of both autonomous agents, adept at handling lower level control tasks, and humans, superior at handling higher-level cognitive tasks.

Autonomous agents excel at performing repetitive tasks and precise movements without fatigue, and advances in autonomy have afforded robots with abilities such as navigating corridors and avoiding obstacles. Tasks such as "go-to-goal" or placing objects in precise locations are regularly performed by robots. However, perceptive and cognitive heavy tasks are still a weak area in robotic capabilities. Creating robots that can navigate uneven complex terrain or determining what the goal in a "go-to-goal" behavior is are not easy tasks or often infeasible to program into robots. However, humans are already quite good at these tasks and are often present during robotic operations in order to ensure safe operation and troubleshooting. So, developing control structures that take advantage of both robot and human capabilities is a way to push the capabilities of robotic systems that lack the required cognitive intelligence and is a shrewd use of resources.

Therefore, in order to take advantage of both human operator and automatic controller strengths, a controller is required that allows a user to give commands that accomplish a high-level cognitive task while the controller simultaneously attempts to honor the intention of the human and complete a low-level control task. Schemes to compose these two commands (such as in [9, 72]) have been proposed, but can either high-level or low-level task completion be guaranteed? This thesis will present a control framework on how to combine human and robot control input in an effective way. Inherent in such control schemes is

the need for the notion of human intent (e.g. [86]) to encode high-level task completion commands, i.e. what does the human want the system to do, and the ability to continually update this notion as the operator commands evolve.

Applications, where these types of controllers can be of benefit, range from human interaction with rescue robots [65], assisted human control of mining vehicles [82], intelligent mobility devices [18], to robotic-assisted surgery [66]. In each of these applications, there are elements where automation can lead to safer, more efficient operation while human operators can perform high-level decision-making and cognitive tasks. An important research question in these applications is: how does one design controllers that take into account both user inputs as well as automated commands while still leading to task completion for both user and automaton.

In response, human-robot interaction (HRI) and cooperation is extensively studied in Urban Search and Rescue Scenarios (USAR) as in [64]. Here, identification of victims in highly complex disaster areas is a difficult task to program autonomous robots to accomplish reliably, however, humans are quite adept at this task. On the other hand, it is advantagous to send a robot into disaster areas as myriad dangers are posed to any humans entering them, while damage to or the destruction of robots is an acceptable consequence. Additional advantages include the fact that a robot can be designed to fit through spaces a human rescue worker could not. Complete teleoperation is also difficult in that visual or haptic cues are not enough to operate robots in such complex environments, leading to failures like fallen or trapped robots. As such, USAR applications are a prime reseasrch area where coordinated human-robot task completion would make an impact.

Similarly, shared control architectures can also be applied to cooperative human-robot control and navigation of heavy machinery in mining, construction, and deforestation industries. The tasks associated with operating these vehicles are repeititive, yet require high cognitive and physical workloads [82, 78, 27], so control schemes where partial control of tasks is alleviated from the operator while still allowing high-level task control are desired. An example is mine navigation for mining vehicles, where operators often drive large vehicles in low-visibility conditions and work hours are often limited by cognitive and physical

fatigue. If the vehicle can avoid obstacles and collisions and autonomously drive to certain load-drop off locations, while allowing the operator to choose specific locations to mine, then operator workload and performace would be greatly increased [82].

Another area of research is in human control of multi-vehicle (e.g. Unmanned Aerial Vehicles (UAVs)) coordination, where one or more human users may be issuing commands to multiple robots while the robots must maintain a formation or cooperate to complete a task. Currently, multiple operators are required to fly just one UAV [19] and some level of autonomy could lead to a reduction in the number of operators needed and overall operator workload. The challenge here is executing potentially conflicting user commands while maintaining coordination among the vehicles (e.g. vehicle formations). Previous approaches often involve the control of a lead UAV or by controlling the center of mass of the formation [24, 5], but these methods allow little human interaction with the shape of the formation.

Coordinated manipulation tasks are another research area where human and automated system commands could be utilized to accomplish grasp and movement tasks of objects [50]. The teleoperated nature of these tasks limits what sensory feedback the human user receives, while high-level task goals are difficult to program in automated manipulation. Hence, shared control schemes where grasp pressure for deformable or fragile objects can prevent damaging or dropping objects when a human operator is driving a remote serial manipulator arm. Specifically, these sorts of capabilities are integral for robotic assistants for astronauts [23] and robotic surgery [66].

In this thesis, we seek to address the problems highlighted by these application areas by developing a generalized control framework for combining human and robot inputs. Specifically, this thesis will present a control framework for composing human input (for high-level aspects of tasks) and automatic control (for low-level aspects of tasks) during cooperative tasks that will:

- Provide theoretical guarantees of low-level task completion
- Afford high-level task completion to human operators

- Result in low-level task performance-based sliding autonomy
- Provide task performance and operator workload benefits
- Apply to a variety of cooperative human-robot tasks including USAR and multi-robot tasks

The applications in this thesis highlight the strengths of the developed control framework in that they address the issues with cooperative human-robot control mentioned earlier in this chapter. For instance, the types of problems identified for vehicle navigation of heavy machinery are addressed in an assisted navigation task for mobile robots through human operator experiments. In particular, the control framework developed in this thesis is employed to allow operators to accomplish high-level tasks while an automated low-level task is simultaneously being completed, thus allowing the operator to concentrate on the high-level task. The operator experiments will show that such human-robot coordination can lead to lower operator workload.

In regards to USAR, an application treated in this work is the shared control of a search-and-rescue crawler. The crawler is designed for human-guided navigation through cluttered and unpredictable disaster sites to search for and rescue human victims. Here, a haptic/visual feedback teleoperated quadruped rescue robot is jointly controlled by a human operator as well as an automatic controller. The human operator is charged with using an on-board camera and haptic feedback to choose suitable foot placement in a cluttered and uncertain terrain, however, despite this task feedback, there is little operator situational awareness regarding static stability for the quadruped robot. Hence, the work in this thesis is applied so that the operator shares control of the robot with an automatic controller that ensures subsequent leg lifts and placement will not result in the robot tipping over or falling down.

We extend the presented control framework to apply to a multiple human, multiple robot coordination task for manipulating an object in the environment by way of a multi-robot formation. Here, multiple robots are required to drive in a formation (forming a gripper), while the operators can influence the formation (acting as a gripper) to manipulate an

object. This requires high-level cognitive tasks to position the gripper appropriately, while low-level task assistance is necessary in ensuring the multi-robot formation is maintained. In this way, we facilitate multiple operator interactions with large numbers of robots along with methods for coordinating possibly conflicting operator commands by guaranteeing formation maintenance.

Before we can employ the control framework for these applications, this thesis first discusses gaps in previous work in related fields followed by a presentation of the control framework that will address these gaps. Next, a treatment on the necessary control theoretic machinery required to make guarantees about the control framework is presented. This is followed by a discussion on the tools required to apply the framework to the discussed human-robot coordination problems. By first discussing these development details, we lay the foundation to finally deploy the developed control framework in order to provide cooperative human-robot control for application domains in USAR and swarm control.

Specifically, this thesis will be organized as follows: Chapter 2 will address background research and what challenges this thesis will address from the literature. Chapter 3 presents and develops the theoretical and philosophical approach to the human-robot cooperative control problem along with theoretical low-level task completion guarantees. Chapter 4 derives of a closed-form control law to solve a specific class of the problem presented in Chapter 3, and this control law is applied to a couple of examples of human-robot cooperative control problems. In Chapter 5, work on a human operator study is presented comparing the methods in this thesis versus manual control and demonstrating high-level task completion. The presented control framework is further applied in Chapter 6 to the shared foot-placement control of a quadruped rescue robot in cluttered search and rescue environments. Chapter 7 presents developments in the framework for multiple-operator-multiple-robot tasks. Finally, in Chapter 8, we conclude with a summary and future directions of work.

In the next chapter, we will discuss previous work, which has been heavily focused on user-interfaces, that allow humans to interact with a particular automaton by choosing its behavior or monitoring its progress and intervening when necessary. These approaches tend to utilize behavior-based control architectures [4] or logic based operations [58]; however,

these past approaches rely on experimental results and give no analytic guarantees on performance or task completion. These approaches also tend to be designed at fixed or discrete levels of autonomy, while often relying on the user to set these levels, despite the possibility that the user may not be capable of making such a decision.

CHAPTER II

BACKGROUND

The basic question underpinning this thesis is how to combine human operator control with autonomous control so that each entity can provide input to carry out high-level cognitive tasks (that humans excel at) as well as low-level tasks (that automatic controller excel at). Previous research in this area has roots in the Human Robot Interaction (HRI) field as well as in the teleoperation control literature. Researchers in the HRI field refer to the type of interaction proposed here as mixed initiative interaction, or sliding/shared autonomy (e.g., [34, 76]) in that the robot is roughly on the same level of authority/autonomy as the human in terms of completing the task. In the teleoperation domain, this type of interaction can be referred to as shared control or human-in-the-loop, as in [37, 85, 2], since both controllers (human and computer) act on the same dynamic system.

In this next section, we will discuss previous work in human-robot interaction and the related shared control field. This portion of the literature discussed will focus on the single operator paradigm. In the following section, theoretic tools in model predictive control and graph-based multi-agent control will also be discussed, as these tools are utilized to develop the control theoretic framework presented in this thesis. This background section will also cover applications of human-robot cooperation in the Urban Search and Rescue (USAR) domain as the work in this thesis has been applied to the shared control of a quadruped rescue robot. Finally, we discuss how previous work in the fields of HRI and shared autonomy/control has addressed systems with multiple operators and multiple robots working in concert.

2.1 HRI for Cooperative Control Tasks

In [34], a Levels of Autonomy (LOA) scale, as seen in Figure 1, is used to define the types of human-robot interaction, ranging from direct human control of the robot (teleoperation)

to peer-level (mixed-initiative) interaction between human and robot. Mixed-initiative interaction is accomplished through dynamic autonomy which incorporates interactions that range from direct control to full autonomy based on which type of interaction is most effective. We will refer to this scale throughout the following discussion. To address dynamic

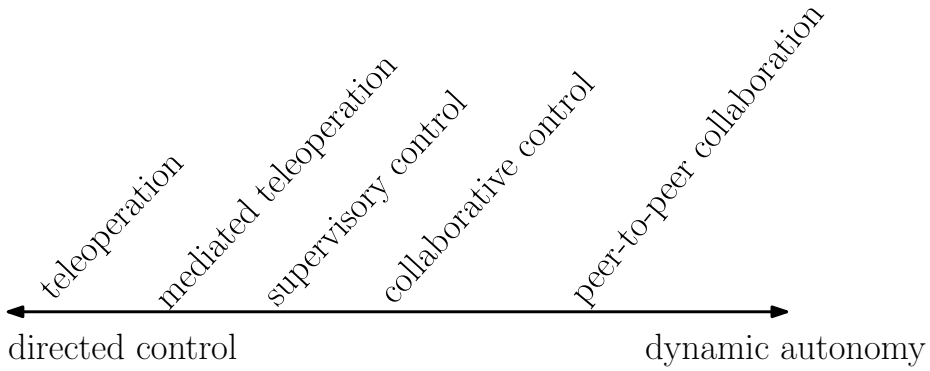


Figure 1: As in [34], the Levels of Autonomy(LOA) scale of Human-Robot Interaction is described.

autonomy, Desai and Yanco [22] extoll the need for sliding along this scale of autonomy, where the mode of interation changes between humans and robots smoothly as the cooperative task evolves, not just in disrete levels of autonomy as is commonly implemented. This need is highlighted in USAR tasks, where a robot's autonomous capabilities and the human operator's situational awareness (i.e. what aspects of the environment the operator can perceive in relation to the task) may vary based on the type of disaster site, thus requiring a shift in the level of autonomy required. The following section will include a discussion on not only previous techniques on how to combine human and robot control, but also on how previous research has or has not addressed sliding autonomy.

A large body of work within the HRI field has focused on user-interfaces, planning-level human intervention, and work-flow models for interaction (e.g. [64, 81, 6, 84, 31, 63, 8]). These types of interaction are often fixed towards the supervisory side of the Levels of Autonomy scale.

In [31], a design approach is detailed where a user interface that displays information gathered by a monitoring system that tracks the robot's state and planned actions in a mobile rescue robot scenario. The human and robot "trade-off" control of the system and the

benefits of the trade-off in control is highlighted in the scenario where poor visibility makes it difficult for the human to make decisions on the robot's operations and, subsequently, relies on the robot's autonomous capabilities to navigate the environment. Again, attention is mainly paid to how to switch between robot and human control instead of simultaneous control. This level of autonomy is on the supervisory side of the Levels of Autonomy scale. Similarly, the work by Mano et. al. [54] allows the human user to switch between teleoperation and an autonomous mode during rescue robot tasks, but does not take advantage of concurrent operation of the two modes.

The researchers in [8] present a hybrid deliberative/reactive robot architecture with four discrete LOA: a teleoperative mode, a safe mode, a shared mode, and an autonomous mode. The safe mode is similar to the teleoperative mode except that an obstacle avoidance behavior is implmented using a "guarded motion behavior" that slows the robot as it gets near obstacles despite user-issued joystick commands. In shared mode, the user has the ability to select behaviors through a graphical user interface but does not directly drive the robot. This mode lies on the supervisory end on the LOA scale, while the safe mode leans towards the teleoperative end of the scale. These modes are chosen a priori by the user and rely on the user to have the requisite situational awareness to set the appropriate level of autonomy.

Morris et. al. in [63] propose a shared control system implemented on a robotic walker that aids the user in navigating a cluttered and uneven environment through a haptic interface. Efforts were made to model user intent using force-sensor data on walker handles which was then fed to the control interface. Control is implemented in three discrete levels of autonomy: a teleoperative mode, a shared control mode, and an autonomous mode. The shared control mode slows the walker if the user-generated trajectory (to a predetermined goal) differs greatly from a robot determined path. The robot will completely stop if the user-generated path deviates too much from the robot-planned path. In this sense, the walker attempts to keep the user following a robot determined path. However, if the robot is capable of completing the task without user input and the task is halted if the user deviates from the plan, it is unclear what purpose the user input serves in that such a

system can be implemented in way that lets the human and robot trade-off command.

An approach to composing human and robot inputs is proposed by Arkin and Ali [4], who utilize a schema-based approach in which human input is incorporated by treating it as another behavior in a behavior-based robot architecture. The human input is summed with other weighted behaviors and normalized in the common schema-based behavior arbitrator. The robot's behavior here is "emergent", so no conclusions can be drawn about whether the task is ever completed or how the human input effects the system.

Alternatively, Wegner and Anderson [87] propose a "teleautonomous" approach that trades off, or "blends," autonomy for teleoperation commensurate with the level of difficulty of a search and rescue task. The control allows the user to set the level, through a user interface, to which the human commands are weighted versus the autonomous control. The control algorithm also has a software agent check the validity of the human commands, i.e. will the human commands result in safe operation. However, in order to set the human command weighting, this approach relies on the human operator to be constantly aware of the difficulty of the task and the capabilities of the robot while simultaneously recognizing the state of the system in regards to autonomous task-completion.

In [22], Desai and Yanco argue that mixed intiative requires the human and robot to interact on sliding levels of autonomy, whereas shared control operates at a fixed LOA, and previous mixed intiative approaches simply operate with a few discrete LOAs. In their work, the authors present a behvaior-based mobile robot control architecture to slide between four discrete LOA through the use of a behavior arbitrator. In other words, sliding autonomy is accomplished by varying weights on human and robot input in the behavior arbitrator. Speed commands are a weighted sum of human and behavior-based inputs, while the rotational component combines the human component with an obstacle avoidance behavior and virtual-wall force fields (enforced in four cardinal directions). Experimental results are given showing mobile robot navigation through an obstacle-laden environment, however, no performance or safety guarantees are made by the authors.

In a different approach, Fong et al. [32] present a multi-robot driving task using collaborative teleoperative control, which allows for different levels of autonomy by allowing the

operator to give low-level ("turn right", "turn left") as well as high-level commands ("Is this a rock?","Can I drive through?"). This work concentrates on relaying task dependent data or "dialogue" between human and robot during task execution. Such communication is mainly accomplished through a user interface that allows for both robot-to-human queries as well as human-to-human dialogue when multiple human users are involved with the task. The system allows for tasks with one or several mobile robots during robot navigation and surveillance tasks and is experimentally demonstrated with one and several mobile robots.

In [53], a mixed initiative control utilizing navigation-function-based controllers is combined with human input to drive a differential drive robot to a goal state. The navigation functions are cost functions with a global minimum representing a goal state such that the control drives the system to this goal state. When human input is incorporated into the controller, the human user can drive the robot away from the planned path and once the user stops issuing commands, the controller will drive the system towards the goal state again. In other words, the robot will follow the gradient of the navigation function to the goal state. This work incorporates the tenets of mixed initiative in that the robot can carry out the task while accepting human input to change course with equal authority. Since no guarantees are made about reaching the goal state with concurrent human input, the user may well be able to prevent the robot from ever reaching the goal state. Guarantees are only made when the user stops issuing commands, which more resembles the control trade-off scenarios discussed previously.

2.2 Shared Control

The shared control literature also addresses the problem of integrating human inputs with inputs for automated tasks. These works more often address concurrent control of systems (instead of trading off) and address application domains like haptic control of manipulation devices, mobility assisted walkers and wheelchairs, and shared control of heavy machinery (such as mining vehicles and robot manipulators as in [36, 42, 86, 61, 82, 18]).

Earlier work in shared control has focused on graphical user interfaces or sensory feedback (such as haptic or visual feedback) to relay task-dependent data to the human and

to relay human control information to an automatic controller or autonomous agent (e.g., [43], [10]). Here we instead focus on works that deal with the design of the actual control laws.

In [37], shared control for vehicle steering during a road following task was examined using a motorized steering wheel and human driver. An automatic controller applied torque to the steering wheel to maintain a vehicle heading that follows the road while the driver had to overcome this torque to make any corrections to the steering angle. Hence, the control scheme relied on physical human force to impart the intended behavior on the system. This system allowed the human to override the automatic control, but there is no guarantee that the vehicle remains on the road as the human can overcome the automatic control.

Parikh and Kumar [70] present research in which a deliberative planner, reactive object avoidance behavior, and human input are integrated to control a smart wheelchair. The planning behavior is implemented using a potential function for global planning, where the robot follows a gradient towards a specified goal. The reactive behavior calculates a gradient, tangent to an obstacle, that steers the wheel chair away from obstacles. If human inputs follow the gradients of both behaviors, then the human input is used as it was commanded. However, if the human input is not consistent with these gradients, then the human input is projected onto the combined obstacle avoidance and goal gradients. This resulted in the robot heading towards the goal and avoiding obstacles while attempting to move in the direction the human intended. In this way, the human can relinquish control to the robot or influence the robot control. If human commands vary greatly from the intended path towards the goal, then the deliberative goal commands are neglected and only the user and obstacle avoidance commands are used. In this sense, the human's role ranges from a teleoperative one to a supervisory role. Experimental results are presented showing task completion, but analytic guarantees of task completion are absent and the issue of the human continually giving commands that prevent the robot from ever reaching the goal is not addressed.

In another approach, the authors in [58] utilize a fuzzy-logic scheme to compose go-to-goal commands issued by the user with an obstacle avoidance input. Discrepencies in

human input and behavior-based control is categorized into five different fuzzy-logic states, and a fuzzy logic matrix is used to decide how to combine these inputs. As with previous works, the fuzzy logic matrices and membersip functions are experimentally determined. Therefore, no guarantees of task-completion are given, although, the fuzzy states determine how much the controller will interject, resulting in varying levels of human control.

In [86], the authors utilize motion capture data in an attempt to discern user intent from forces/moments applied by the user on assisted-walker handles. The authors discuss the idea of "user intent" as reflecting "the control system's estimation of the human user's goal." This intent is then compared to the intelligent walkers actual movement and the authors discuss how shared control can have an impact for assisted walkers. Namely, the authors examine whether the control system can provide user safety while simutaneously executing the user's intent, and experimental results are given for a particular assisted-walker.

The authors in [3] propose a model-predictive method for composing human operator commands for steering a vehicle with automated safety control. Experimental results are presented showing that human driver inputs are modified to keep the vehicle on a particular road surface, however, no analytic guarantees are made about the safety constraints on the system. In fact, the author discusses the need for safety guarantees for such controllers.

2.3 Control Theory Tools

The following is a discussion of previous work regarding the theoretical and analytical tools utilized in this thesis. Namely, model predictive control is employed in the human-in-the-loop control formulation, and graph-based multi-agent control is important for the multi-agent formation control.

2.3.1 Model Predictive Control

Model Predictive Control (MPC), or Receding Horizon Control (RHC) as it is sometimes referred to as, has roots in the process control industry, where plant dynamics are slow and often difficult to model. In addition, constraints on state and control input (such as saturation limits) play a significant role. In MPC formulations, an optimal control problem is solved at *each* time instant subject to state and input constraints.

A major advantage of MPC is that the control is updated by the measured state of the system at each time instant for dynamic models that are only valid for shorter periods of time. This is in contrast to classical optimal control methods, where the feedback law relies on an accurate long-term model of the dynamics. This is an important feature of MPC that will be exploited in this thesis. Recalculating the control at every instance, of course, requires computation time; however, slow model dynamics make these computations feasible.

Although application driven at first, the model predictive control literature has a strong analytic foundation [30, 35, 57, 75, 56] with results showing closed-loop stability for discrete-time, linear and nonlinear systems. Conditions and proofs required to guarantee stability (convergence to the origin or equilibrium point) are often constructed using Lyapunov analysis and the following works will form a basis on which the proposed work will build upon.

In [57], the authors present a survey of the model predictive literature and present an over-arching view of the control formulation, giving conditions for stability. In particular, the authors address constrained linear systems in discrete time with constraints on state, input, and terminal state. Techniques utilizing special terminal cost and terminal constraint sets to ensure stability are given and proven using Lyapunov theory, since the existence of constraints naturally make the controller nonlinear regardless of whether the system is linear or not. In Mayne and Michalska's work [60], the authors detail a robust continuous time receding horizon (or MPC) controller for nonlinear dynamic systems, where the optimal control *horizon* as well as the optimal control is calculated such that the cost is minimized over the control horizon.

The authors in [75], propose a dual-mode MPC scheme where the first mode is an MPC controller that drives the system to a constraint set. Within this constraint set, a local controller (the second mode), that is invariant on the constraint set, can be used to stabilize the system. Again, Lyapunov theory is used to show stability and clearly the constraint set must be chosen carefully to ensure there exists a controller that can asympotically stabilize the system. It should be noted that this second mode is intended as a way of estimating the cost of reaching stability within the constraint set and is not necessarily applied to the

actual system when within the constraint set. The human-in-the-loop control formulation presented in this thesis will utilize this particular type of MPC.

Thomas et al. [80] propose a Shrinking Horizon MPC (SHMPC) formulation for the control of Autoclave curing of composite materials. In this application the control horizon is shrunk as the process goes on because the process has a termination time at which the process must cease. The authors detail how SHMPC is applied to the process control but give no analytic results for convergence or feasibility. In this thesis, we will present a different method for varying the control horizon in MPC.

With ever faster computing, MPC can now be applied to dynamical systems with faster plant dynamics. In particular, researchers have applied MPC to robotics applications such as mobile robot planning and robot formation control such as in [83, 88, 26, 3, 48].

Murray and Dunbar [26] utilize an MPC formulation for the control of multi-vehicle formations with nonlinear dynamics. The cost over a finite time horizon and terminal cost are used to drive the system towards a set of states that match a reference formation. Simulation of three vehicles show that the controller asympotically stabilizes the vehicles to the desired formation without specifying a leader or specifying specific vehicle locations. Instead, the formation is defined by a set of possible vehicle states. Simulation results are provided that show a reference formation is maintained. Similarly, Wesselowski and Fierro [88] utilize a dual-mode MPC to achieve robot formations, where a terminal constraint set is used to stabilize the formation instead of the terminal cost employed by Murray and Dunbar. Simulation results are given showing asympototic stability to a leader-follower defined formation. Analytic stabilization results are inherent in the use of well-established MPC formulations.

2.3.2 Graph-based Multi-agent Control

Another set of theoretical tools utilized in this thesis is graph-based multi-agent control which has a rich literature base. Specifically, we will discuss multi-agent formation control with inter-agent interactions modeled by graphs [59, 68].

The work by Olfati-Saber and Murray in [74] discusses graph-based linear consensus

protocols for distributed networks of agents where agents are modeled as nodes and inter-agent communication is modeled as edges in a graph. Algebraic graph theory is used to show convergence to an agreement state.

In [59], the authors present tools for modeling and characterizing multi-agent systems as graphs. Consensus algorithm-based control is developed for driving multi-agent networks to formations, and concepts such as controllability of multi-agent formations and graph rigidity are presented to characterize how these types of systems can be controlled.

The authors in [47, 69] model multi-agent formations as graphs and provide definitions and conditions on types of distance-based graph formations. Graph-based methods for controlling these formations utilize properties of the rigidity matrix, which serves as a way of defining distance-based formations. In particular, agents update their positions, using the null space of the rigidity matrix, such that inter-agent distances are preserved. More detail on rigidity properties are discussed in [77]. In this thesis, we will address multi-agent control of formations that are not necessarily rigid, but we will utilize the rigidity matrix to formulate formation control methods.

In [79], the authors propose a decentralized control scheme whereby a group of non-holonomic vehicles display flocking behaviors by maintaining heading and spacing with the rest of the group while avoiding collisions. The control law involves a potential force field and alignment field to accomphish the flocking. System theoretic results, namely utilizing Lyaponuv analysis and algebraic graph theory, are given that show convergence of the group to aligned headings/velocities while maintaining inter-vehicle spacing. The resulting multi-agent behavior is often referred to as swarm behavior.

2.4 Target Applications

The remainder of this section discusses applications where the proposed work would have an impact in terms of addressing how human input incorporated into automated tasks would lead to greater effectiveness and safety for engineered systems. The first application is integrating human input for semi-autonomous robots operating in USAR environments, while the second application addresses human command of multi-robot formations.

2.4.1 Human-in-the-Loop Control of Rescue Robots

In [64], the author reviews the state-of-the-art HRI progress in the rescue robot domain. The benefits and challenges of incorporating robots into a search and rescue environment are discussed by detailing the tasks and types of interaction needed between robots and humans to complete these tasks. Specifically, the lack of situational awareness (limits on what aspects of the environment the robot or operator can perceive) adds to the difficulty in performing search and rescue tasks. Adding autonomy to the robot can lead to benefits such as decreasing the ratio of human operators to robots as well as improved information flow among human responders and rescue robots.

The authors in [65] highlight the advantages of having varying levels of autonomy for robots in USAR scenarios. Safe navigation of complicated and cluttered environments may require more teleoperation of the robot, but scenarios where humans have limited situational awareness may require more autonomy on the robot's part. This autonomy may include identifying and avoiding obstacles not in view of the operator's camera.

The authors in [29] present a human-machine interface that enhances teleoperative control for a quadruped robot by allowing the user multi-modal control of the robot in that the human can specify the mode of control, i.e. specific joint angles for foot placement or higher level instructions that manipulate body velocity for task completion. The LOA of this implementation is weighed heavily towards teleoperation, and simultaneous control of the robot is not discussed.

Bruemmer et. al. [7] conduct extensive (100 participants) novice user studies to experimentally test the effectiveness in carrying out mixed-initiative control (with their user interface) in a USAR scenario. They investigate four discrete modes of operation: teleoperative mode, safe mode (obstacle avoidance), shared mode (directional input), and autonomous mode (where the user chooses high-level tasks for the robot to complete). Results show that operators using the shared mode performed signifantly better than when using the safe mode, suggesting that adding robot autonomy to human commands for robotic tasks in USAR is beneficial. However, again, there are no guarantees about task completion in terms of the automatic control tasks and there is a lack of sliding autonomy.

Issues associated with teleoperation and control of a quadruped robot in cluttered and rough terrain environments are often addressed by attempting to provide the user with greater situational awareness of the work environment through the use of haptic feedback (e.g. [62],[40], [41]). Specifically, previous works have presented haptic feedback assisted teleoperation of a mobile robot navigated through a world with obstacles (e.g. [49], [40]). The results in these papers show that the addition of haptic feedback results in greater performance of the human operator during navigation tasks and an increased feeling of user immersion in the task environment. Hence, the haptic feedback provides a richer human-robot interaction as the human operator has a better sense of the environment the robot is working in. However, no guarantees for robot stability are given, i.e. visual and haptic feedback fail at informing the user of robot static stability. In this thesis, we utilize the presented control framework to address this issue.

2.4.2 Human Control of Multi-Robot Formations

The bulk of previous research in this area focuses on user interfaces and human control of a leader in a leader-follower network scheme for robot formations. Balch and Arkin [5] present a behavior-based control to drive robot formations by the unit-center (i.e. the center of mass of the robot team) as well as leader referenced approaches. The control combines "go-to-goal", "avoid-obstacles", and "maintain-formation" behaviors to form and maintain line, column, diamond, and wedge formations. This approach allows for a human operator to control the leader of these formations, since the authors argue that it is not feasible to ask the human operator to calculate the unit-center during operation. This approach assumes the operators can effectively issue commands that do not conflict with the specified formation (i.e. leader motions can be infeasible and lead to instances where leader nodes cannot accomplish the desired formation as is the case for non-rigid graphs). In other words, there are no guarantees that the leaders can maintain the specified formations.

Desai et. al. [21] describe a computationally feasible feedback control for driving distributed leader-follower networks into different formations, where the leader is controlled by an external controller such as a human operator or optimal controller. Simulation results

show that the formation is maintained while the leader robot is driven around obstacles in the environment. In addition, the network can be driven from an initial formation to a different goal formation. Here, stability is shown for a rigid class of formations without any restrictions on the operator input. This allows the human operator to translate and rotate the formation, but since this method only addresses rigid formations, there is no freedom afforded the human to alter the formation to complete tasks.

As discussed earlier, the authors in [26, 88] utilize an MPC formulation for multi-vehicle formation control with nonlinear dynamics. The controller asympotically stabilizes the vehicles to a desired formation by specifying a set of possible vehicle states. However, human input into such a control scheme is not addressed and not trivial.

In [24], the authors propose a framework to allow a single pilot to control a group of UAVs following a leader-follower configuration. The framework allows the pilot to switch from a pilot-controlled mode, where the pilot can directly drive the lead UAV, to an autonomous mode, where the UAVs carry out a specified autonomous task. This falls under the supervisory control on the LOA scale, where an optimal timing control is employed to aid the pilot in deciding when the pilot should switch to pilot-mode or autonomous mode.

In [19], the author discusses the challenges of implementing human control of large networks of agents (swarms) and states that the complexity of interactions between many agents naturally leads to the need for inter-agent autonomy. However, the author stresses that the role of the human can range in terms of levels of autonomy and that further investigation is need to determine the most the effective roles.

While these methods effectively allow a human operator to translate and rotate rigid formations, this previous work does not address formation control for flexible (or non-rigid) graphs. With flexible graphs, human operators would have more freedom to alter the formation while still maintaining some formation structure. However, full human operator control of such non-rigid formations can lead to scenarios where maintaining inter-agent distances is infeasible. In order to assist human operators in maintaining formations, we would again require a controller that combines human operator input with an automatic controller that is able to maintain the desired non-rigid graph formation. These issues are

addressed in Chapter 7 of this thesis.

The additional freedom associated with such graphs also allows for the possibility (and in some cases the necessity) of additional human operators to manage the additional complexities of flexible graphs. The following discusses the challenges associated with previous work dealing with multiple operators and multiple robots.

2.4.2.1 Multiple Operator Control of Multiple Robots

More recent work in human-robot interaction and shared autonomy have begun to include multi-operator scenarios where one operator cannot handle all the cognitive tasks required to control a large group of robots (or in some cases referred to as swarms). In [89], the authors study the case when adding robots and situational awareness are not enough to enhance performance of these Multi-Operator, Multi-Robot (MOMR) systems. In some cases, the authors show that operators can be added to improve task performance.

The bulk of the literature on MOMR interaction focuses on user interfaces to increase operator situational awareness, as in [91, 44, 11, 50, 51]. Many approaches in dealing with operators and multiple robots often rely on turn-taking [64] where the robots tend to act autonomously and human operators intervene when deemed necessary by either the robot or operator (see for example [84, 44, 50, 39, 71]). However, these approaches do not allow concurrent control where the advantages of autonomous behaviors could be combined with human cognitive abilities while conducting the task. This is important as there may be instances when the autonomous control or human operators cannot complete the task on its own.

Research in MOMR control, when the two entities are concurrently issuing control commands (referred to as shared control or shared autonomy as in [89]), is limited. In [76], the authors use genetic algorithms to set fuzzy-logic levels to combine multiple operator and multiple robot control input. As discussed earlier when addressing single-operator, single-robot approaches, there are no guarantees that such fuzzy-logic will result in the automatic portion of the control being effective when added to the human control inputs.

The authors in [45] present a multi-tiered method of controlling large groups of robots.

The first tier involves allowing operators to set and adjust swarm-level interaction parameters like attraction and repulsion gains. The second tier involves allowing user to control virtual agents through these same attraction and repulsion gains in order to influence the behavior of the swarm. As with the behavior-based approaches in the single human/robot paradigm, the behavior is emergent and no analytic results can verify whether or not this system will accomplish the desired behavior.

2.5 Thesis Contributions

Previous research in the area of human-robot control lacks a formalization for combining human input and automatic control, which guarantees that the automatic control task will be completed in the face of possibly competing human operator input. This especially applies to tasks where it is difficult for human operators or robots to complete the task without the assistance of the other. As such, a smoothly sliding autonomy controller is needed that balances human input with automatic control based on low-level task performance of the human operator. This controller also needs to guarantee low-level task completion when the human operator is not capable of completing the low-level task, while also allowing the human operator freedom to complete high-level cognitive tasks. In order to address these inequities, this thesis presents the following human-in-the-loop control framework.

Specifically, in this thesis, a novel control theoretic formulation of the human-in-the-loop problem is presented by framing it as a receding (fixed or variable) finite horizon optimal control problem with a terminal state constraint as well as presenting a projection-based solution to a specific class of system dynamics and low-level task constraints. Additionally, a system identification approach is developed to make human input predictions in addition to a linear extrapolation method and the performance of the system identification is used to determine a variable horizon for the receding horizon control. The resulting control law is then applied to human-robot cooperation problems to highlight how one may apply these algorithms. The control framework is further studied on an experimental platform utilizing the developed controller in a mobile robot navigation task to investigate the human operator effects.

The work presented here will not only provide a method for composing human commands with automatic commands, but it will also give convergence guarantees that the low-level task constraint will be satisfied. The controller will execute the human commands exactly as they are issued if the controller predicts that these commands will accomplish the lower-level task and hence will also complete the higher level task. However, the control will alter the commands issued to the system if the human commands will not complete the lower level task, but will try to mimic the human commands as closely as possible to preserve higher level human intent. This approach allows for varying levels of autonomy based on how effective the human operator is at completing the low-level task. The proposed control law utilizes feedback to udpate the control effort at every instant and, for this reason, inherits robustness properties similar to infinite horizon controllers.

Although we largely ignore the user interface issues that arise in the mixed initiative domain, our controller does not exclude the state-of-the-art in providing situational awareness and decreased operator workload/increased preformance interface design. The work in this paper seeks to advance the capabilities of those systems by providing capable methods of composing human and autonomous behavior such that both entities concurrently have control over the system.

To demonstrate the effectiveness of this controller in real human-robot cooperation tasks, we deploy the control framework on an experimental platform for the shared control of a legged rescue crawler. Here, the human is vital to successfully navigated cluttered search and rescue terrain, while concurrently, the human needs assistance in maintaining balance of the robot as this situational awareness is lacking in audio/visual feedback and additional cognitive workload is not desired. Utilizing the control framework presented in this these, we can make guarantees about robot stability while giving the operator freedom to choose foot placements.

The advantages that this framework provide allow us to incorporate multiple operators and multiple robots with the same convergence guarantees. Specifically, through a simulation environment, human operators are given control over leader agents that must form

a non-rigid formation, for which swarm agents will gather around to form a swarm gripper. The non-rigid formation allows the human operators freedom to "actuate" the gripper, which thereby requires that the operators are assisted in managing this freedom. In other words, the presented control framework will be used to help the operators avoid leader agent configurations that make the non-rigid graph infeasible to maintain. Additionally, the operators are given the high-level cognitive task of commanding the swarm gripper to interact with an object in the environment and move it to a goal area. Operator study results are also given for this scenario showing both low-level task performance benefits as well as lower operator workload.

CHAPTER III

A MPC-BASED APPROACH TO COOPERATIVE CONTROL

In this chapter, we formulate a model-predictive controller to implement the composition of human input and automatic task completion. The formulation specifically treats the low-level task as a set of goal states that satisfy a constraint, meaning that if the system is driven to this set of goal states, then the low-level task has been satisfied. However, the enabling idea behind this formulation is the accommodation of high-level task freedom within this goal set. We make no claims about what exactly the high level task is (we do not define it in the formulation), however, the operator can influence which state in the set of goal states to drive to, and the operator can influence how the system reaches this goal set. We refer to this influence as "human intent" for the system and we postulate that this intent can be inferred by the operator's inputs to the system. Hence, the model-predictive control formulation follows.

3.1 Problem Formulation

Suppose we have a discrete-time dynamic system,

$$x_{k+1} = f(x_k, v_k) \tag{1}$$

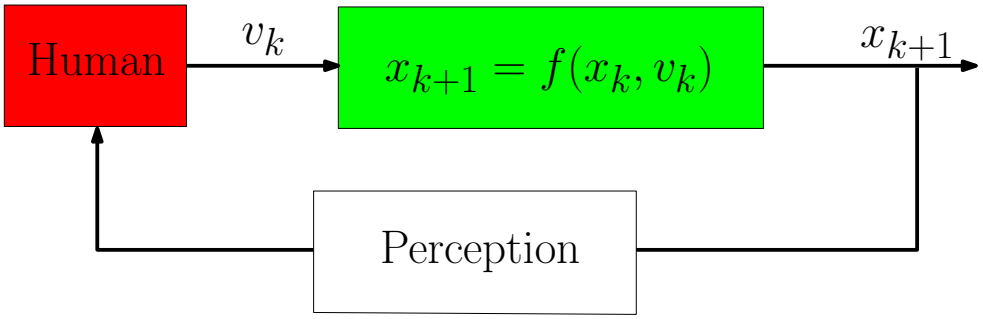
with $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$, $x_k \in \mathbb{R}^n$, where a human operator is issuing the commands $v_k \in \mathbb{R}^m$ as shown in Figure 2(a). If part of the task is to satisfy certain state constraints, commanding the system to do so may not be a trivial task. On the other hand, it is certainly possible to design an automatic controller that can handle the task of satisfying state constraints, as in [17]. The problem we wish to address in this thesis is to devise a controller that drives the system in such a way that both the state constraints are satisfied (low-level task) and the human operator's "intentions" for the system behavior (high-level task) are respected as much as possible.

In order to preserve the human operator's intentions for the system behavior, we wish

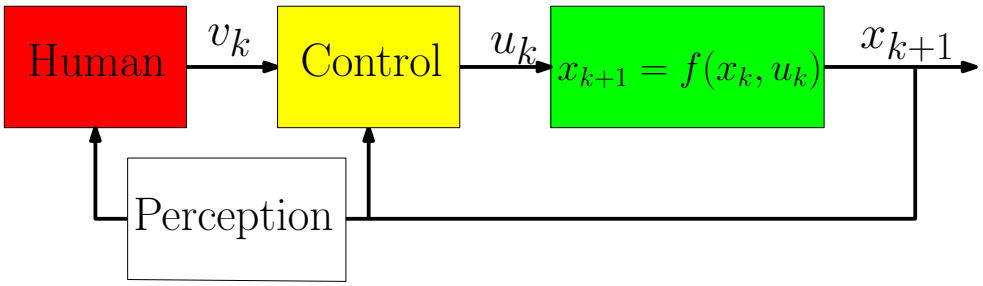
to design a control law that minimizes deviations from the human input while also ensuring that the state constraints will be satisfied. In order to do so, we will replace v_k in (1) by a control input u_k, which will drive the system to satisfy the state constraints, as

$$x_{k+1} = f(x_k, u_k), \tag{2}$$

where u_k is somehow close to v_k (the human input) as shown in Figure 2.



(a) Block Diagram for Human Control of a Dynamic System



(b) Block Diagram for Human-in-the-Loop Control of a Dynamic System

Figure 2: Control philosophy for human-in-the-loop control of a dynamic system: The proposed controller will stay close to the human input signal while completing a lower-level task.

To accomplish this, we must predict where the human operator intends to drive the system which, in turn, requires a prediction of future human operator inputs. In Section 4.1, we will present three methods for making this prediction, zero-order hold, first-order hold, and system identification. Thus, at every time instant, k, we have a predicted sequence of human input values, denoted by $\mathcal{V}_k = \{v_k, \ldots, v_{k+N_k-1}\}$, where $N_k \in \mathbb{N}$ is a fixed or variable time horizon. Note that for fixed horizons, N_k is constant for all k.

We want to find a control sequence, $\mathcal{U}_k = \{u_k, \ldots, u_{k+N_k-1}\}$, that minimizes its deviations from $\mathcal{V}_k$, while ensuring that the state resulting at the end of the sequence satisfies a state constraint. The states that satisfy the state constraint are given by the set $\mathbb{X}_f \subset \mathbb{R}^n$.

Now, this state constraint can be thought of as a terminal state constraint for the following receding horizon optimal control problem,

$$\mathcal{P}_{N_k} : \min_{\mathcal{U}_k} V_{N_k}(\mathcal{V}_k, \mathcal{U}_k) \tag{3}$$

where

$$V_{N_k}(\mathcal{V}_k, \mathcal{U}_k) = \sum_{i=k}^{k+N_k-1} L(v_i, u_i), \tag{4}$$

$$L(v_i, u_i) = (u_i - v_i)^T (u_i - v_i), \tag{5}$$

such that

$$x_{k+1} = f(x_k, u_k), \tag{6}$$

subject to

$$x_{k+N_k} \in \mathbb{X}_f \tag{7}$$

with $x_k \in \mathbb{R}^n$ and $u_k \in \mathbb{R}^m$. The actual control, u_k, that will be applied to (2) at time k is the first element in the optimal sequence $\mathcal{U}_k$. It is important to note that, as part of model-predicitive control, this optimal control problem is solved at every time instant, k. This allows the control to be updated with new state and human input information. Additionally, the MPC framework only requires that this human input prediction be a valid model of future human input over the control horizon N_k. Therefore, the framework relies on human input predictions that are updated at every time step and only need be accurate for a short time into the future.

In this formulation, the cost (5) penalizes deviations from the human command in order to preserve human intent. The terminal state constraint (7) guarantees that the state constraint, which is required for the lower level task, is enforced at the end of the time horizon. Without the terminal constraint, (7), the control would simply equal the predicted human input. However, the terminal constraint may cause the control to deviate from the human input in order to ensure that the terminal constraint is satisfied.

The choice of finite horizon, N_k, is crucial in that a large N_k requires that the future prediction of the human input be accurate over this long time horizon, otherwise the control

will not reflect the intent of the user. If N_k is too small, the control effort attempts to reach the constraint set within a small amount of time, so deviations from the human input can be large. As such, N_k must be chosen short enough such that the prediction of the future user input is valid and long enough that user intention for the state is maintained. This is the impetus behind the variable horizon formulation of the model predictive control which is further discussed in Section 5.

The problem $\mathcal{P}_{N_k}$ is a standard optimal control problem and can be solved analytically or numerically. An analytic control law is presented in the next chapter for a specific class of terminal state constraint, namely a linear state constraint of the form $\mathbb{X}_f = \{x \mid Mx = b\}$ for $M \in \mathbb{R}^{l \times n}$, $b \in \mathbb{R}^l$, and linear dynamics.

Dynamic autonomy is achieved by this controller in that any raw human operator command (i.e. the control sequence $\mathcal{V}_k$), that drives the system to the goal set at the end of the control horizon will be used as given (i.e. fully manual control). If the contrary, then the system will seek to correct the command while trying to also accomplish operator intent. In this way, the human operator can be a supervisor or have a more active role depending on how close the human commands come to carrying out the lower-level task. Additionally, if the human provides no input, the robot will satisfy the low-level task without any human intervention (full autonomy). The result is a controller with varying levels of autonomy without any explicit need to specify the level. The ideas of minimizing deviations from human input and driving to a goal set lead to the freedom of expressing human operator intent (high-level task completion) while the following convergence discussion will guarantee low-level task completion. We will address high-level task completion with data resulting from human operator studies in Chapter 5.

3.2 Proof of Convergence

Previous works in model predictive control have established conditions and methods to prove asymptotic stability for these controllers (e.g., [35], [75], [30], [57]). This formulation differs from these earlier works in that the proposed control law does not necessarily result in asymptotic stability but in the convegence of the state to a terminal constraint set. In

addition, the constraint may not necessarily contain an equilibrium point of the dynamic system and the convergence results, moreover, have to hold for both fixed and variable horizons.

Here, we adapt the stability proofs detailed in [35] and [75] to show that the optimal control will indeed drive the state to the terminal constraint set given by (7). This will give us low-level task completion guarantees regardless of whether or not the user is issuing commands that will satisfy the low-level task. The proof is based on the dual-mode MPC scheme (see [57]) where the system is driven to satisfy a constraint set by MPC, then another (locally invariant) controller is employed, within the constraint set, to ensure that the system drives to the stability point. This second mode is mainly used as a way of estimating the cost once in the constraint set and serves as an intermediate step for the proof. However, when using such a control framework on engineered systems, the locally invariant controller does not need to be used while the state is in the constraint set.

For these techniques to be applicable, the stage cost has to be bounded below by a K-function, which gives us a positive and increasing cost with respect to the norm of the difference between human input and control input. We will need to introduce an additional condition, namely that the stage cost should be zero when in the terminal constraint set (goal set), which requires that the human input will be used as the "locally invariant" controller in the proof. Thus, we require that the human operator must not "knock" the system away from the goal set once it has been reached or, in other words, the operator is capable to keeping the system in the constraint set (i.e. locally invariant). In addition, an input constraint will be placed on the human control when the state of the system is in the terminal constraint set, so that the human cannot apply large inputs to "knock" the system out of the constraint set. Specifically, the input constraint set, $\mathbb{V}(x_i)$, is a subset of $\mathbb{R}^m$ and is a function of the state, x_i, in that the human input is only restricted to $\mathbb{V}(x_i)$ when $x_i \in \mathbb{X}_f$. Hence, the following conditions will be needed to show convergence to the terminal set:

C1 $L(v_k, u_k) \geq \gamma(\|(u_k - v_k)\|)$, where γ is a K-function and $L(0,0) = 0$.

C2 $L(v_k, u_k)$=0 for all $x_k \in \mathbb{X}_f$.

C3 The set $\mathbb{X}_f$ is positively invariant under control v_i such that $f(x_i, v_i) \in \mathbb{X}_f$, $\forall x_i \in \mathbb{X}_f$ and $\forall v_i \in \mathbb{V}(x_i)$, where $\mathbb{V}(x_i)$ is an input constraint on the human input.

C4 A solution to $\mathcal{P}_{N_k}$ exists for a set of initial states denoted by $\mathbb{F}$.

Condition C1 is clearly satisfied by our choice of stage cost (5), while C2, C3, and C4 are assumptions that we must make about our system. Conditions C2 and C3 ensure that once the system reaches the terminal set, the stage cost is zeroed and the system will not be driven out of the constraint set, i.e. $u_i = v_i$ (the applied control is the human input) and $x_{i+1} = f(x_i, u_i) \ \in \mathbb{X}_f \ \forall \ x_i \in \mathbb{X}_f$. These two conditions imply a "strong" assumption in that we assume that the bounds on the human operator control and the ability of the operator is sufficient for keeping the state within the constraint set once this set has been reached. In other words, the human operator is trusted with the control to make $\mathbb{X}_f$ invariant. This is a reasonable assumption because once the system has converged to the state constraint set, it should be obvious to the human operator that large incorrect command inputs will force the system out of the constraint set.

It is important to note that conditions C2 and C3's primary purpose is to facilitate the asymptotic convergence proof by providing a way to estimate our cost function when the state is in the constraint set, which is zero. In terms of implementation, it is unnecessary for the human to be actually given full control over the system and the MPC control law can still be used when in the constraint set (this is true for the original dual-mode framework as well). Conditions C2 and C3 are part of an intermediate step required for the proof, but, in practice, the MPC scheme can be used even when the system state is in the goal set. In this way, the human is still assisted in the task while the state is in the constraint set.

For the rest of this section, we use the notation $V^{opt}_{N_k}(\mathcal{V},\mathcal{U}^{opt}_k)$ to denote the optimal cost given the horizon N_k and the optimal control sequence $\mathcal{U}^{opt}_k$ at time k. Similarly, $V^{fea}_{N_k}(\mathcal{V},\mathcal{U}^{fea}_k)$ will denote the feasible cost given the horizon N_k and the feasible control sequence $\mathcal{U}^{fea}_k$ at time k. Also, $\{0\}$ denotes a sequence of zero vectors in $\mathbb{R}^m$ with $N_k - 1$ elements.

To show asymptotic convergence, the cost $V_{N_k}(\mathcal{V}_k,\mathcal{U}_k)$ will be used as a Lyapunov function and we will show that the following properties hold, which in turn will be sufficient to ensure convergence:

P1 $V_{N_k}(\mathcal{V}_k,\mathcal{U}_k) \geq \gamma(||u_k - v_k||)$ for some K-function $\gamma(.)$.

P2 $V_{N_k}(\{0\},\{0\}) = 0$.

P3 $V^{opt}_{N_k}(\mathcal{V}_{k+1},\mathcal{U}^{opt}_{k+1}) - V^{opt}_{N_k}(\mathcal{V}_k,\mathcal{U}^{opt}_k) \leq -\gamma(||u_k - v_k||)$ for all $x_k \notin \mathbb{X}_f$.

The cost, $V^{opt}_{N_k}(\mathcal{V}_{k+1},\mathcal{U}^{opt}_{k+1})$, is the result of applying the optimal control, u^{opt}_k, to the system at time k to get the state at time $k+1$, x_{k+1}.

3.2.1 The Fixed Horizon Case

First, we will show asymptotic convergence to the goal set for fixed horizons, and we will thus let $N_k = N$ for a constant $N \in \mathbb{N}$.

Theorem 3.2.1 *Given Conditions C1-C4, a fixed horizon N, and an initial state for which a feasible solution exists to $\mathcal{P}_N$, the state will converge to the constraint set, $\mathbb{X}_f$, as $k \to \infty$, when the first element in the optimal solution to $\mathcal{P}_N$ is applied at each iteration.*

Proof 1 *The solution to $\mathcal{P}_N$ exists for $x_k \in \mathbb{F}$, where $\mathbb{F}$ is the set of initial states for which a feasible solution can be computed. From [75], a solution is feasible if the solution results in the satisfaction of the state and input constraints on the optimization problem. The optimal control law that solves $\mathcal{P}_N$, given $\mathcal{V}_k = \{v_k, v_{k+1}, \ldots, v_{k+N-1}\}$, results in the following control and state sequences,*

$$\mathcal{U}^{opt}_k = \{u^{opt}_k, u^{opt}_{k+1}, \ldots, u^{opt}_{k+N-1}\} \quad (8)$$

$$\mathcal{X}^{opt}_k = \{x^{opt}_k, x^{opt}_{k+1}, \ldots, x^{opt}_{k+N-1}, x^{opt}_{k+N}\}, \quad (9)$$

with $x^{opt}_{k+N} \in \mathbb{X}_f$.

We will show that for all $x_k \in \mathbb{F}$, the successive state resulting from the first control in the optimal control sequence at time k, x_{k+1}, also has a feasible solution (i.e. $x_{k+1} \in \mathbb{F}$). Written another way, at time k, u^{opt}_k is applied to the system with state x_k, to produce

x_{k+1}, *for which a feasible solution can be found given a human input sequence. Let* $x_{k+1}^{opt} = f(x_k, u_k^{opt})$. *Then, a feasible control and state sequence at time* $k+1$, *given the human input sequence* $\mathcal{V}_{k+1} = \{v_{k+1}, v_{k+2}, \ldots, v_{k+N-1}, v_{k+N}(x_{k+N}^{opt})\}$, *is*

$$\begin{aligned} \mathcal{U}_{k+1}^{fea} &= \{u_{k+1}^{opt}, u_{k+2}^{opt}, \ldots, u_{k+N-1}^{opt}, v_{k+N}(x_{k+N}^{opt})\} \quad (10) \\ \mathcal{X}_{k+1}^{fea} &= \{x_{k+1}^{opt}, x_{k+2}^{opt}, \ldots \\ &\quad \ldots \ , x_{k+N-1}^{opt}, x_{k+N}^{opt}, f(x_{k+N}^{opt}, v_{k+N}(x_{k+N}^{opt}))\}, \quad (11) \end{aligned}$$

assuming v_1, *from* $\mathcal{V}_k$, *is the human input at* $k+1$. *Recall that, by Condition C3, the human input control,* $v_{k+N}(x_{k+N}^{opt})$, *when* x_{k+N} *is in the state constraint set, provides* $\mathbb{X}_f$ *with invariance for all* $x_{k+N} \in \mathbb{X}_f$, $v_{k+N}(x_{k+N}^{opt}) \in \mathbb{V}(x_{k+N})$. *Hence, all states starting in the set of feasible initial states, will always stay in the set of feasible initial states.*

Properties P1 and P2 are satisfied by the choice of cost function (4) and Condition C1. It remains to show that Property P3 is satisfied. Using the sequences (8)-(11), we can show that the value function decreases by at least the initial stage cost,

$$V_N^{opt}(\mathcal{V}_{k+1}, \mathcal{U}_{k+1}^{opt}) - V_N^{opt}(\mathcal{V}_k, \mathcal{U}_k^{opt}) \leq -L(v_k, u_k),$$

since the optimal cost at $k+1$ *is bounded above by the feasible cost as given by optimality (i.e* $V_N^{opt}(\mathcal{V}_{k+1}, \mathcal{U}_{k+1}^{opt}) \leq V_N^{fea}(\mathcal{V}_{k+1}, \mathcal{U}_{k+1}^{fea})$*). So, the change in cost may be rewritten as*

$$\begin{aligned} &V_N^{opt}(\mathcal{V}_{k+1}, \mathcal{U}_{k+1}^{opt}) - V_N^{opt}(\mathcal{V}_k, \mathcal{U}_k^{opt}) \leq \\ &\quad V_N^{fea}(\mathcal{V}_{k+1}, \mathcal{U}_{k+1}^{fea}) - V_N^{opt}(\mathcal{V}_k, \mathcal{U}_k^{opt}). \end{aligned}$$

We can now continue by plugging in the sum of the stage costs,

$$\begin{aligned} & V_N^{opt}(\mathcal{V}_{k+1}, \mathcal{U}_{k+1}^{opt}) - V_N^{opt}(\mathcal{V}_k, \mathcal{U}_k^{opt}) \\ \leq \ & \sum_{i=k+1}^{k+N} L(v_i, u_i^{fea}) - \sum_{i=k}^{k+N-1} L(v_i, u_i^{opt}) \\ = \ & L(v_{k+1}, u_{k+1}^{opt}) + \cdots + L(v_{k+N-1}, u_{k+N-1}^{opt}) + L(v_{k+N}, v_{k+N}) - \\ \ldots \ - \ & L(v_k, u_k^{opt}) - L(v_{k+1}, u_{k+1}^{opt}) - \cdots - L(v_{k+N-1}, u_{k+N-1}^{opt}) \\ = \ & L(v_{k+N}, v_{k+N}) - L(v_k, u_k^{opt}). \end{aligned}$$

Given Condition C1 and $L(v_{k+N}, v_{k+N}) = 0$, the change in cost from time k to time $k+1$ is given by

$$\begin{aligned} V_N^{opt}(\mathcal{V}_{k+1}, \mathcal{U}_{k+1}^{opt}) - V_N^{opt}(\mathcal{V}_k, \mathcal{U}_k^{opt}) &\leq -L(v_k, u_k) \\ &\leq -\gamma(||u_k - v_k||). \end{aligned}$$

Hence, for all $x_k \in \mathbb{F}$, the state will converge to the constraint set, $\mathbb{X}_f$, as $k \to \infty$.

Note that $V_N^{fea}(\mathcal{V}_{k+1}, \mathcal{U}_{k+1}^{opt}) - V_N^{opt}(\mathcal{V}_k, \mathcal{U}_k^{opt}) = -L(v_k, u_k)$ implies that

$$V_N^{fea}(\mathcal{V}_{k+1}, \mathcal{U}_{k+1}^{opt}) \leq V_N^{opt}(\mathcal{V}_k, \mathcal{U}_k^{opt}) \tag{12}$$

for fixed horizons, as this will be used in the proof for the variable horizon case.

3.2.2 The Variable Horizon Case

We must now show that varying the horizon, N_k, will still result in the state asymptotically converging to the constraint set.

For a given horizon N_k and human input sequence $\mathcal{V}_k = \{v_k, v_{k+1}, \ldots, v_{N_k-1}\}$, let the solution to $\mathcal{P}_{N_k}$ be given by

$$\begin{aligned} \mathcal{U}_k^{opt} &= \{u_k^{opt}, u_{k+1}^{opt}, \ldots, u_{N_k-1}^{opt}\} \\ \mathcal{X}_k^{opt} &= \{x_k^{opt}, x_{k+1}^{opt}, \ldots, x_{N_k-1}^{opt}, x_{N_k}^{opt}\}, \end{aligned}$$

with $x_{N_k}^{opt} \in \mathbb{X}_f$.

We will show that at each iteration the value function is bounded above by a cost that is converging to the zero, i.e. the state will converge to the constraint set. We do, however, impose a lower bound on the horizon, namely

C5 $N_k \geq \mathcal{M} \geq 1$, for some $\mathcal{M} \in \mathbb{N}$

In other words, the horizon must never be so small that the constraint set cannot be reached by the end of the horizon.

Theorem 3.2.2 *Given Conditions C1- C5, a variable horizon, and an N_k chosen at each time k with current state x_k such that a feasible solution exists to $\mathcal{P}_{N_k}$, the state will converge to the constraint set, $\mathbb{X}_f$, as $k \to \infty$, when the first element in the optimal solution to $\mathcal{P}_{N_k}$ is applied at every time instant.*

Proof 2 *We will show the existence of feasible solutions at the time step $k+1$ and show that the value function goes to zero as k goes to infinity by proving that the value function is bounded above by a value function that converges to zero, for both expanding and contracting horizon cases.*

In the following proof, we will be introducing a reference horizon denoted as $N_r \in \mathbb{N}$ for which the notation $V_{N_r}^{ropt}(\mathcal{V}, \mathcal{U}_k^{opt})$ will denote the optimal cost given the horizon N_r and the optimal control sequence $\mathcal{U}_k^{opt}$ at time k. Also, $V_{N_r}^{rfea}(\mathcal{V}, \mathcal{U}_k^{fea})$ will denote the feasible cost given the horizon N_r and the feasible control sequence $\mathcal{U}_k^{fea}$ at time k.

By Condition C5, there exists a horizon, $\mathcal{M}$, such that $\mathcal{M} \leq N_k$ for all k, i.e. all horizons are bounded from below by $\mathcal{M}$. At time k with human input $\mathcal{V}_k = \{v_k, v_{k+1}, \ldots, v_{N_r-1}\}$, let the following be an optimal control and state sequence using a reference horizon, $N_r = \mathcal{M}$,

$$\begin{aligned} \mathcal{U}_k^{ropt} &= \{u_k^{ropt}, u_{k+1}^{ropt}, \ldots, u_{N_r-1}^{ropt}\} \\ \mathcal{X}_k^{ropt} &= \{x_k^{ropt}, x_{k+1}^{ropt}, \ldots, x_{N_r-1}^{ropt}, x_{N_r}^{ropt}\}, \end{aligned}$$

with $x_{N_r}^{ropt} \in \mathbb{X}_f$.

Let $x_{k+1}^{ropt} = f(x_k, u_k^{ropt})$. Then, at time $k+1$, a feasible control and state sequence for the reference horizon N_r given human input $\mathcal{V}_{k+1} = \{v_{k+1}, v_{k+2}, \ldots, v_{N_r}\}$ is

$$\begin{aligned} \mathcal{U}_{k+1}^{rfea} &= \{u_{k+1}^{ropt}, u_{k+2}^{ropt}, \ldots, u_{N_r-1}^{ropt}, v_{N_r}\} \\ \mathcal{X}_{k+1}^{rfea} &= \{x_{k+1}^{ropt}, x_{k+2}^{ropt}, \ldots, x_{N_r-1}^{ropt}, x_{N_r}^{ropt}, x_{N_r+1}\}, \end{aligned}$$

with $x_{N_r}^{ropt}, x_{N_r+1} \in \mathbb{X}_f$.

For both the expanding and contracting horizon cases, we will show that there is a feasible cost, at time $k+1$ for the horizon N_{k+1}, that is equal to the feasible cost given for the reference horizon, N_r, at time $k+1$,

$$V_{N_r}^{rfea}(\mathcal{V}_{k+1}, \mathcal{U}_{k+1}^{rfea}) = V_{N_{k+1}}^{fea}(\mathcal{V}_{k+1}, \mathcal{U}_{k+1}^{fea}). \tag{13}$$

We will use this fact, along with (12) to show that the feasible cost for the reference horizon is an upper bound to the optimal cost for horizon N_k. First, we show that (13) holds for both the expanding and contracting horizon cases.

Case 1*: $N_k \leq N_{k+1}$ (Expanding Horizon)*

A feasible control and state sequence for the human input $\mathcal{V}_k = \{v_k, v_{k+1}, \ldots, v_{N_r-1}, \ldots, v_{N_k-1}\}$ at time k is given by

$$\begin{aligned}
\mathcal{U}_k^{fea} &= \{u_k^{ropt}, u_{k+1}^{ropt}, \ldots, u_{N_r-1}^{ropt}, v_{N_r}, \ldots, v_{N_k-1}\} \\
\mathcal{X}_k^{fea} &= \{x_k^{ropt}, x_{k+1}^{ropt}, \ldots, x_{N_r-1}^{ropt}, x_{N_r}^{ropt}, \\
&\quad \ldots \ , x_{N_k-1}, x_{N_k}, \},
\end{aligned}$$

with $\{x_{N_r}^{ropt}, \ldots, x_{N_k-1}, x_{N_k}, \} \subset \mathbb{X}_f$, assuming the control $v_{N_r}, \ldots, v_{N_k-1}$ provides $\mathbb{X}_f$ with invariance as given by C3.

Let $x_{k+1}^{ropt} = f(x_k, u_k^{ropt})$, then, a feasible control and state sequence given $\mathcal{V}_{k+1} = \{v_{k+1}, \ldots, v_{N_r}, \ldots, v_{N_k-1}, \ldots v_{N_{k+1}-1}\}$ is

$$\begin{aligned}
\mathcal{U}_{k+1}^{fea} &= \{u_{k+1}^{ropt}, \ldots, u_{N_r-1}^{ropt}, v_{N_r}, \ldots, v_{N_k-1}, \\
&\quad \ldots \ , v_{N_{k+1}-1}, \} \\
\mathcal{X}_{k+1}^{fea} &= \{x_{k+1}^{ropt}, \ldots, x_{N_r-1}^{ropt}, x_{N_r}^{ropt}, \ldots, x_{N_k}, \ldots, x_{N_{k+1}}\},
\end{aligned}$$

with $x_{N_r}^{ropt}, \ldots, x_{N_k}, \ldots, x_{N_{k+1}} \in \mathbb{X}_f$. Therefore

$$V_{N_r}^{rfea}(\mathcal{V}_{k+1}, \mathcal{U}_{k+1}^{rfea}) = V_{N_{k+1}}^{fea}(\mathcal{V}_{k+1}, \mathcal{U}_{k+1}^{fea}),$$

because

$$\begin{aligned}
V_{N_r}^{rfea}(\mathcal{V}_{k+1}, \mathcal{U}_{k+1}^{rfea}) &= L(v_{k+1}, u_{k+1}^{ropt}) + \\
&\quad \cdots + L(v_{N_r-1}, u_{N_r-1}^{ropt}) \qquad (14)
\end{aligned}$$

and

$$
\begin{aligned}
& V^{fea}_{N_{k+1}}(\mathcal{V}_{k+1}, \mathcal{U}^{fea}_{k+1}) = \\
= \;& L(v_{k+1}, u^{ropt}_{k+1}) + \cdots + L(v_{N_r-1}, u^{ropt}_{N_r-1}) \\
+ \;& + L(v_{N_r}, v_{N_r}) + \cdots + L(v_{N_k-1}, v_{N_k-1}) \\
+ \;& \cdots + L(v_{N_{k+1}-1}, v_{N_{k+1}-1}) \\
= \;& L(v_{k+1}, u^{ropt}_{k+1}) + \cdots + L(v_{N_r-1}, u^{ropt}_{N_r-1}),
\end{aligned}
$$

where $L(v_{N_r}, v_{N_r}) + \cdots + L(v_{N_k-1}, v_{N_k-1}) + \cdots + L(v_{N_{k+1}-1}, v_{N_{k+1}-1}) = 0$ *by C2 and C3.*

Case 2*:* $N_k \geq N_{k+1}$ *(Contracting Horizon)*

A feasible control and state sequence at time k *is given by*

$$
\begin{aligned}
\mathcal{U}^{fea}_k &= \{u^{ropt}_k, u^{ropt}_{k+1}, \ldots, u^{ropt}_{N_r-1}, v_{N_r}, \ldots, v_{N_{k+1}-1}, \\
&\ldots \quad v_{N_k-1}\} \\
\mathcal{V}_k &= \{v_k, v_{k+1}, \ldots, v_{N_r-1}, \ldots, v_{N_{k+1}-1}, \ldots v_{N_k-1}\} \\
\mathcal{X}^{fea}_k &= \{x^{ropt}_k, x^{ropt}_{k+1}, \ldots, x^{ropt}_{N_r-1}, x^{ropt}_{N_r}, \\
&\ldots \quad , x_{N_{k+1}}, \ldots, x_{N_k}, \},
\end{aligned}
$$

with $\{x^{ropt}_{N_r}, \ldots, x_{N_{k+1}}, \ldots, x_{N_k}, \} \subset \mathbb{X}_f$ *by C3.*

A feasible control and state sequence using the contracted horizon at time $k+1$*,* N_{k+1}*, is given by*

$$
\begin{aligned}
\mathcal{U}^{fea}_{k+1} &= \{u^{ropt}_{k+1}, \ldots, u^{ropt}_{N_r-1}, v_{N_r}, \ldots, v_{N_{k+1}-1}\} \\
\mathcal{V}_{k+1} &= \{v_{k+1}, \ldots, v_{N_r-1}, \ldots, v_{N_{k+1}-1}\} \\
\mathcal{X}^{fea}_{k+1} &= \{x^{ropt}_{k+1}, \ldots, x^{ropt}_{N_r-1}, x^{ropt}_{N_r}, \ldots, x_{N_{k+1}}\},
\end{aligned}
$$

with $\{x^{opt}_{N_r}, \ldots, x_{N_{k+1}}, \} \subseteq \mathbb{X}_f$*.*

From this, we have

$$\begin{aligned} & V^{fea}_{N_{k+1}}(\mathcal{V}_{k+1},\mathcal{U}^{fea}_{k+1}) = \\ = \;& L(v_{k+1},u^{ropt}_{k+1}) + \cdots + L(v_{N_r-1},u^{ropt}_{N_r-1}) \\ + \;& L(v_{N_r},v_{N_r}) + \cdots + L(v_{N_{k+1}-1},v_{N_{k+1}-1}) \\ = \;& L(v_{k+1},u^{ropt}_{k+1}) + \cdots + L(v_{N_r-1},u^{ropt}_{N_r-1}), \end{aligned}$$

where $L(v_{N_r},v_{N_r}) + \cdots + L(v_{N_{k+1}-1},v_{N_{k+1}-1}) = 0$. *This along with (14) gives*

$$V^{rfea}_{N_r}(\mathcal{V}_{k+1},\mathcal{U}^{rfea}_{k+1}) = V^{fea}_{N_{k+1}}(\mathcal{V}_{k+1},\mathcal{U}^{fea}_{k+1}).$$

Thus, we have shown (13) for both the expanding and contracting horizon cases. In addition, the optimality of $\mathcal{U}^{opt}_{k+1}$, *i.e.*

$$V^{fea}_{N_{k+1}}(\mathcal{V}_{k+1},\mathcal{U}^{fea}_{k+1}) \geq V^{opt}_{N_{k+1}}(\mathcal{V}_{k+1},\mathcal{U}^{opt}_{k+1}),$$

results in

$$V^{rfea}_{N_r}(\mathcal{V}_{k+1},\mathcal{U}^{rfea}_{k+1}) \geq V^{opt}_{N_{k+1}}(\mathcal{V}_{k+1},\mathcal{U}^{opt}_{k+1}).$$

Since N_r *is a fixed horizon, we utilize (12) to show that*

$$\begin{aligned} V^{ropt}_{N_r}(\mathcal{V}_k,\mathcal{U}^{ropt}_k) &\geq V^{rfea}_{N_r}(\mathcal{V}_{k+1},\mathcal{U}^{rfea}_{k+1}) \\ &\geq V^{opt}_{N_{k+1}}(\mathcal{V}_{k+1},\mathcal{U}^{opt}_{k+1}). \end{aligned} \tag{15}$$

Given Theorem 3.2.1 and that N_r *is fixed, the cost sequence* $V^{ropt}_{N_r}(\mathcal{V}_k,\mathcal{U}^{ropt}_k)$ *converges to zero as* $k \to \infty$. *Since,* $V^{ropt}_{N_r}(\mathcal{V}_k,\mathcal{U}^{ropt}_k) \geq V^{rfea}_{N_r}(\mathcal{V}_{k+1},\mathcal{U}^{rfea}_{k+1})$ *for all* k, *then,* $V^{rfea}_{N_r}(\mathcal{V}_{k+1},\mathcal{U}^{rfea}_{k+1})$ *also converges to zero as* $k \to \infty$. *This is illustrated in Figure 3.*

By (15), $V^{rfea}_{N_r}(\mathcal{V}_{k+1},\mathcal{U}^{rfea}_{k+1})$ *is an upper bound to* $V^{opt}_{N_{k+1}}(\mathcal{V}_{k+1},\mathcal{U}^{opt}_{k+1})$. *Given (2), the minimum cost using any horizon is zero, implying that the state is in the constraint set,* $\mathbb{X}_f$, *for the duration of the sequence,* k *to* $k+N_k$. *Hence,* $V^{opt}_{N_{k+1}}(\mathcal{V}_{k+1},\mathcal{U}^{opt}_{k+1})$ *also has a lower bound of zero. As a result,*

$$\begin{aligned} \lim_{k\to\infty} \sup V^{opt}_{N_{k+1}}(\mathcal{V}_{k+1},\mathcal{U}^{opt}_{k+1}) &= \\ \lim_{k\to\infty} V^{rfea}_{N_r}(\mathcal{V}_{k+1},\mathcal{U}^{rfea}_{k+1}) &= 0, \\ & \textit{and} \\ \lim_{k\to\infty} \inf V^{opt}_{N_{k+1}}(\mathcal{V}_{k+1},\mathcal{U}^{opt}_{k+1}) &= 0. \end{aligned}$$

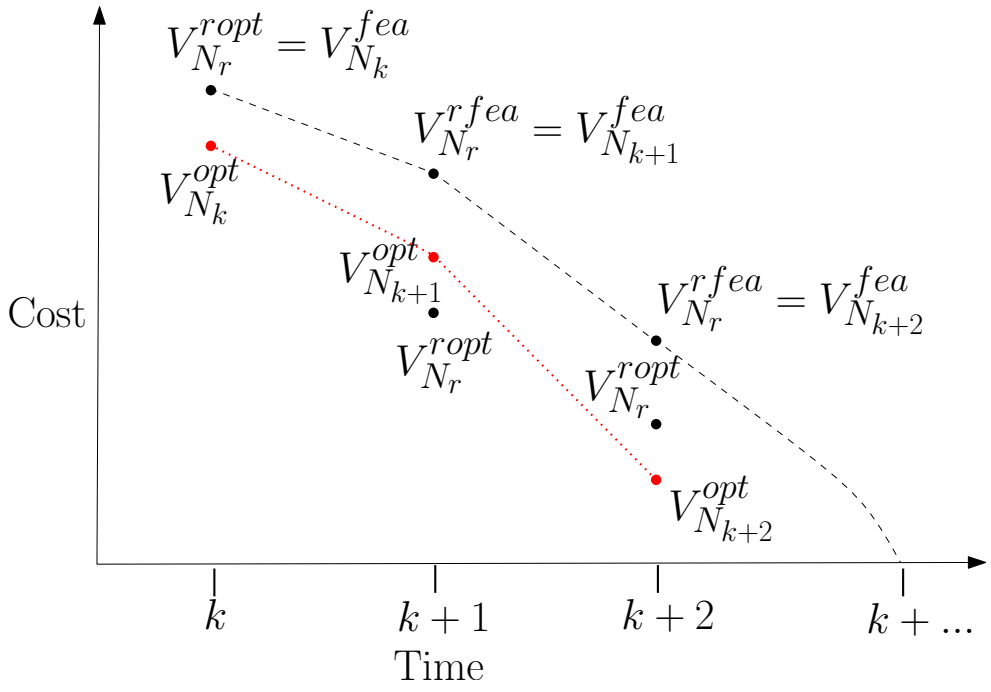


Figure 3: Illustration of the variable horizon proof where the optimal cost for any horizon N_k is bounded above by the feasible cost for the reference horizon N_r. This feasible cost is bounded above by the optimal cost for the reference horizon at the previous time step and the optimal cost for the reference horizon is known to asymptotically converge to the minimum.

Therefore, the cost converges to zero for all horizons greater than or equal to the reference horizon. As a consequence, the cost associated with the reference horizon bounds the optimal cost associated with either an expanding or contracting horizon since $N_k \geq N_r$ for all k. In other words, we are guaranteed asympototic convergence for any combination of horizon expansion and contraction given C5.

It should be noted here that the value function at every time instant is not necessarily smaller than the value function at the previous time instant. Hence, it is possible that the state temporarily moves away from the constraint set. However, for the expanding horizon case, it can be shown that the value function is monotonically decreasing at every time instant.

3.3 Conclusions

In this chapter, we have developed a control framework with asymptotic convergence to the constraint set. This means we have guarantees about accomplishing the low-level task we have assigned the robot to complete. In addition, there is still freedom in the way that low-level task is accomplished in that the human input can influence what states to reach in the constraint set as well as how the robot reaches the constraint set. In other words, human operator input influences how the low-level task is accomplished while having the leeway to complete a high-level task. In the next chapter, we turn our attention to how to make the human input predictions prescribed in the framework, and we discuss how those predictions benefit from varying the horizon. We also present a control law that solves the optimal control problem at every time instant given linear system dynamics.

CHAPTER IV

HORIZONS, PREDICTIONS, AND CONTROL

In the last chapter, we developed a control framework with proofs showing low-level task completion. However, within this framework, methods for making human input prediction, choosing control horizons, and solving the optimal control problem are still needed in order to implement the control. We discuss such methods in this chapter which begins with a discussion on input prediction followed by a varying horizon method to increase input prediction performance and finally a control law for a specific class of systems dynamics and constraints. Then, in the last part of the chapter, we apply these methods to some example human-robot tasks.

In the first example, we demonstrate how a fixed horizon, a linear extrapolation prediction method, and a closed-form control law can be used to synchronize pendula swinging while still allowing high-level human control. This example also set the foundation for multi-operator scenarios discussed in Chapter 7. In the second example, we implement different horizon/prediction methods on a shared mobile robot control task where the human-robot team must drive to a goal line. This example sets the foundation for the operator studies presented in Chapter 5.

4.1 Human Input Prediction Methods

As already seen, we need to be able to predict human inputs in order to compute the optimal solution. In this section, we discuss such prediction methods, although we note that other such methods are conceivable. One set of methods is Zero-Order Hold (ZOH), where only the current human input is needed to make the prediction, and First-Order Hold (FOH), where the previous and current inputs are used. The second method is prediction by Least Squares system identification (SID), which requires that we store a certain number of past human inputs. As a result, there is an initialization period required to accumulate past human input values before system identification can be used. Therefore, ZOH will be used

in conjunction with SID in order to make predictions while previous human inputs are still being accumulated.

These are clearly not the only approaches available for human prediction, but they represent two types of approaches that can be used: high frequency input with low computation versus low frequency with high computation. The more complex method tries to predict longer term trends in human input but require more computation, while the other method is quick to compute and handles predictions based on high frequency changes in human input. We investigate the differences in terms for human operator experience in Chapter 5.

4.1.1 Zero-Order Hold Prediction (ZOH)

Since only the current human input is used to make future human input predictions, this method can be used as soon as the task begins. Given v_k, the predicted human input sequence is given by

$$\mathcal{V}_k = \{v_k, v_k, \ldots, v_k\}.$$

This prediction is simple and will be shown to be effective in experimentation. The computational and memory demands of this method are lower in comparison to the system identification method that requires matrix computations and storage of many past human inputs.

4.1.2 First-Order Hold (FOH)

Since only the current and previous human inputs are used to make future human input predictions, this method can be used as soon as the task begins (initially assuming the previous human input was zero). The human input prediction sequence, $\mathcal{V}_k$, is given by linearly extrapolating the previous and current human input. Given v_k, v_{k-1}, and $\sigma_k = v_k - v_{k-1}$, the predicted human input sequence is given by

$$\mathcal{V}_k = \{v_k, v_k + \sigma_k, \ldots, v_k + (N_k - 1)\sigma_k\}.$$

Similar to ZOH, this prediction is simple and will be shown to be effective in both simulation and experimentation. This method is also effective for hardware implementations with low computational power or memory.

4.1.3 Least-Squares System Identification (SID)

Human input predictions using linear least-squares system identification allows us to make predictions that reflect longer-term trends in the human input than prediction by linear extrapolation allows for. For instance, a system indentification approach would make better predictions of periodic human inputs, such as sinusoidal inputs. This method not only serves as a way to use past information to predict future human inputs, but we will be able to gauge the performance of this prediction and update the variable horizon accordingly.

At time k, let the $N_s \in \mathbb{N}$ past and current human input values be denoted as $s_k = [v_{k-N_s}, \ldots, v_k]^T$ and

$$H_k = \begin{bmatrix} v_{k-N_s-N_s} & \cdots & v_{k-N_s} \\ \vdots & \vdots & \vdots \\ v_{k-2-N_s} & \cdots & v_{k-2} \\ v_{k-1-N_s} & \cdots & v_{k-1} \end{bmatrix},$$

with model parameters,

$$\phi^T = \begin{bmatrix} \phi_{N_s} \\ \vdots \\ \phi_2 \\ \phi_1 \end{bmatrix}. \tag{16}$$

The least squares problem, $s_k = H_k \phi^T$, is then solved, resulting in

$$\phi = (H_k^T H_k)^{-1}(s_k^T H_k). \tag{17}$$

However, the quantity, $(H_k^T H_k)$ is a potentially singular matrix, so the Levenberg-Marquardt procedure [52] is used to regularize this matrix. This results in

$$\phi = (H_k^T H_k + \delta I)^{-1}(s_k^T H_k), \tag{18}$$

for some small $\delta \in \mathbb{R}$. The collection of past, current, and one time step in the future human

inputs is given by $s_{k+1} = [v_{k-N_s+1}, \ldots, v_k + 1]^T = \Gamma s_k$, where

$$\Gamma = \begin{bmatrix} 0 & & \\ \vdots & I & \\ 0 & & \\ \phi_{N_s} & \cdots & \phi_1 \end{bmatrix} \in \mathbb{R}^{N_s \times N_s}.$$

From the last row of that matrix expression, we have the predicted human input one time step into the future, $v_{k+1} = \phi s_k$.

Similarly, the predicted human input two time steps in the future is made using $s_{k+2} = \Gamma s_{k+1}$ and getting $v_{k+2} = \phi s_{k+1}$. Repeating this procedure $N_k - 1$, the future human input sequence is produced, $\mathcal{V}_k = \{v_k, v_{k+1}, \ldots, v_{k+N_k}\} = \{v_k, \phi s_k, \ldots, \phi s_{k+N_k-1}\}$.

4.2 Choosing A New Horizon

As already mentioned, there is an inherent trade-off between prediction horizon and prediction quality. As such, we may want to be able to dynamically adjust the horizon as a function of the prediction quality.

In order to measure the performance of the human prediction at time k, we propose to utilize the system identification model obatined at the current time and produce a human input signal backwards in time for the length of the current control horizon. The performance measure is a cost on the deviation from this signal to the actual human input signal recorded over that time. If the deviations are large, then the predicted model is not accurate (i.e. not performing well) and the horizon should be shortened. On the other hand, if the deviations are small, then the human input model is performing well and the horizon can be increased. Hence, we want to find the horizon that minimizes the deviations in the predicted and actual human input signals based on past data. For more details on this method for choosing control horizons, see [25]

For computational reasons, we formulate this problem in continuous time where t is continuous time with Δ being the continuous time horizon in seconds (i.e. $\Delta = N_k dt$, where dt is the discrete time step). Furthermore, $v(t)$ is the human input signal, and $\hat{v}(t, s)$ is the predicted human input at time s using the system identification model obtained at

time t.

Therefore, we will minimize the following cost,

$$J(t, \Delta) = -\int_{t}^{t-\Delta} \mathcal{F}(v(s), \hat{v}(t, s))ds + \mathcal{G}(\Delta),$$

The purpose of $\mathcal{F}(v(s), \hat{v}(\tau, s))$ is to punish deviations between the predicted human input, $\hat{v}(\tau, s)$, and the actual human input, $v(s)$, while $\mathcal{G}(\Delta)$ is meant to punish small values of Δ. In other words, $\mathcal{G}(\Delta)$ does not allow the horizon to be zero, where the deviation in predicted and actual user input is clearly zero. Examples of this include,

$$\mathcal{F}(v(s), \hat{v}(\tau, s)) = \frac{1}{2}\gamma(v(s) - \hat{v}(\tau, s))^2,$$
$$\mathcal{G}(\Delta) = \frac{\rho}{\Delta},$$

where the quantity, $\hat{v}(t, t-\Delta)$, is computed using the current system identification and generating human input backwards in time, and ρ,γ are scalar weighting values. This specific cost was chosen to penalize differences between predicted and actual human input by the squared norm of this difference, which gives us a quadratic cost. The terminal cost was chosen to penalize small horizons, as without this, the system would choose a horizon of the smallest length. Small horizons make this quantity large, so there is some balance between large and small horizons. This particular method allows us to evaluate how effective the current system identification-based prediction is performing.

We utilize a gradient descent method to minimize the cost and we thus derive the gradient for this cost function with respect to the horizon, Δ

Using the Liebniz Integration rule, we arrive at

$$\frac{\partial J}{\partial \Delta}(t, \Delta) = \frac{d\mathcal{G}}{d\Delta}(\Delta) + \mathcal{F}(v(t-\Delta), \hat{v}(t, t-\Delta))$$

An Armijo step-size gradient descent algorithm (see [25]) can be employed to find the horizon that minimizes J. Using this, we can now update the horizon used in the human-in-the-loop controller along with the system identification.

4.3 Control Law Derivation

The framework proposed in this thesis utilizes model predictive control, and this involves solving an optimal control problem at each time instant. This is potentially computationally intensive and in some instances infeasible. Therefore, closed-form solutions to optimal control problems have great utility. The following section details such a closed-form controller for the specific class of our problem, namely a linear dynamic system with a linear state constraint. This control law is used in the experiments detailed in the following sections, greatly reducing the computational burden associated with using MPC. Hence, we can run our algortihm at reasonable control bandwidths, especially when combined with the computation required for SID and the variable horizon algorithm.

Specifically, the following derivation will address a version of $\mathcal{P}_{N_k}$, where the dyanmics are linear ($x_{k+1} = Ax_k + Bu_k$) and the task is modeled by the constraint, $x(k+N) \in \mathbb{X}_f = \{x \mid Mx = b\}$ We begin by letting the human input and control input sequences over the horizon, N_k, be in the Hilbert space, $l_2^m[k, k+N_k-1]$, which we will denote as l_2 from now on. Specifically, l_2 is the space of square summable sequences with cardinality $N_k - 1$ with elements in $\mathbb{R}^m$.

Let $\mathcal{H} = l_2$ with the inner product given as $\langle y, w\rangle = \sum_{i=k}^{k+N_k-1} y_i^T w_i$ for all $y, w \in \mathcal{H}$ and the norm squared defined as $||y||^2 = \langle y, y\rangle = \sum_{i=k}^{k+N_k-1} y_i^T y_i$. Hence,

$$\mathcal{P}_{N_k} = \min_{\mathcal{U}_k} ||\mathcal{V}_k - \mathcal{U}_k||^2 \tag{19}$$

is a minimum norm problem, subject to (6) and (7), to which we can apply Hilbert's projection theorem.

The following general steps are taken to find the projection of the human input signal, a point $\mathcal{V}_k \in \mathcal{H}$, onto an affine variety, $\mathbb{V}_\alpha$, representing the space of control inputs for which the terminal state constraint is satisfied. First, we must find a subspace of $\mathbb{V}_0 \in \mathcal{H}$, that is parallel to $\mathbb{V}_\alpha$. We then find the subspace orthogonal to $\mathbb{V}_0$, which is also orthogonal to $\mathbb{V}_\alpha$ and translate that subspace so that it passes through point $\mathcal{V}_k$ (the predicted human inputs). The point that lies in both $\mathbb{V}_\alpha$ and the translated orthogonal space is the unique minimizer. This is shown graphically in Figure (4) as in [90]. For this to work we need to

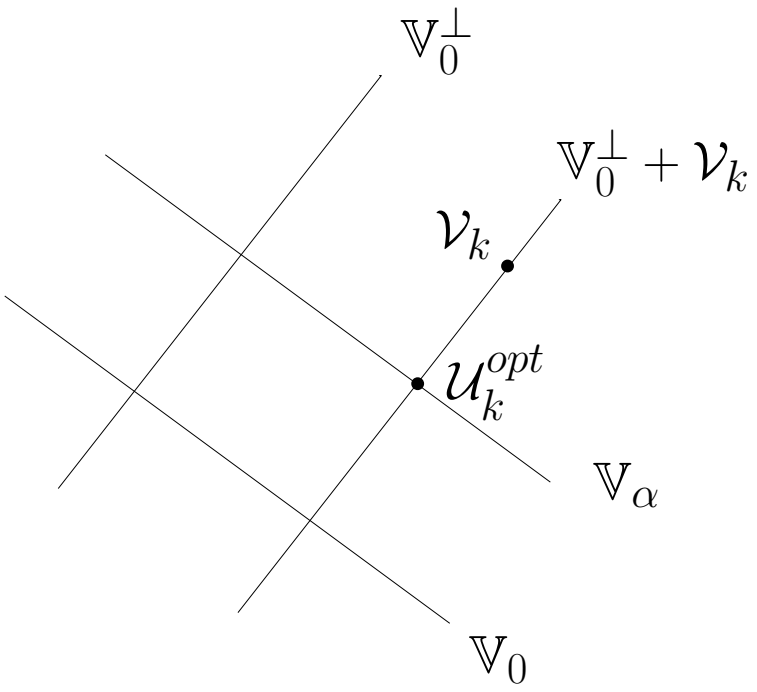


Figure 4: Hilbert space projection of $\mathcal{V}_k$ onto the goal subspace $\mathbb{V}_\alpha$ to find the unique minimizing control $\mathcal{U}_k^{opt}$

define the constraint space as an affine variety in $\mathcal{H}$. The state at the end of the control horizon N_k is given by

$$x_{k+N_k} = A^{N_k} x_k + \sum_{i=k}^{k+N_k-1} A^{k+N_k-1-i} B u_i \tag{20}$$

and it is required that this state satisfies the linear terminal constraint (7). The linear operator $\mathcal{L} : \mathcal{H} \to \mathbb{R}^n$ is defined such that

$$\mathcal{L}\mathcal{U}_k = \sum_{i=k}^{k+N_k-1} A^{k+N_k-1-i} B u_i,$$

i.e.,

$$x_{k+N_k} = A^{N_k} x_k + \mathcal{L}\mathcal{U}_k.$$

Hence, we can rewrite the terminal constraint, $M x_{k+N_k} = b$, as

$$M\mathcal{L}\mathcal{U}_k = b - MA^{N_k} x_k. \tag{21}$$

Let $\mathcal{L}^* : \mathbb{R}^n \to \mathcal{H}$ denote the adjoint operator

$$\mathcal{L}^* = \{B^T (A^{N_k-1})^T, B^T (A^{N_k-2})^T, \ldots, B^T\}$$

Furthermore, the grammian $\mathcal{L}\mathcal{L}^*$ is given by

$$\mathcal{L}\mathcal{L}^* = \sum_{i=k}^{k+N_k-1} A^{k+N_k-1-i} B B^T (A^{k+N_k-1-i})^T.$$

Next, (21) will be used to construct a subspace and the corresponding affine variety. Let $\mathbb{V}_0$ be defined as a subspace of $\mathcal{H}$ and $\mathbb{V}_\alpha$ be the affine variety such that

$$\mathbb{V}_0 = \{\mathcal{U}_k \in \mathcal{H} \mid M\mathcal{L}\mathcal{U}_k = 0\},$$

and

$$\mathbb{V}_\alpha = \{\mathcal{U}_k \in \mathcal{H} \mid M\mathcal{L}\mathcal{U}_k = \alpha\},$$

with $\alpha = b - MA^{N_k}x_k$.

Next, we will find a subspace orthogonal to $\mathbb{V}_0$ and $\mathbb{V}_\alpha$. The orthogonal complement $\mathbb{V}_0^\perp$ to $\mathbb{V}_0$ is

$$\mathbb{V}_0^\perp = \{s \in \mathcal{H} \mid \langle \mathcal{U}_k, s\rangle = 0,\ \forall\, \mathcal{U}_k \in \mathbb{V}_0\}.$$

$\mathbb{V}_0^\perp$ is obtained by letting d be some point in $\mathbb{R}^l$ such that

$$0 = \langle M\mathcal{L}\mathcal{U}_k, d\rangle_{\mathbb{R}^l} = \langle \mathcal{U}_k, \mathcal{L}^* M^T d\rangle_{l_2},$$

i.e.,

$$\mathbb{V}_0^\perp = \{s \in \mathcal{H} \mid s = \mathcal{L}^* M^T d,\ d \in \mathbb{R}^l\}.$$

The orthogonal complement can be translated by $\mathcal{V}_k$, giving

$$\mathbb{V}_0^\perp + \mathcal{V}_k = \{w \in \mathcal{H} \mid w = s + \mathcal{V}_k,\ s \in \mathbb{V}_0^\perp\}$$

such that

$$\mathbb{V}_0^\perp + \mathcal{V}_k = \{w \in \mathcal{H} \mid w = \mathcal{L}^* M^T d + \mathcal{V}_k,\ d \in \mathbb{R}^l\}.$$

Now, to find the unique minimizer to (19), the intersection of $\mathbb{V}_0^\perp + \mathcal{V}_k$ and $\mathbb{V}_\alpha$ gives

$$\begin{aligned} M\mathcal{L}w &= b - MA^{N_k}x_k \qquad (22)\\ M\mathcal{L}(\mathcal{L}^* M^T d + \mathcal{V}_k) &= b - MA^{N_k}x_k \\ M\mathcal{L}\mathcal{L}^* M^T d &= b - MA^{N_k}x_k - M\mathcal{L}\mathcal{V}_k. \end{aligned}$$

So, we can solve for d with

$$d = (M\mathcal{L}\mathcal{L}^* M^T)^{-1}(b - MA_k^N x_k - M\mathcal{L}\mathcal{V}_k). \tag{23}$$

Therefore, plugging (23) back into (22) and using (21), the optimal control sequence is given by

$$\mathcal{U}_k^{opt} = \mathcal{L}^* M^T (M\mathcal{L}\mathcal{L}^* M^T)^{-1}(b - MA^{N_k} x_k - M\mathcal{L}\mathcal{V}_k) + \mathcal{V}_k$$

For the receding horizon formulation, only the first element of $\mathcal{U}_k^{opt}$ is applied to (6). Thus, the optimal control law at time k is

$$u_k^{opt} = \mathcal{L}_k^* M^T (M\mathcal{L}\mathcal{L}^* M^T)^{-1}(b - MA^{N_k} x_k - M\mathcal{L}\mathcal{V}_k) + v_k. \tag{24}$$

This control law minimizes the predicted cost over the horizon so we state this as a theorem:

Theorem 4.3.1 *Given the terminal constraint receding finite horizon optimal control problem, $\mathcal{P}_{N_k}$, and the predicted human input sequence, $\mathcal{V}_k$, the optimal control law is given by (24).*

This gives a closed-form solution to the optimal control problem that needs to be solved every time step, instead of solving a potentially computationally intensive constrained quadratic program numerically. Also note that this closed form solution is valid for any N_k so it is applicable for both fixed and varible horizons.

We have presented methods for human input prediction as well as a method to vary the horizon to facilitate human input prediction. A closed-form control law was also developed for linear systems with linear low-level state constraints. Now that all aspects of the control framework can be computed, we apply this technique to some example problems in the next section to evaluate the viability of the developed methods.

The first example is the task of synchronizing the swinging of two mass-cart-pendula while driving the carts to a particular position. This is presented as a simulation of the pendula with actual human operators, and the example demonstrates the low and high level capabilities of the control framework. The example also shows that a fixed horizon

and a relatively simple human input prediction method can be effective in this control framework. This example incorporates two human operators and lays the foundation for the multi-operator work presented later in this thesis.

The second example is a human-robot navigation task to investigate the viability of the variable horizon with system identification human input prediction techniques. The human is tasked with guiding a robot to a particular position on a goal line, while the low-level robot task is to ensure the robot reaches the goal line. This example serves as a foundation for a more complex navigation task studied in Chapter 5, where operator performance and workload are studied for different horizon and input prediction methods.

4.4 Example: Human Operation of Simulated Mass-Cart-Pendula Synchronization

We apply the developed approach to a simulation of a two mass-cart-pendula synchronization problem under the command of a human operator. The human operator issues force commands to one mass-cart-pendulum, while another human issues commands to the other. The human commands have saturation limits (input constraints) while the automatic control effort does not as specified in C3.

The human operators are tasked with driving the pendula to a desired location (i.e. the high-level task) while the controller task is to ensure cart separation and pendula oscillation synchronization (i.e. the low-level task) as in Figure 5. The operators visually monitor the progress of the system through a graphic display as seen in Figure 5. In the following, the system dynamics and low-level task constraints are detailed.

4.4.1 Mass-Cart Pendula Dynamics and Synchronization Constraints

As seen in Figure 6, the force, F applied in the P_x direction, is the control input, u, to the single pendulum system. No damping force is considered in this model as pendula can be approximated as zero damping systems. The linearized continuous dynamics about the hanging equilibrium point, gives the single pendulum system for pendulum $i = 1, 2$, as

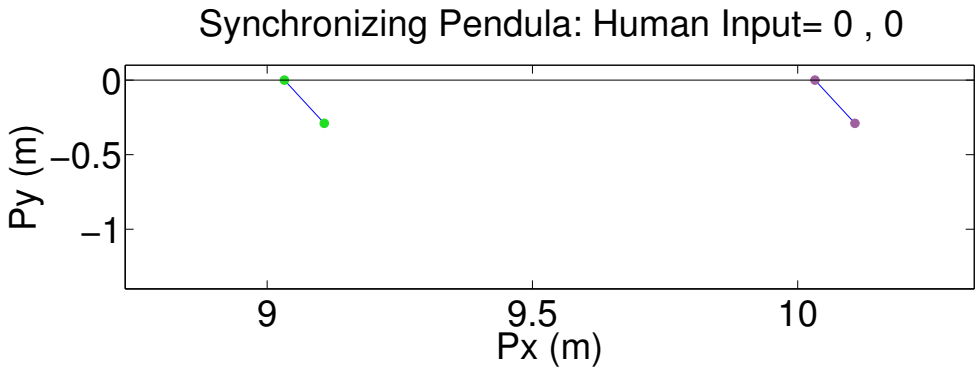


Figure 5: Pendula Graphic Display showing inter-cart distance and oscillation synchronization. The carts lie on the $Py = 0$ line while the pendula are shown mid-swing.

$\dot{x}_i(t) = \tilde{A}_i x_i(t) + \tilde{B}_i u_i(t)$, where:

$$x_i = \begin{bmatrix} P_{x,i} \\ \dot{P}_{x,i} \\ \theta_i \\ \dot{\theta}_i \end{bmatrix}, \tilde{A}_i = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{-(M+m)g}{Ml} & 0 \end{bmatrix}, \tilde{B}_i = \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{-1}{Ml} \end{bmatrix}.$$

Note that this pair, $(\tilde{A}_i, \tilde{B}_i)$, is controllable.

For a 2 planar-pendula system, the dynamics can be written as $\dot{x}(t) = \tilde{A}x(t) + \tilde{B}u(t)$, where

$$\begin{aligned} x^T &= \begin{bmatrix} x_1^T & x_2^T \end{bmatrix}, u = \begin{bmatrix} u_1 & u_2 \end{bmatrix}^T, \\ \tilde{A} &= \begin{bmatrix} \tilde{A}_1 & 0 \\ 0 & \tilde{A}_2 \end{bmatrix}, \tilde{B} = \begin{bmatrix} \tilde{B}_1 & 0 \\ 0 & \tilde{B}_2 \end{bmatrix}. \end{aligned}$$

Note that $\tilde{A}_1 = \tilde{A}_2$ and $\tilde{B}_1 = \tilde{B}_2$, since the pendula are assumed to be homogeneous. The system is discretized to

$$x_{k+1} = Ax_k + Bu_k \tag{25}$$

with $x_k \in \mathbb{R}^8$ and $u_k \in \mathbb{R}^2$.

When the mass-cart-pendula are synchronized, the mass-carts are positioned a specified distance apart with identical cart velocity, while the pendula have the same angle and angular

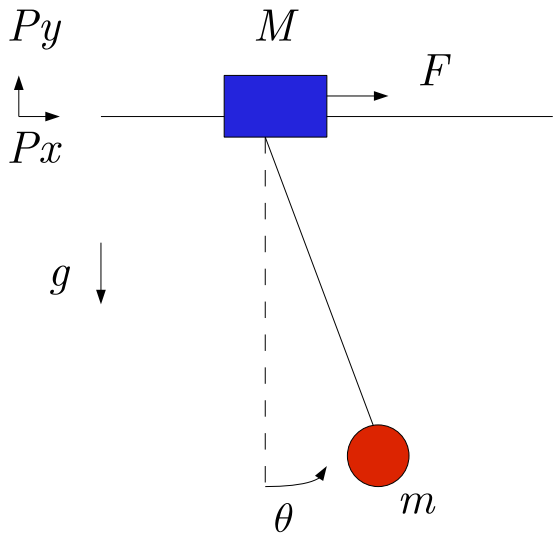


Figure 6: Single Pendulum Diagram

velcocity. Therefore, the synchronization constraints are $P_{x,1} - P_{x,2} = d$, $\dot{P}_{x,1} - \dot{P}_{x,2} = 0$, $\theta_1 - \theta_2 = 0$, $\dot{\theta}_1 - \dot{\theta}_2 = 0$, i.e., $\mathbb{X}_f = \{x \mid Cx = b\}$, where

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix}, \; b = \begin{bmatrix} d \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

4.4.2 Results

The pendula dynamics are simulated with the nonlinear system dynamics but the control is calculated using the linearized system dynamics. The following simulations were run with parameters: $d = 1\ m$, $l = 0.3\ m$, $m = 2\ kg$, $M = 3\ kg$, $g = 9.8\ m/s^2$, $N = 1.0\ s$, with a sample time of $0.1\ s$.

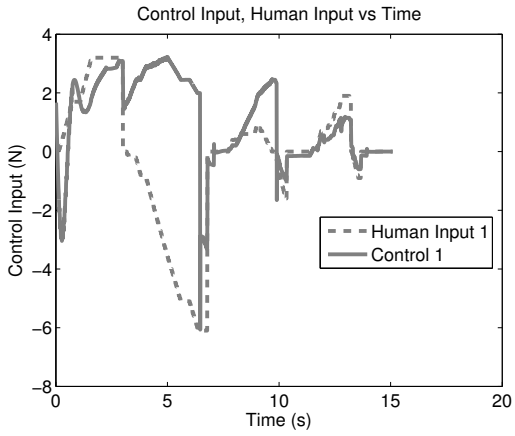
The following results were generated with two human operators who input force commands by using keyboard arrow keys to increment or decrement the force in 0.1 N increments. The inputs have a maximum of $10N$ and a minimum of $-10N$, and operators are able to zero the force input. The control law, (24), is applied to (25) with $\mathcal{V}_k$ given by the linear extrapolation method presented earlier in this chapter. The fixed horizon/linear extrapolation version of the controller is used for this example.

Figures 7 and 8 contain a set of plots resulting from two human subjects attempting to drive the mass-cart-pendula so that the position of the right-most pendulum was approximately at the $10m$ mark.

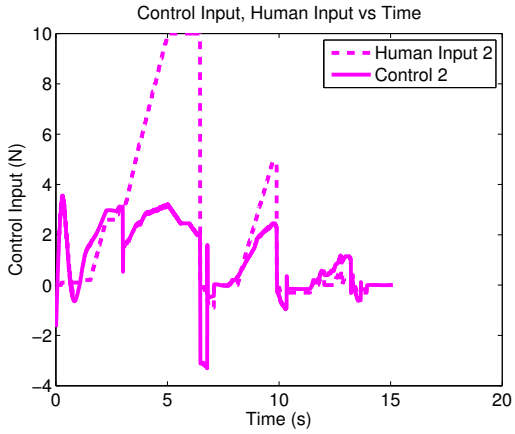
In Figures 7(c) and 8, the state converges to the constraint set within $2s$. In Figures 7(a) and 7(b), we can see how the control effort deviates from the human control input. In Figure 7(a), the control responds to the input given by Human Operator 2 at the $5s$ mark and swings away from the Human Operator 1 input. In other words, while Human Operator 1 tries to move Pendulum 1 to the right, the control moves Pendulum 1 to the left so that synchronization is maintained. This of course will be disconcerting for the human operator during operation.

In Figure 8(a), the human operators were able to drive the right mass-cart-pendulum to the $10m$ position. The plots in Figures 8(a)-8(d) show that the system converges to the linear constraint where both pendula oscillations are synchronized and the carts distances are a fixed distance apart. In addition, the human operators can drive the synchronized pendula left or right, albeit not with the immediacy that direct control would allow for.

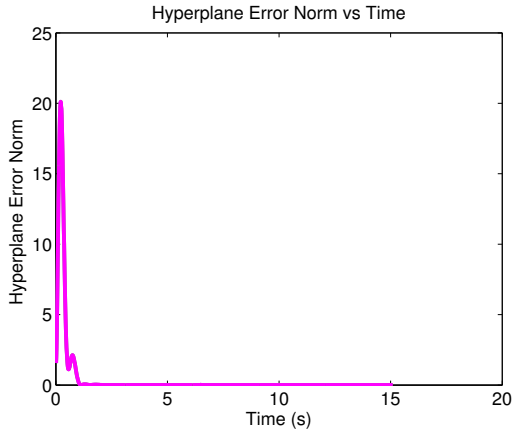
In summary, the control framework is effective in that the system is driven to both synchronize the pendula (low-level task) as well as being driven to a specific cart position (high-level task). This task is quite difficult without the help of the shared control. We also see that the linear extrapolation method for human input prediction is effective over the chosen fixed horizon. We now present an example that examines different horizon and human input prediction methods for a shared navigation task.



(a) Human 1 input vs Control 1 input.

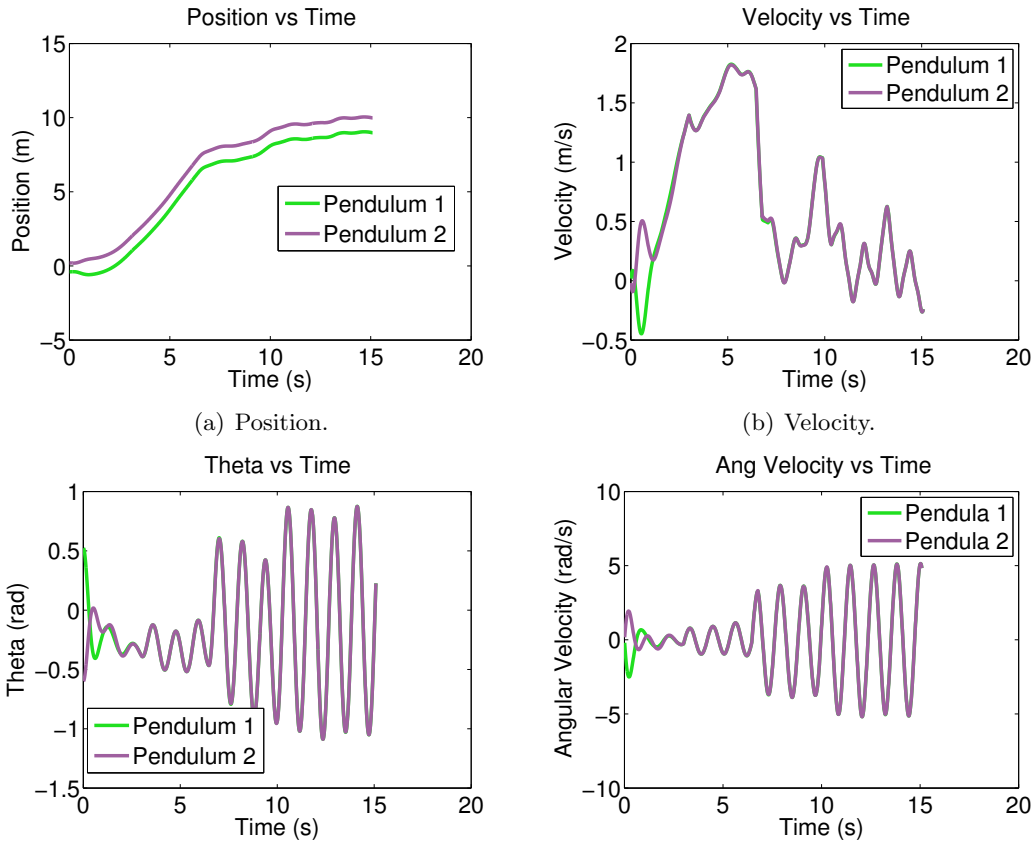


(b) Human 2 input vs Control 2 input.



(c) Squared Norm of Error from Constraint Set

Figure 7: Plots showing human operator and control input during task to drive to position $10m$. Note in (c) that the state reaches the constraint set within $2s$.



(a) Position.

(b) Velocity.

(c) Angle.

(d) Angular Velocity.

Figure 8: Plots show state versus time during the task. Note the desired inter-cart distance, identical velcity, identical angles, and identical angular velocity are acheived as well as the final cart positions.

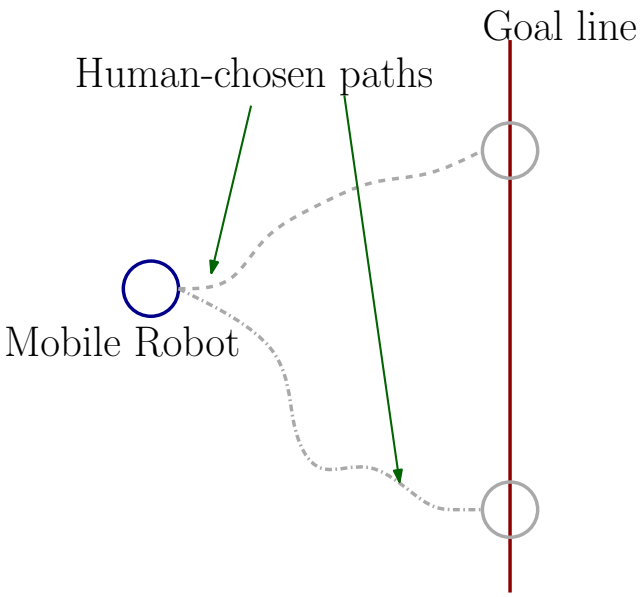


Figure 9: An example of the shared control of a mobile robot navigation task. The automated controller drives the robot to the goal line, while the user can guide the robot to a location on the goal line.

4.5 Example: Human Guidance of a Mobile Robot Navigation Task

Here, we apply the control algorithm to the shared control task of navigating a mobile robot. The low-level task is simply to get to a goal line and stay on it while the human is given the high-level task of chosing a suitable location on the goal line. This is illustated in Figure 9. This shared control task is implemented on a hardware platform described in this section.

4.5.0.1 Khepera Dynamics

The Khepera mobile robot is modeled with nonlinear unicycle dynamics

$$\begin{aligned} \dot{x_1} &= v\cos x_3 \\ \dot{x_2} &= v\sin x_3 \\ \dot{x_3} &= \omega \end{aligned} \tag{26}$$

where x_1, x_2 are the planar cartesian coordinates of the robot on a flat surface and x_3 is the orientation of the robot as shown in Figure 19. The inputs to the robot are a velocity command, v, and a rotational velocity, ω.

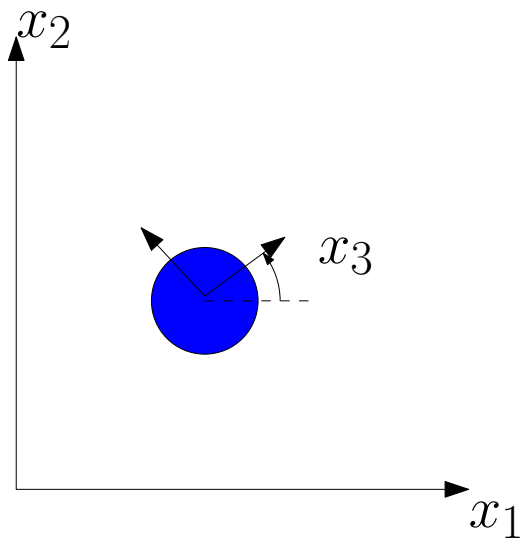


Figure 10: Mobile Robot Diagram

However, to apply the controller in the form presented, we utilize a discrete linear system. Using the near-identity diffeomorphism as developed in [67], a linear system model can be formulated for the mobile robot where

$$\begin{aligned} \tilde{x}_1 &= x_1 + \delta\cos(x_3) \\ \tilde{x}_2 &= x_2 + \delta\sin(x_3) \\ x &= [\tilde{x}_1\ \tilde{x}_2]^T. \end{aligned}$$

for some small $\delta \in \mathbb{R}$. Hence, we can rewrite the system as $\dot{x} = \tilde{A}x + \tilde{B}u$, where

$$\tilde{A} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},\ \tilde{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

and

$$u = R(x_3)\begin{bmatrix} v \\ \delta\omega \end{bmatrix}, \tag{27}$$

where $R(x_3)$ denotes the planar rotation matrix for the angle x_3. We can discretize this system to obtain the system $x_{k+1} = Ax_k + Bu_k$. In addition, the automated task, to reach to the goal line, is modeled as the linear constraint, $\mathbb{X}_f = \{x \mid Cx = b\}$. For example, a goal line $x_1 = 0$ is given by $C = [1\ 0]$ and $b = 0$.

4.5.1 Hardware

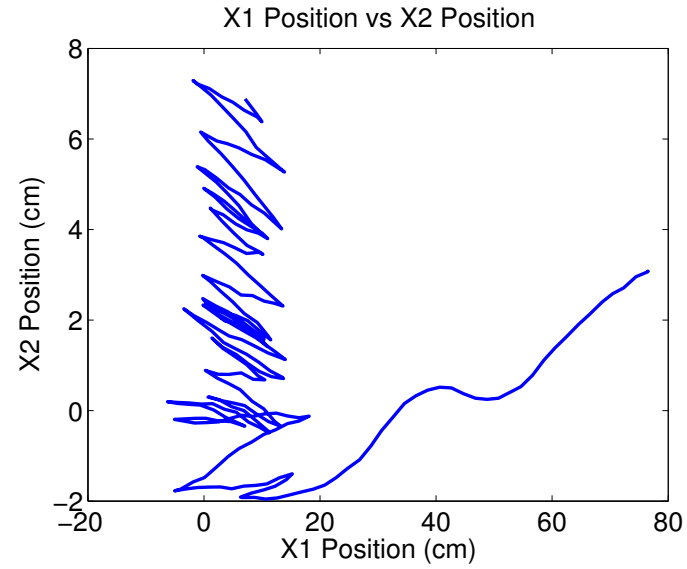
We utilize the Pancakes software architecture [55] for this hardware implemention in Java with a version running on both a laptop and the Khepera Mobile robot in the Georgia Robotics and Intelligent Systems (GRITS) Laboratory. A serial analog joystick is used in conjuction with the laptop as the human interface to the system. A Vicon motion capture system is used to provide robot localization.

4.5.2 Results

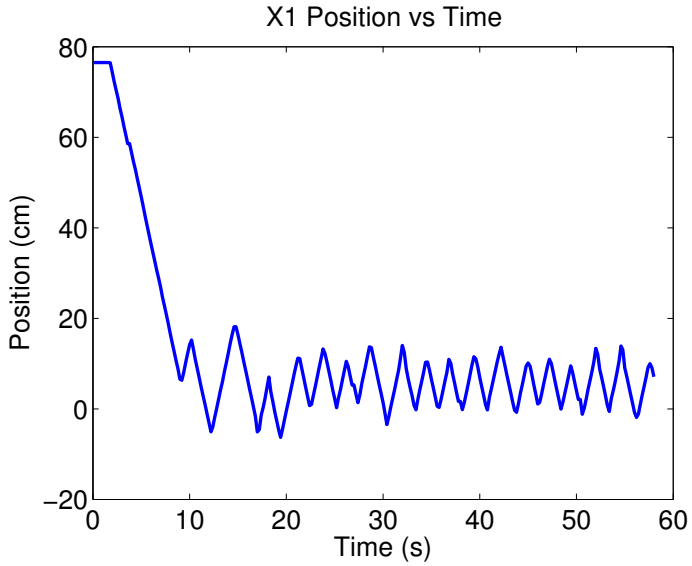
In order to verify that the control law, system identification, and variable horizon algorithm are functioning as intended, simulated joystick commands were fed into the system as $0.1 \sin t$ for the velocity command and $0.3 \sin 2t$ for the rotational velocity command. In Figure 11(a), the robot starts at a position away from the goal line and drives towards the goal line ($x_1 = 0$ for this implementation). Then once it reaches the goal line, it drives up and down the $x_1 = 0$ line. Note the robot can stray slightly from the line because of the nature of our model-predictive controller having a short horizon, i.e., the robot can leave the constraint set at times. In Figure 11(b), the robot oscillates around the line as time progresses. Therefore, the control law is functioning as intended. It should be noted that the receding horizon formulation provides a secondary function in that it aids in dealing with errors in localization or discrepancies in the dynamic model and true robot dynamics by recalculating the control at every time instant using new state information.

In Figures 12(a) and 12(b), the actual human commands are shown in solid red and at each instant the predicted sequence of future human input is plotted in dashed blue. As shown, in the first $20s$, the linear extrapolation is applied to predict future human commands. Then, once the system identification matrices are filled, system identification is used to make the predictions. Here, the system identification is shown tracking the simulated sine wave commands.

In Figure 13, the time horizon being used for the model predictive control is increasing to the maximum allowed horizon of 8s. Therefore, the variable horizon function is functioning as intended. The horizon should be expected to increase since the system identification



(a) Cartesian Position



(b) x_1 Position vs Time

Figure 11: Plots showing mobile robot position for sinusoidal input. The robot is driven to the $x_1 = 0$ line using simulated sinusoidal human input and the system identification/variable horizon version of the proposed controller.

is tracking the human commands accurately (i.e., we can "trust" our human input predictions). This 8s limit was chosen experimentally because the algorithm computationally slows down when processing horizons larger than this limit.

Then, random noise was fed into the system instead of joystick commands. As expected the system identification did not track these commands as well as it had with the sinusoidal inputs as shown in Figures 14(a) and 14(b). Hence, in Figure 15, we see the time horizon decrease. The robot also reached the goal line with no discernible change in performance from the sinusoidal inputs.

To show the effectiveness of the control framework, an actual human operator was tasked with driving the robot starting from an initial condition away from the goal line and instructed to drive the robot to an "X" on the floor, not on the shortest path from robot to goal line. This "X" is meant to represent a high-level cognitive decision made by the human operator during the task to drive to this point (i.e. the operator can see this point on the line but the robot cannot). The fixed horizon and initial horizon for the variable horizon case was chosen large enough such that the control sequence is feasible given the limits on mobile robot velocity. The operator performed the task once with no control law (i.e. manually), so human commands were sent directly to the robot. Then, the user performed the task with the presented control with linear extrapolation prediction and fixed horizon .Finally, the user performed the task with the presented controller with system identification prediction and variable horizon. All three trials were deemed successful in that the robot reached the "X" at some point during the run.

The plots in Figure 16 show the robot indeed being driven to the goal line when using the presented controller with system identification prediction and variable horizon for a human operator. For this same controller, the plots in Figure 17 show how the applied control tracked the human commands and Figure 18 shows how the time horizon changed during the task. When the system identification did not perform as well the horizon was decreased to the minimum allowable time horizon, 3s (as at the 20s mark), for which horizons lower than this generally "feel" like the human commands are being discarded. In Figure 18, we see that the horizon decreases when the human input is changing with high frequency and

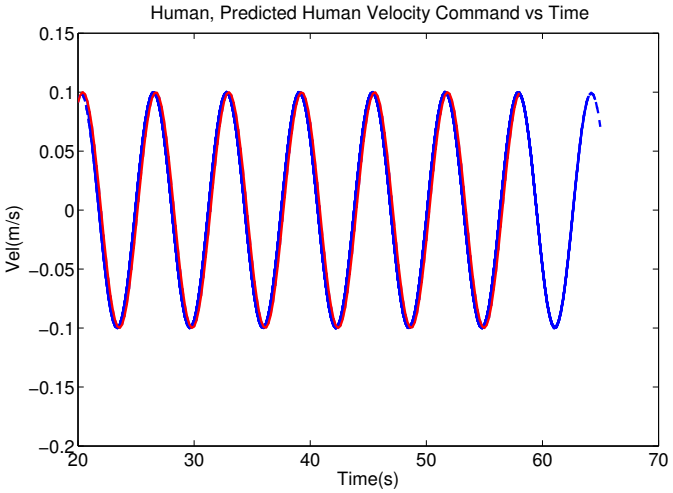
the horizon increases after $40s$ when the user input steadies.

For both runs involving the controller with fixed horizon/linear extrapolation and the controller with variable horizon/system identification, the user reported that the task of getting to the goal line was easier but understanding what the robot was doing was much more difficult.

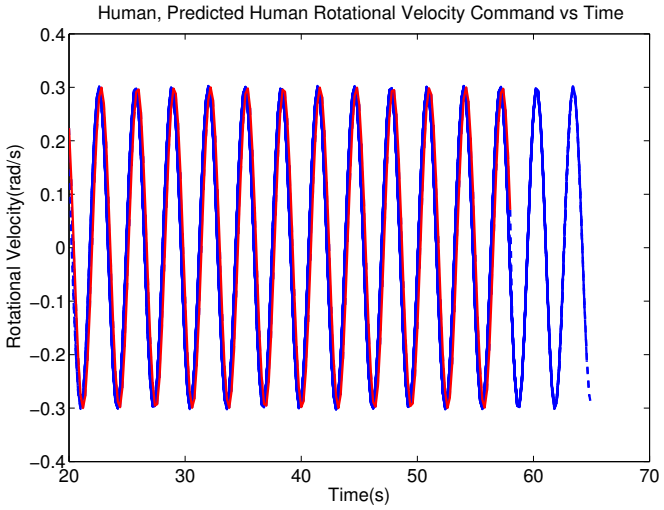
4.6 Conclusions

In summary, we have presented methods for human input prediction as well as a method to vary the horizon to facilitate human input prediction. A closed-form control law was also developed for linear systems with linear low-level state constraints. We have applied the developed control to two different examples of tasks that require both human and robot control input to complete the task. The pendulum synchronization task and mobile robot navigation showed that human operators are afforded the ability to complete high-level cognitive tasks. In the mobile robot task, we demonstrate that operators can accomplish the task for different methods of human input prediction and control horizon variation.

Both of these examples succeeded in showing that human operators were able to carry out high-level tasks while the low-level task was being completed. However, a human operator study is required to demonstrate that the ability to carry out high level tasks within the presented control framework is feasible over a larger population of human operators. In the next chapter, we present such a study and in addition to task completion, we also study task performance and operator workload to evaluate how human operators respond to this control framework.



(a) Sinusoidal Human Velocity Command



(b) Sinusoidal Human Rotational Velocity Command

Figure 12: Plots showing human and predicted human commands for simluated sinusoidal inputs. After $20s$, the system identification predictions begin and the plots show that they track the simulated sinusoidal human input.

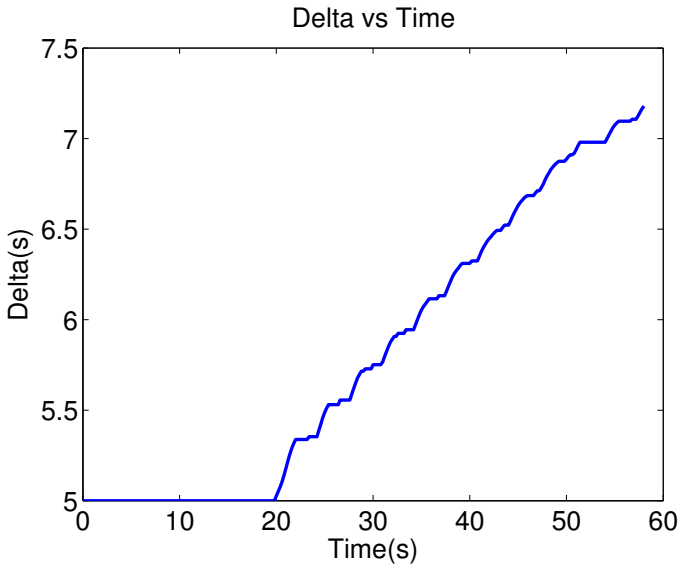
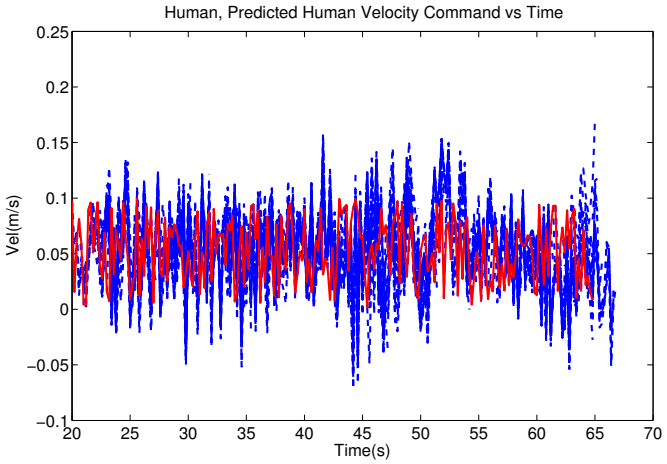
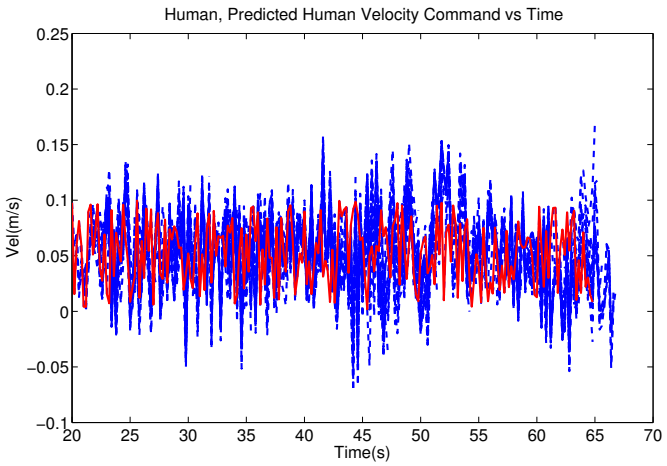


Figure 13: Plot showing the time horizon for sinusoidal input. The system identification based prediction of human input is accurate so the time horizon is increased as expected.



(a) Random Noise Human Velocity Command



(b) Random Noise Human Rotational Velocity Command

Figure 14: Plots show human and predicted human commands for random noise inputs. After $20s$, the system identification predictions begin and the plots show that they do not track the simulated random human input.

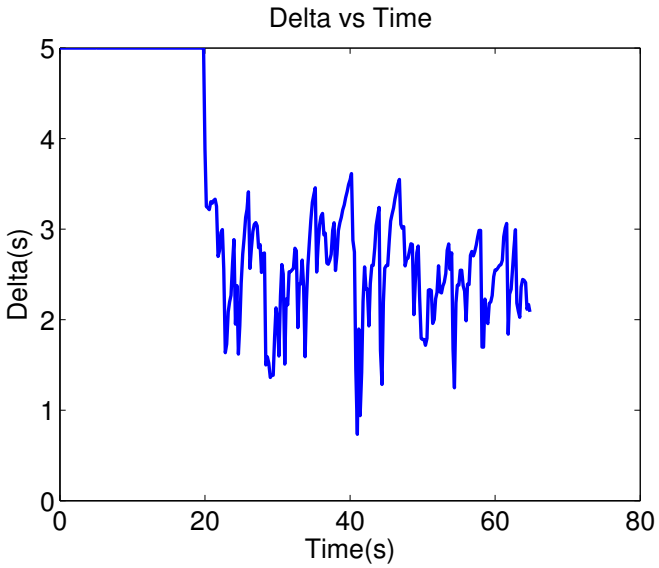
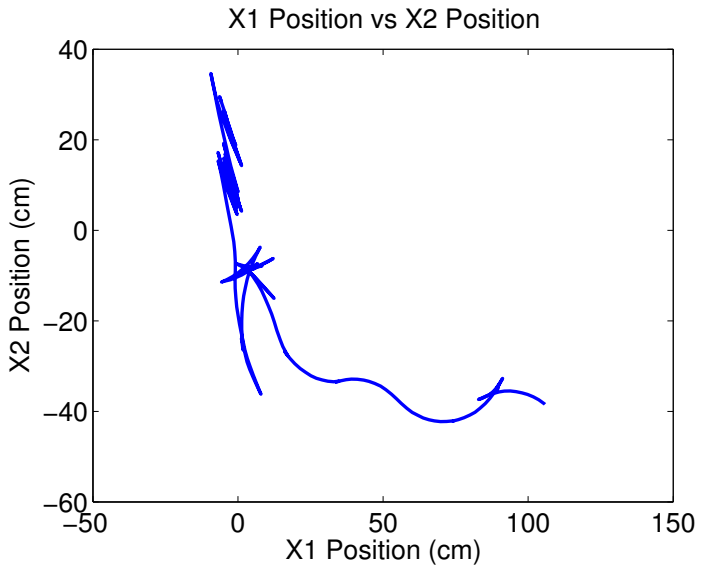
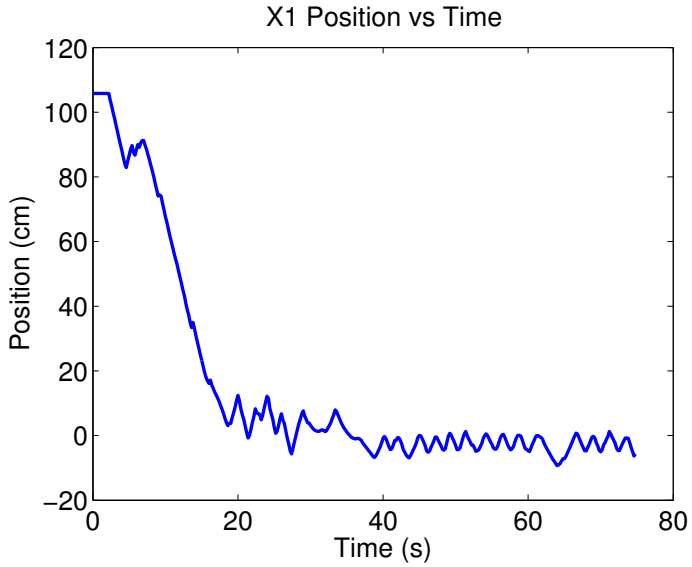


Figure 15: Plots show the time horizon with random noise input. The system identification based prediction of human input is not accurate so the time horizon is decreased as expected.

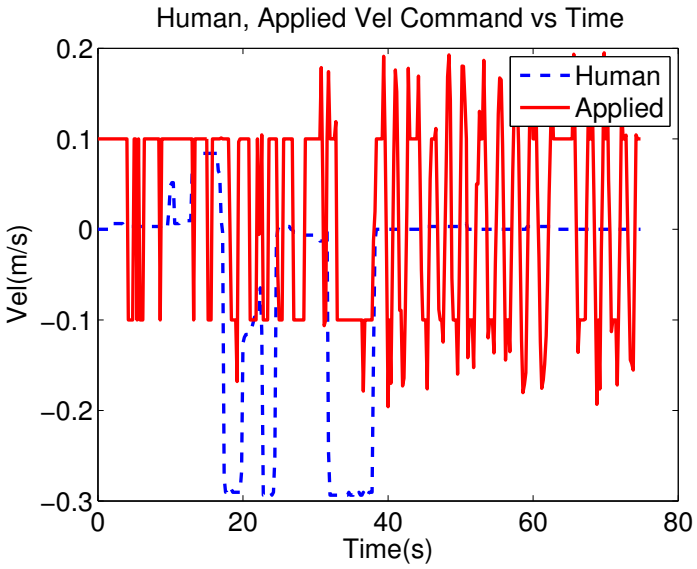


(a) Cartesian Position

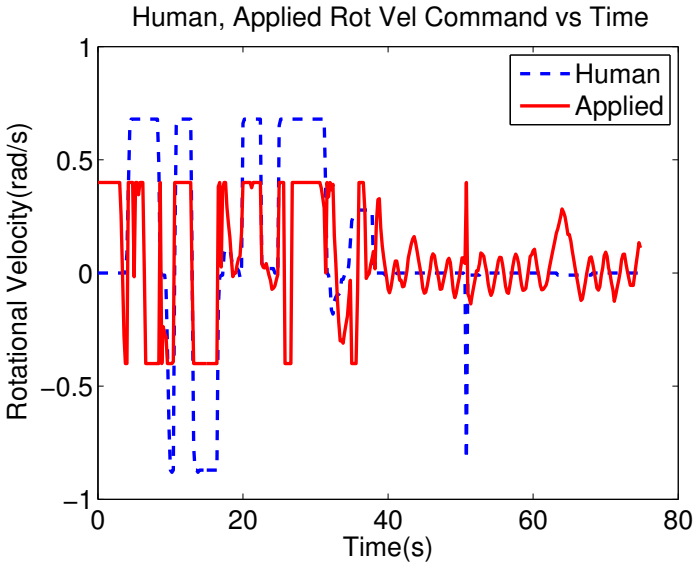


(b) X1 Position vs Time

Figure 16: Plots show mobile robot position for user input. The robot is driven to the $x_1 = 0$ line using human input and the system identification/variable horizon version of the proposed controller.



(a) Human Velocity Command



(b) Human Rotational Velocity Command

Figure 17: The actual control inputs to the system are plotted against the commanded human inputs for the task with a controller with system identification and variable horizon.

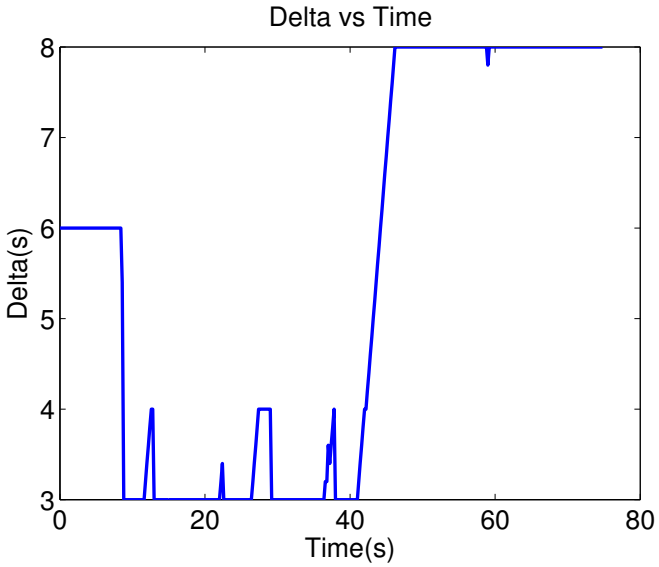


Figure 18: Plot shows changing time horizon for user input for the task with the controller with system identification and variable horizon. The horizon is decreased as the human input changes frequently and then increased as the human input becomes steady after $40s$.

CHAPTER V

HUMAN OPERATOR STUDY

The purpose of the human operator studies is multi-faceted in that we not only want to gauge the effectiveness of the lower-level control in an experimental setting, but, more importantly, we investigate whether or not human operators are afforded the freedom required to accomplish high-level tasks. In addition, we would like to measure overall task performance of different versions of the controller versus manual control as well as measure any operator workload differences. The versions of the controller are Zero-order Hold with Fixed Horizon (ZOH), Least Squares System Identification with Fixed Horizon (FSID), Least Squares System Identification with Variable Horizon (VSID), and Manual Control (Manual). This allows us to compare zero-order hold versus system identification for fixed horizons and fixed versus variable horizons for system identification prediction.

Aspects of the task pertaining to the human interface mechanism or situational awareness are not addressed here and are out of the scope of this thesis. The operators are in the same room as the work environment and are, therefore, assumed to have sufficient knowledge of the work environment.

The experimental scenario is inspired by a search and rescue operation navigation task where three points of interest have been identified (where potential victims may be) before the task begins. The automatic controller will drive the robots to any one of these points, but the human operator is to guide the robot to the points in the order deemed appropriate by the operator. The human also has the power to influence the path taken by the robot to each of these points. During the task, the human identifies a possible new area of interest (where there could be additional victims) on the way between two of the predefined points. The human must then actively alter the robot path to visit this point without altering the low-level task. This scenario requires sliding autonomy in that human involvment ranges from a little to a lot of operator interaction with the automatic controller.

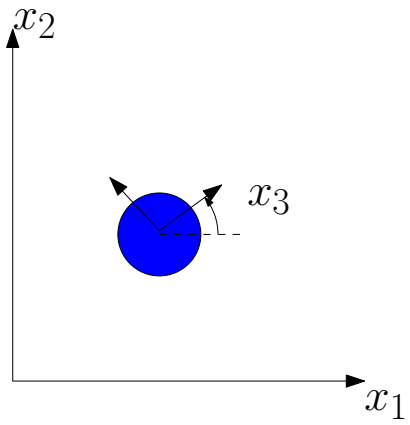


Figure 19: Mobile Robot Diagram

Based on pilot studies, it is hypothesized

Hypothesis: The human-in-the-loop controller with the simplest human input prediction technique (namely the ZOH controller) will result in both low and high level task completion with the shortest completion times as well as the lowest operator workload.

The following section details the experimental platform and this is followed by details and results of the experiments.

5.1 Experimental Platform

The experiments in this thesis were run using a Khepera mobile robot wirelessly receiving commands from a Ubuntu PC running Robotic Operating System ([73]). Localization information is supplied by a VICON motion capture system giving planar position and orientation.

A low-level unicycle controller (for example see [1]) that takes in planar change-in-position commands and outputs velocity and angular velocity commands (which are the inputs to the robot) allows us to use a discrete linear control system model to command the Khepera mobile robot,

$$z_{k+1} = z_k + u_k \tag{28}$$

where $z = (x_1, x_2)$ are the planar Cartesian coordinates of the robot and x_3 is the orientation of the robot as shown in Figure 19. We now have discrete system dynamics for $\mathcal{P}_{N_k}$, however, a description of the goal set $\mathbb{X}_f$ is still needed. The low-level task in this chapter will be for

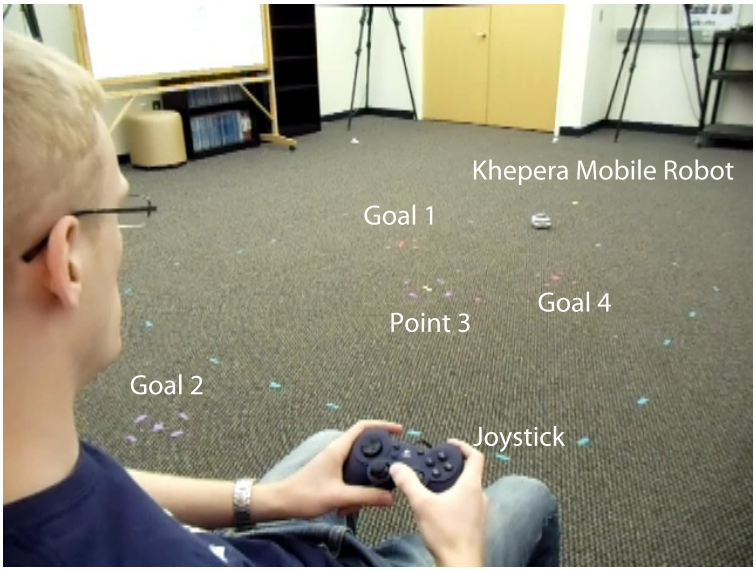


Figure 20: Navigation Task Evironment with Khepera Mobile Robot. The joystick shown is used as the operator interface.

the robot to reach one of three goal points, each modeled as a linear constraint, $Cx_k = b_i$ for $i = 1, 2, 3$ where $C = I_2$ (the identity matrix in $\mathbb{R}^{2\times 2}$) and b_i is the planar goal location. Hence, for $\mathbb{X}_i = \{x \mid Cx = b_i\}$ with $i = 1, 2, 3$, the constraint set on the optimal control problem is

$$\mathbb{X}_f = \mathbb{X}_1 \cup \mathbb{X}_2 \cup \mathbb{X}_3. \tag{29}$$

The choice of this goal set will be further discussed in the next section. Once in the goal set, the operators are given full manual control to stay at this goal point or move on to another goal point.

Human commands are supplied by way of a video game-like gamepad with joysticks as shown in Figure 20. The joystick allows the operator to issue change-in-position commands in the global frame without regard to the orientation of the robot. This simplifies the control as well as allowing for a direct comparison of manual control versus mixed-intiative control, since both are acting on the same linear system. In Figure 20, images of the laboratory set-up and the joystick are shown.

5.2 Human Study Results

We apply the control algorithm presented in this thesis experimentally to a laboratory-equivalent of the aforementioned search and rescue scenario. As shown in Figure 21, the predefined goal points are labeled as Goal 1, Goal 2, and Goal 4, whereas the mid-task goal-point is labeled Point 3. The low-level task is to ensure that the robot does indeed reach one of the predefined goal points (Goals 1, 2, and 4). The high-level task consists of chosing which order the goals are visited as well as visiting the goal point not predefined as a goal (Point 3) in between Goal 2 and Goal 4. The operator is situated in the same room as the task environment and has full view of work environment as shown in Figure 20. The laboratory environment does not simulate the same robot mobility, situational awareness, and workload challenges found in actual search and rescue operations, but it successfully serves the purpose of requiring the human-robot team to operate at different levels of autonomy.

The participants were instructed to visit the goals in the order: Goal 1, Goal 2, Point 3, Goal 4. They were instructed that they must drive the robot into the circles (5cm radius) surrounding each goal point and to complete the task as quickly as they can. The following details the 4x1 within-subjects design experiment where each subject used each of the four controllers in a counter-balanced order. Here, high-level task completion is measured by whether or not the operators are able to visit each goal point in the order required by the task. Overall task performance is measured by the time required to reach the final goal point while workload is measured by the NASA TLX workload scale.

5.2.1 Participants

10 operators were recruited for the experiment from the Georgia Institute of Technology community. None of the operators have had previous experience with mobile robot control. Particpants were between the ages 20-30 with 3 female participants and 7 male participants.

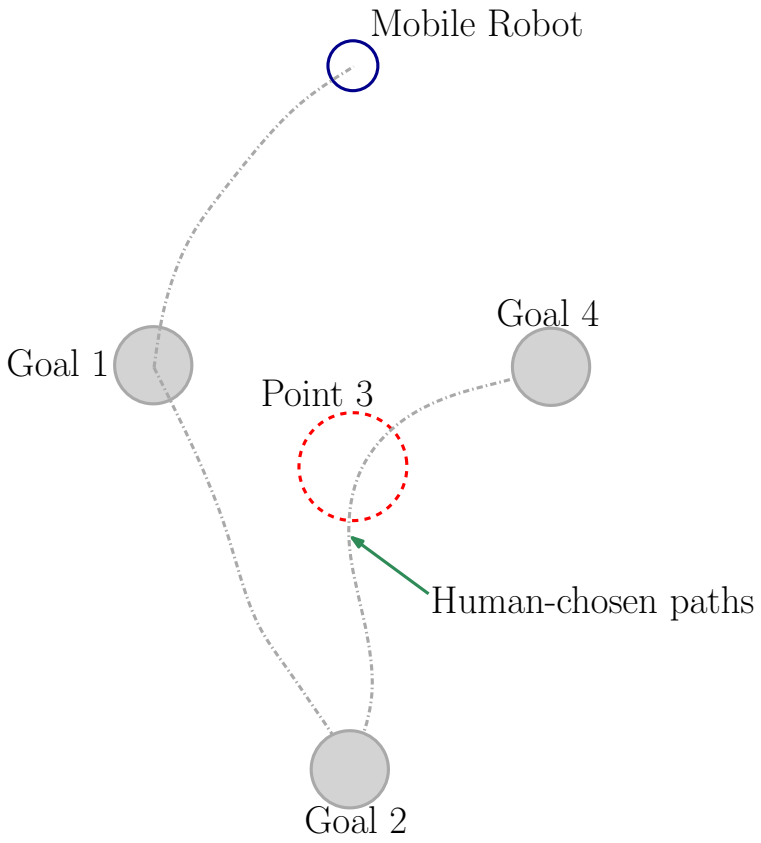


Figure 21: An example of the shared control of a mobile robot navigation task. The automated controller drives the robot to the goal points (Goal 1, Goal 2, Goal 3), while the user specifies the order the goals are visited and influences the robot path so as to visit Point 3 on the way to the final goal point.

5.2.2 Navigation Task

For each trial, the participant was asked to drive the robot to Goal 1, then Goal 2, then to pass through Point 3, on the way to Goal 4 as seen in Figure 21. The ordering of the goals was set before the trials by the study administrator and were the same for every participant. The controller is programmed with Goal 1, Goal 2, and Goal 4 given a priori however, Point 3 is not part of the low-level task. and the controller is not given the order in which the goals must be visited. In this way, human control naturally shifts from a supervisory type of control towards that of a more manual control to visit Point 3.

5.2.3 Procedure

Participants read standard written instructions on the task and then physically shown the task environment with verbal instructions on how to complete the navigation task. The participants were first allowed to practice the task using only manual control for a maximum of three times. Then, the participants were given a training session with each of the controllers before performing the task with recorded data. Each training session consisted of a maximum of three attempts at the given task. Each recorded run was followed by a NASA TLX workload survey. The order of the four controllers were counter-balanced to account for any ordering effects. After the four trials, the particpants were given an exit survey comparing the four controllers.

5.2.4 Results

The robot is defined as successfully reaching a goal point if its position was recorded as being within a 10 cm ball as shown by the red and purple circles in Figure 20. In all 40 trials, the robot successfully reached all three of the goal points, demonstrating low-level task completion. More importantly, the operators in every trial were able to guide the robot in the specified goal order given by the test administrator as well as visiting Point 3. As a result, it can be concluded that operators and the robot able to complete both low-level and high-level functions with the manual control as well as complete high-level functions while the three different versions of the mixed-initiative controller ensured low-level task

completion. The repeated-measures approach to these experiments isolates the effects of the different controllers without the effects of operator-to-operator variability.

In order to evaluate any performance and workload differences between manual control and the three versions of the proposed controller, we analyzed the total task completion time (i.e. time to reach Goal 4) and NASA TLX workload data. Figure 43 shows the task completion times for each participant with each of the four controllers. The ZOH controller resulted in the fastest completion times across *all* participants. The mean completion times for each controller are as follows: 46.5 sec for ZOH, 60.8 sec for Manual, 76.3 sec for FSID, and 77.1 sec for VSID. The hypothesis in the chapter was that the ZOH controller would out-perform the other controllers, so a repeated-measures pairwise t-test was performed for the ZOH controller against the other three controllers.

The resulting t-value and p-values for each pairwise test are ZOH versus Manual ($t(10)= 3.071$, $p=0.0133$), ZOH versus FSID ($t(10)= 5.798$, $p=0.0002$), and ZOH versus VSID ($t(10)= 5.056$, $p=0.0007$). Using the convention that p-values less than 0.05 are deemed statistically significant, we can see that the ZOH controller statistically significantly out-performs the other controllers.

Workload is measured using the NASA TLX survey and our analysis is carried out over the total raw NASA TLX scores. The raw total scores for all participants are plotted in Figure 23 where we see a trend that the workload scores for the ZOH controller tend to be less compared to the other controllers. The mean raw scores for each controller are as follows: 30.7 for ZOH, 51.4 for Manual, 51.4 for FSID, and 53.6 for VSID. A repeated measures pairwise t-test was performed for the ZOH controller against the other three controllers to test the hypothesis. The resulting t-value and p-values for each pairwise test are ZOH versus Manual ($t(10)= 3.898$, $p=0.0036$), ZOH versus FSID ($t(10)= 2.938$, $p=0.0165$), and ZOH versus VSID ($t(10)= 3.318$, $p=0.0089$).

Hence, the ZOH controller was shown to statistically significantly have a lower operator workload than manual control and the other prediction methods. The results of these human studies have shown that not only will the mixed-initiative control scheme guarantee low-level task completion, but we have shown that human operators have the freedom to

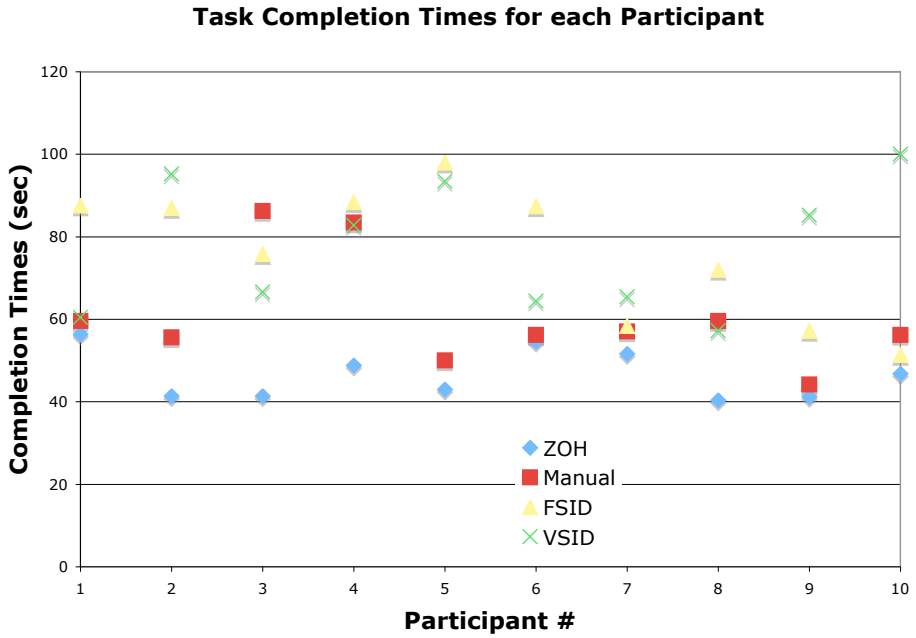


Figure 22: Task Completion times for all participants show that the ZOH-based controller results in the lowest task completion times across all participants.

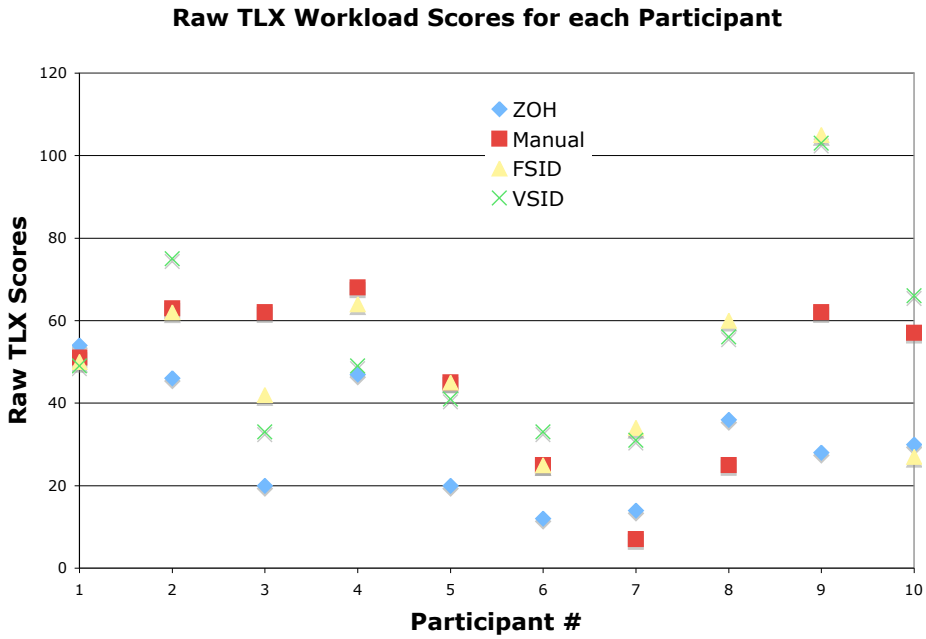


Figure 23: Raw Total NASA TLX Workload survey scores for all participants. The ZOH controller workload scores tend to be lower than the other controllers. Statistical analysis confirms this statistically significant advantage.

accomplish high-level tasks with benefits to performance and operator workload in this particular experiment.

Finally, the final survey asked the operators to choose which of the controllers they would prefer to use again, which was the most frustrating to use, and which controller did they trust the most. The results of the survey are shown in Figure 24. The ZOH controller was both the most preferred and trusted controller while the VSID controller was the most frustrating. ZOH, along with FSID, was the least frustrating to use. It can also be seen that no one preferred or trusted Manual. In summary, the hypothesis in the chapter was supported by these results.

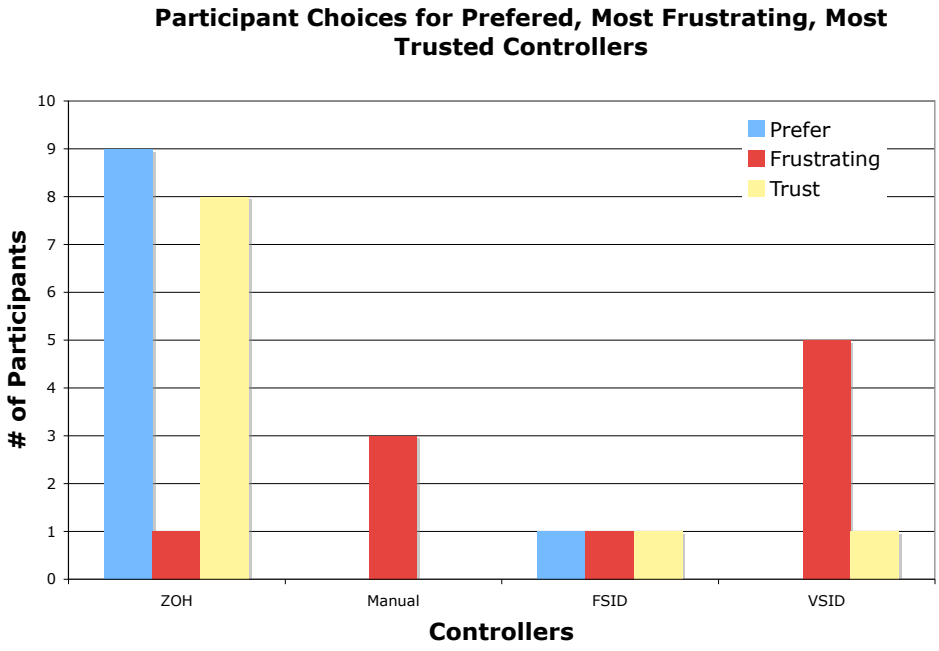


Figure 24: Final Survey results showing number of participants indicating the controllers they preferred, thought were the most frustrating to use, and thought they could trust the most. The ZOH controller was the most trusted and preferred, while the VSID was the most frustrating.

5.2.5 Discussion

Here, we discuss some qualitative aspects of the study. While all participants were able to complete the manual task, the test administrator observed that driving the robot to exactly within the circles was not easy and participants would often have to make major corrections when close to the goal. On the other hand, the mixed-initiative controllers allowed the user to relinquish control to the controller when the robot was close to the goal point and the controller would guide the robot smoothly to the goal. We see this result in the form of the time completion advantage the ZOH control had over Manual control as well as Manual control having the second highest frustration level in the final survey.

The other major outcome was the advantage of the ZOH prediction method over the SID method with both fixed and variable horizons despite the simplicity of ZOH. The authors hypothesize that there are two factors at play here. One is that users tended to use the joystick "stop-to-stop" in that the users push the joystick to the physical limits when driving. A ZOH approximation of this type of joystick input is a reasonable method and responds well to high frequency changes in human input whereas the SID method stores and makes predictions based on 2 seconds worth of human input data, proving inadequate.

The other factor is the notion that the human operator is also "running system identification" on how the robot is reacting to human commands. In other words, human operators give commands based on how they think the robot will alter their commands. With the ZOH method, this alteration to the human commands is fairly intuitive, while the SID methods obscures what this alteration will be. This lack of intuition, coupled with the potential for the variable horizon method to instantaneously move away from the goal (though it does not affect asymptotic convergence), leads to the high frustration score with the VSID controller seen in the final survey. While the system identification approach and the variable horizon scheme theoretically improve human prediction and therefore improve the model predictive control, in practice with human operators, the simpler fixed horizon/linear extrapolation scheme is more effective.

This leads to the question: are sophisticated prediction schemes for human input appropriate when humans are in the control loop? Future human studies would be required

to investigate this aspect and how it applies to mixed-initiative control. While varying the horizon to improve prediction proved ineffective, varying the horizon in response to operator intent is still a viable option and a future direction of work. Since the horizon length effects how much the human control is "weighted", a scheme based on how well the operator is performing the low-level task could be used to vary the horizon (i.e. expand the horizon when performing well to give more freedom and contract the horizon when performing poorly to further assist the human operator).

5.3 Conclusions

The developed control framework was applied to a USAR inspired mobile robot navigation task in which the controller displayed naturally sliding levels of autonomy. While experimental results confirmed theoretical low-level task completion guarantees, the experimental results with human operators also showed that the control scheme allows for high-level task completion with benefits to performance and operator workload for a specific search-and-rescue-motivated mobile robot navigation task. These benefits were shown to be statistically significant with the Zero-Order Hold with Fixed Horizon version of the controller.

So far in this thesis, we have presented a model-predictive control with theoretical guarantees of low-level task completion and experimental results showing high-level task completion for an effective human input prediction method and closed-form control law. Using these results as a foundation, we apply this control framework to a robot being developed specifically for human-in-the-loop USAR tasks. The USAR scenario is the cooperative human-robot control of a quadruped rescue robot meant to assist search and rescue missions in disaster sites where sending humans would be too dangerous. This work is presented in the next chapter.

CHAPTER VI

SHARED CONTROL OF A QUADRUPED RESCUE ROBOT

As discussed earlier in this thesis, USAR robotics is an important domain for HRI research, as it requires the high-level skills only human operators are currently capable of providing while trying to take advantage of the safety and efficiency of automated tasks that state-of-the-art robotics affords. In the USAR scenario, humans are kept in the control loop, while being safely out of harm's way. In response, we apply the control framework presented in this thesis to the shared control of a USAR robot. Therefore, we deploy the developed control framework to compose human-controlled front foot-placement commands with an automatic controller that prevents unstable foot-placement for a quadruped rescue robot.

The Compact Rescue Robot (CRR) is a pneumatically-actuated tele-operated rescue robot being developed by the Intelligent Machine Dynamics Laboratory (IMDL) at Georgia Tech (Figure 25). The quadruped robot is a testbed of the Center for Compact and Efficient Fluid Power, which seeks to demonstrate the benefits of fluid power to engineering technologies. The CRR is controlled using an interface that allows the two front legs to be manipulated by a human operating two corresponding Phantom$^{\text{TM}}$ haptic joysticks.

The operator can utilize the haptic feedback to aid in foot-placement in uneven terrain commonly found in disaster sites. However, in the previous configuration, the user received little to no feedback on whether or not these foot-placements would lead to an unstable leg configuration that would cause the robot to fall over. This lack of situational awareness for stability warrants an online controller that tries to both preserve the intent of the human operator and satisfy the stability constraints of the robot, i.e. by sending motor commands that minimize errors from the human command while still satisfying a constraint. In other words, human operators are effective in navigating the robot in rough terrain, but are poor at preventing the robot from falling over. An effective shared control framework would combine the strengths of both operator and autonomous stability control.

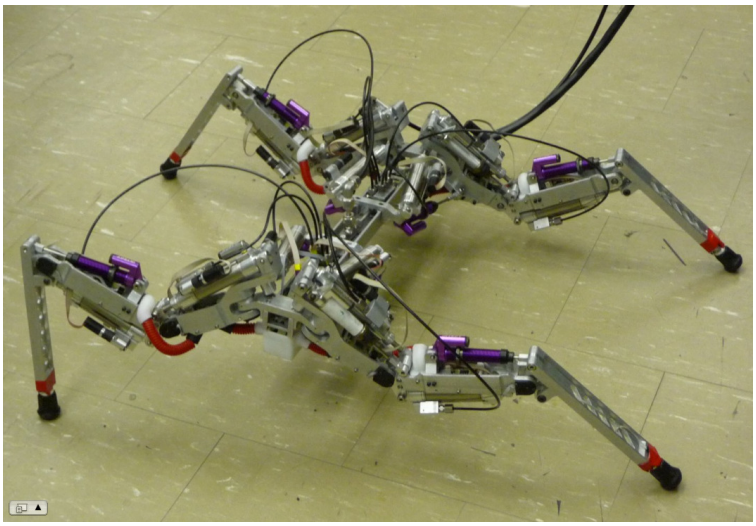


Figure 25: Quadruped Rescue Robot

Specifically, we utilize the presented control framework to implement a controller that applies user foot-placement commands as closely to what the human intended as possible while still maintaining static stability for the next leg placement as well as the rest of the proposed robot gait. The front leg controller is updated with the latest leg positions and human input commands and is composed with human commands so that the stability constraint is not violated at foot touch-down, i.e. the front foot is never placed on the floor such that the center of mass of the robot is outside the polygon created by that leg and the hind legs. This allows for the pick-up and placement of the other front leg. This is demonstrated on a simulation testbed of a quadruped rescue robot.

6.1 Problem Formulation

In the following section, we model how the robot's feet must be placed in order to avoid losing static stability and how the proposed robot gait and control framework is constructed to accomodate human commands and automatic movements. A planar robot model is proposed where all leg and center of mass positions in $\mathbb{R}^3$ are projected onto the ground plane ($\mathbb{R}^2$) as in Figure 26. This is a typical formulation in the quadrupedal robot stability literature (e.g. [28], [29], and [20]). All positions are given in reference to a global coordinate frame.

As discussed in [28], in order to pick up and place a leg, the center of mass must lie in

the triangle formed by the other three legs (commonly referred to as the stability polygon). If this is the case, then, for example, the left leg must be picked up and placed in a position such that right leg can be picked up and placed. This requires that the first leg be placed in such a way that it forms a triange with the two remaining legs that contains the center of mass. The stability polygons for both left and right front leg are shown respectively as the dashed and dash-dotted triangles in Figure 26.

In this chapter, we represent the set of leg placements that lead to the center of mass being in these stability polygons as the stability cones for the front and back legs as shown in Figure 26. We will first address robot stability, then will discuss the gait proposed in Section 6.1.2 and the subsequent proposed front leg control.

6.1.1 Stability Cone

Referring to Figure 26, in order to guarantee a stable right leg lift and place, the front left leg should be placed in the half plane defined by the line passing through the center of mass and back right leg. When the left leg is placed in this half-plane, then the left leg necessarily creates a stability polygon with the back legs such that the center of mass lies within this polygon. Hence, the robot is statically stable when the right leg is lifted. Similarly, the front right leg should be placed in the half plane defined by the line passing through the center of mass and the back left leg. Together, the intersection of these two half-planes define the *stability cone*. If the front left leg is not placed in the stability cone, then the subsequent front right leg placement will result in static instability and the robot will fall.

An analytic expression is derived for this cone using lines orthogonal to the line connecting the center of mass to the back legs. The following notation is used: f_l is the position of the front left leg, f_r is the position of the front right leg, b_r is the position of the back right leg, b_l is the position of the back left leg, and c_m is the position of the center of mass. All positions are in $\mathbb{R}^2$. R_θ is the planar rotation matrix evaluated at angle θ. The half-plane defining where the front left leg can be placed is given by

$$R_{-\frac{\pi}{2}}(b_r - c_m)^T(x - c_m) \geq 0$$

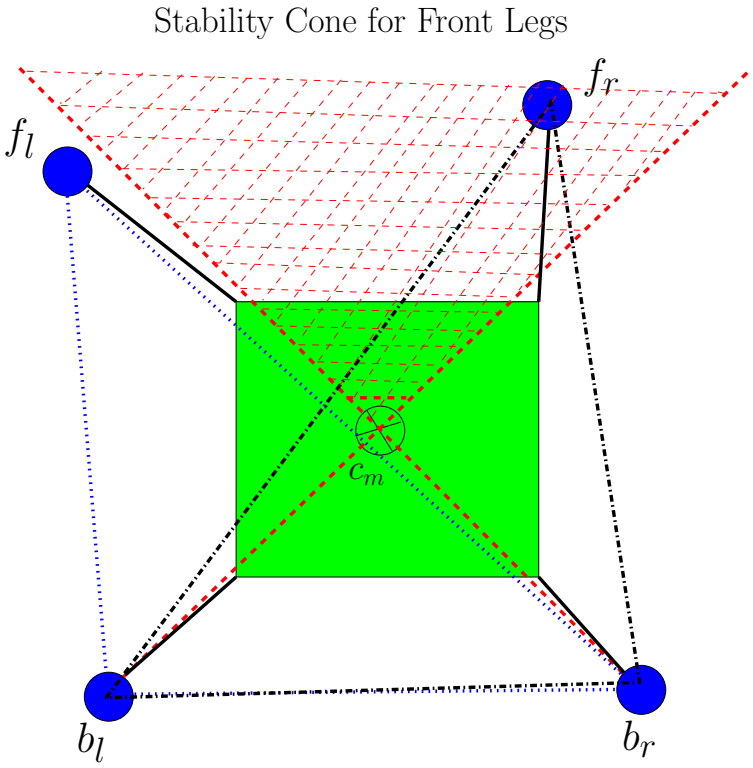


Figure 26: The Front Leg Stability Cone is shown in the dotted hatching, while the stability polygon for the right front leg is shown as the dotted triangle and the stability polygon for the left leg is shown as the dash-dotted triangle.

where x represents all front left leg positions that allow for the stable lifting of the front right leg. Similarly, the half-plane defining where the front right leg can be placed is given by

$$R_{\frac{\pi}{2}}(b_l - c_m)^T(x - c_m) \geq 0$$

where x represents all front right leg positions that lead to stable lifting of the front left leg. As such, the intersection of these two half-planes define the stability cone giving $C_f x \geq e_f$, where

$$C_f = \begin{bmatrix} R_{-\frac{\pi}{2}}(b_r - c_m)^T \\ R_{\frac{\pi}{2}}(b_l - c_m)^T \end{bmatrix}, \; e_f = \begin{bmatrix} R_{-\frac{\pi}{2}}(b_r - c_m)^T c_m \\ R_{\frac{\pi}{2}}(b_l - c_m)^T c_m \end{bmatrix}$$

for all x representing stable front leg positions. In the same vein, the stability cone for the back legs is given by $C_b x \geq e_b$ with

$$C_b = \begin{bmatrix} R_{\frac{\pi}{2}}(f_r - c_m)^T \\ R_{-\frac{\pi}{2}}(f_l - c_m)^T \end{bmatrix}, e_b = \begin{bmatrix} R_{\frac{\pi}{2}}(f_r - c_m)^T c_m \\ R_{-\frac{\pi}{2}}(f_l - c_m)^T c_m \end{bmatrix}$$

for all x representing stable back leg positions. Now that a formal definition of the stability cone is given for the robot, we can address details of the proposed gait.

6.1.2 Gait

The gait sequence will be Move 1: front left leg, Move 2: front right leg, Move 3: center of mass shift forward, Move 4: back right leg, Move 5: back left leg, and Move 6: center of mass shift backward; then, the cycle repeats. The particular gait proposed in this work allows for the placement of the front two legs in succession without a center of mass shift. This is desirable in that one set of position constraints can be used for both front leg movements, which are purely a function of the center of mass and the back legs, and not of the position of the other front leg.

Move 3 is necessary to guarantee that the back right leg can be picked up, since the front right leg is only placed in the stability cone that guarantees stability for left leg placements. This shift also serves the purpose of widening the stability cone for the back legs by moving the center of mass closer to the front two legs as well as dictating the direction the entire robot will move in.

Next, Moves 4 and 5 will place the back two legs in the back leg stability cone, and this portion of the control is fully automated so no human input is needed. The back legs are moved in the same general direction that the front two legs are moved in and are required to be in the back leg stability cone. Lastly, Move 6 moves the center of mass back towards the back legs to widen the stability cone for the front legs and guarantees that the front left leg can be moved.

It is important to note that the proposed gait is a departure from the commonly accepted gait (as in [46]), where the back leg placements start the gait. However, in this case, the human input guides not only the front legs but the general robot motion, so it is required that the front leg placements initiate the proposed gait. After formally defining the stability cone and gaits, the next section details how each of these leg movements are implemented in a control framework.

6.1.3 Dynamic System and Hybrid Control Framework

Human input to the front legs is provided by a change in position vector. As such, the following are the linear discrete dynamics for the front legs,

$$f_l^{k+1} = f_l^k + u_l^k,$$
$$f_r^{k+1} = f_r^k + u_r^k,$$

where the superscript k refers to the current discrete time and $k+1$ is the subsequent time step.

The discrete dynamics describing the back legs and center of mass are formulated differently. This is because low-level controllers achieve the desired positions via displacement commands, $u_{b_l}^k$ for the back left leg,$u_{b_r}^k$ for the back right leg, and u_c^k for the center of mass. The low-level functions are discussed in Sections 6.2 and 6.3. The back legs and center of mass evolve in discrete time as

$$b_l^{k+1} = g_l(b_l^k, u_{b_l}^k),$$
$$b_r^{k+1} = g_r(b_r^k, u_{b_r}^k),$$
$$c_m^{k+1} = g_c(c_m^k, u_c^k).$$

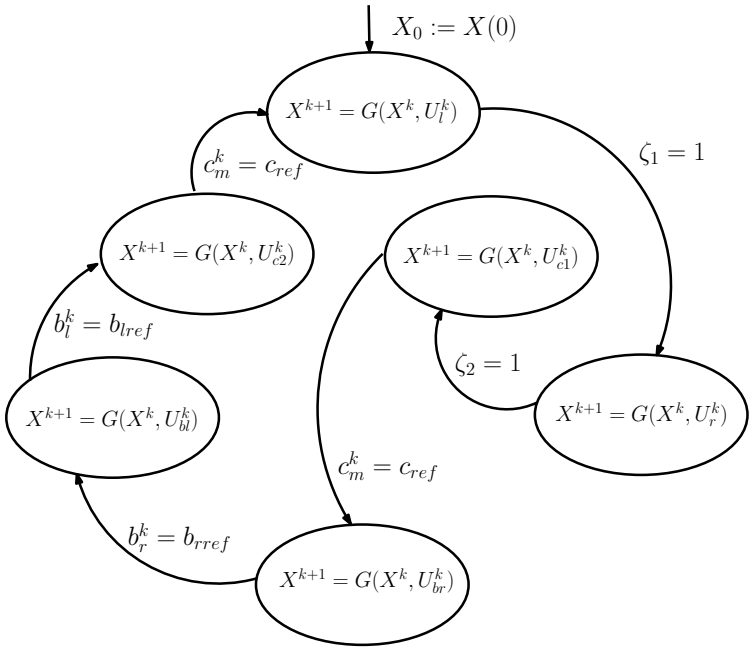


Figure 27: Quadruped Rescue Robot Hybrid Automoton

Collecting the dynamics, let $X^k = (f_l^k, f_r^k, b_l^k, b_r^k, c_m^k)$ and $U^k = (u_l^k, u_r^k, u_{b_l}^k, u_{b_r}^k, u_c^k)$ be column vectors which lead to the dynamics of the system,

$$X^{k+1} = G(X^k, U^k).$$

Let the subscript l in U_l^k denote the vector U^k where every control element is zero but u_l^k, e.g. $U_l^k = (u_l^k, 0, 0, 0, 0)$. This notation follows for the vectors U_r^k, U_{bl}^k, U_{c1}^k, and U_{c2}^k. since the system is decoupled and only one leg or center of mass is moved at a time.

Since each leg and the center of mass is controlled individually, we describe the system as a hybrid system modeled with the hybrid automaton in Figure 27. In the figure, $X(0)$ denotes the initial leg and center of mass positions. The guards for the front legs are simple flags set by the user to indicate completion of a particular leg placement. The guards for the other movements are set when the leg reaches the desired position calclulated by the automatic control.

6.2 MPC-based Shared Control: Front Legs

With the dynamic model and control structure defined, we formulate the control laws needed to both incorporate human commands for the front legs as well as guaranteeing static

stability for the rest of the gait. The notation in this chapter differs slightly from the rest of the thesis, so we review the control formulation in this chapter using the altered notation. Let $\{v^k\} = \{v^k, \ldots, v^{k+N-1}\}$ be a sequence of predicted human inputs that drive one of the front legs for N discrete time steps into the future (N is called the control horizon) and let $\{u^k\} = \{u^k, \ldots, u^{k+N-1}\}$ denote the applied control sequence for one of the front legs, where $u^k = u_l^k$ or $u^k = u_r^k$ and $x^k = f_l^k$ or $x^k = f_r^k$, depending on which leg is being placed (i.e. Move 1 or Move 2). The predicted human input is a linearly extrapolated sequence starting with u^k and with slope $u^k - u^{k-1}$ as in Chapter 3.

The following controller minimizes deviations from the predicted human input sequence while satisfing the constraint that the leg position be in the stability cone at the end of the control horizon. As such, the model predictive optimal controller solves the following optimal control problem, $\mathcal{P}_N$, at each discrete time instant k,

$\mathcal{P}_N$:

$$\min_{\{u^k\}} \sum_{i=k}^{k+N-1} (u^i - v^i)^T (u^i - v^i),$$

such that

$$\begin{aligned} x^{k+1} &= Ax^k + Bu^k, \\ x^{k+N} &\in \mathbb{X}_f = \{x \mid C_f x \geq e_f\}. \end{aligned} \tag{30}$$

where A and B are equal to the identity matrix in $\mathbb{R}^2$ for the front legs.

The control, u^k, applied to (30) is the first element in the sequence $\{u^k\}$. Due to the inequality constraint, this optimal control problem will be solved numerically at each time instant. This is where the control formulation presented in this chapter differs from the control problem solved in Chapter 4, where an analytic solution was given for an equality constraint. In Chapter 3, it is shown that this particular model predictive control will result in the state (i.e. leg position) asymptotically converging to the constraint set, i.e. the stability cone. The control ceases when the user sets a completion flag, thus setting $\zeta_1 = 1$ when the left leg control is completed, and $\zeta_2 = 1$ when the right leg control is completed.

A low-level control function is now needed to implement the commanded front leg displacement as leg joint commands that result in the desired motion on the robot leg. This

motion of the front legs is achieved by mapping the end effector position of the Phantom joysticks to local leg workspaces on the shoulder, and then mapping these positions to desired robot joint angles. With the proposed controller, the user's commands are adjusted, resulting in a sense of haptic guidance. Together with the low-level control functions, we now have a controller for U_r^k and U_l^k as seen in the hybrid automaton diagram in Figure 27.

6.3 Center of Mass and Back Leg Control

This next section addresses the automated control of the center of mass and the back legs. The first center of mass shift moves the center of mass towards the front two legs, in order to advance the robot in the general robot motion direction as dictated by the front legs, as well as guarantee stability for a back right foot placement. Hence, we define a constraint restricting the center of mass to lie in the stability polygon formed by the front two legs and the back left leg, $C_{c1}c_m \geq e_{c1}$, where

$$\begin{aligned} C_{c1} &= (R_{\pi/2}(b_l - f_r))^T (f_l - f_r)(R_{\pi/2}(b_l - f_r))^T \\ e_{c1} &= (R_{\pi/2}(b_l - f_r))^T (f_l - f_r)(R_{\pi/2}(b_l - f_r))^T f_r. \end{aligned}$$

for all c_m that satisfy the stability constraint.

This constraint determines if the center of mass is on the same side of the front-right-leg-to-back-left-leg-line as the front left leg. Let $g = (f_r + f_l)/2$ and, for some parameter $\alpha \in \mathbb{R}^+$, let $d = \alpha(g - c_m)$ define the general direction of motion of the robot which, in other words, is the vector from the center of mass to the midpoint of the line segment connecting the front two legs. In this way, the human operator not only has input on where the front legs are placed, but also on the general direction the robot will move in. Now, we solve the following quadratic program to find a control input, u_{c1}, to shift the center of mass:

$$\min_{u_{c1}^k} (u_{c1}^k - d)^T (u_{c1}^k - d),$$

such that

$$\begin{aligned} c_m^{k+1} &= g_c(c_m^k, u_c^k), \\ C_{c1}c_m^{k+1} &\geq e_{c1}. \end{aligned}$$

In reference to the hybrid architecture, the guard for the first center of mass movement is $c_{ref} = c_m^{k+1}$. This is also solved numerically as a quadratic program and the robot center of mass is driven to the resulting center of mass position using low-level control functions discussed in Section 6.4.3.

Similarly, quadratic programs are solved to move the back legs in the same general robot direction while restricting it to the back leg stability cone as in the following,

$$\min_{u_{b_r}^k} (u_{b_r}^k - d)^T (u_{b_r}^k - d),$$

such that

$$\begin{aligned} b_r^{k+1} &= g_r(b_r^k, u_{b_r}^k), \\ C_{b_r} b_r^{k+1} &\geq e_{b_r}, \end{aligned}$$

giving the back right leg control, u_{b_r}, and

$$\min_{u_{b_l}^k} (u_{b_l}^k - d)^T (u_{b_l}^k - d),$$

such that

$$\begin{aligned} b_l^{k+1} &= g_l(b_l^k, u_{b_l}^k), \\ C_{b_l} b_l^{k+1} &\geq e_{b_l}, \end{aligned}$$

gives the control, u_{b_l}, for the back left leg. The guard conditions are given by $b_{lref} = b_l^{k+1}$ and $b_{rref} = b_r^{k+1}$ for the back legs.

Finally, the second center of mass shift will move the center of mass back towards the back legs to widen the cone for the front legs and guarantee a stable leg configuration to move the front left leg. For this, the constraint, $C_{c2} c_m \geq e_{c2}$ is required, where

$$\begin{aligned} C_{c2} &= (R_{\pi/2}(f_r - b_l))^T (b_r - b_l)(R_{\pi/2}(f_r - b_l))^T \\ e_{c2} &= (R_{\pi/2}(f_r - b_l))^T (b_r - b_l)(R_{\pi/2}(f_r - b_l))^T b_l. \end{aligned}$$

The control for the second center of mass shift, u_{c2} is, therefore, given by the solution to

$$\min_{u_{c2}^k} (u_{c2}^k - d)^T (u_{c2}^k - d),$$

such that

$$\begin{aligned} c_m^{k+1} &= g_c(c_m^k, u_c^k), \\ C_{c2} c_m^{k+1} &\geq e_{c2}, \end{aligned}$$

with $d = \beta((b_r + b_l)/2 - c_m)$ for some scalar β. The guard on the second center of mass shift is given by $c_{ref} = c_m^{k+1}$.

This quadratic program is solved numerically. Together with low-level controllers discussed in the next section, these leg controllers complete the control framework required for the proposed gait.

6.4 Testbed Implementation

The proposed controller is implemented on a physical setup consisting of an operator interface and corresponding robot. Because the physical robot is currently in development, a dynamic simulation (Figure 28) is used in the robot's stead.

6.4.1 Hardware

The operator interface, depicted in Figure 28, consists of two three degree-of-freedom haptic joysticks that use admittance control to relate the operator's desired motions to the actual ones on the robot. Audio-visual feedback is provided through a headset, ensuring that the operator has extensive tele-presence. Each of the end effectors of the Phantom joysticks maps to an end effector of a robot leg and a set of switches is used to turn on the controller and switch between gait states. The joystick software is programmed in C and provides output that is sent via UDP to the robot, which uses a 1.4 GHz PC104 computer running xPC Target/Simulink coupled with a 2.9 GHz host computer.

6.4.2 Robot Dynamic Simulation

The software used is a dynamics library known as SrLib, developed by Seoul National University. This simulation accepts the same desired leg joint angles that would otherwise be sent to position controllers on the actual robot, and returns actual joint angles, position, and orientation, exactly as provided on the actual robot.

Figure 28: Operator Interface and CRR Simulation

6.4.3 Low-level Control Functions

Several key low-level functions that are required for the desired gait motion are built into the Simulink code on the target PC.

Motion of the front legs is achieved by mapping the end effector position of the Phantom joysticks to local leg workspaces on the shoulder, and then mapping these positions to desired robot joint angles. When an individual leg is being moved, the others are held steady, and the unused Phantom joystick remains still to represent the leg that is fixed. With the proposed controller, the user's commands are adjusted by the controller, resulting in a sense of haptic guidance.

Center of mass shifts are commanded by defining a vector pointing from the robot's current center of mass to the desired one, redefining the vector at each operation cycle. The intent is to then shift the robot's center in this direction, while keeping the feet fixed. As is standard practice for parallel robots, the robot is interpreted as four serial robots anchored to the ground by ball-and-socket joints that are mutually linked by a platform. New joint angles are calculated based on incremental motions of this platform with fixed end effector positions. These joint angles are commanded simultaneously to each leg at every operation cycle.

Rear leg motions required a desired displacement. A third-order curve is then fit to the resulting necessary trajectory, mapped to leg joint angles, and commanded to the robot while the other legs are held steady.

6.4.4 Adjustments and Limitations

The described hardware and software configuration enables a proof-of-concept and a demonstration of the feasibility of the user interface. However, there are a few limitations.

First, while the simulation is able to simulate near ideal conditions, some slippage still occurs. To compensate for this effect, the desired center of mass is updated during shifting, which is done by changing the line along which the desired center of mass is projected as the feet slide.

Second, the center of mass shifts and rear leg motions use path trajectories that are

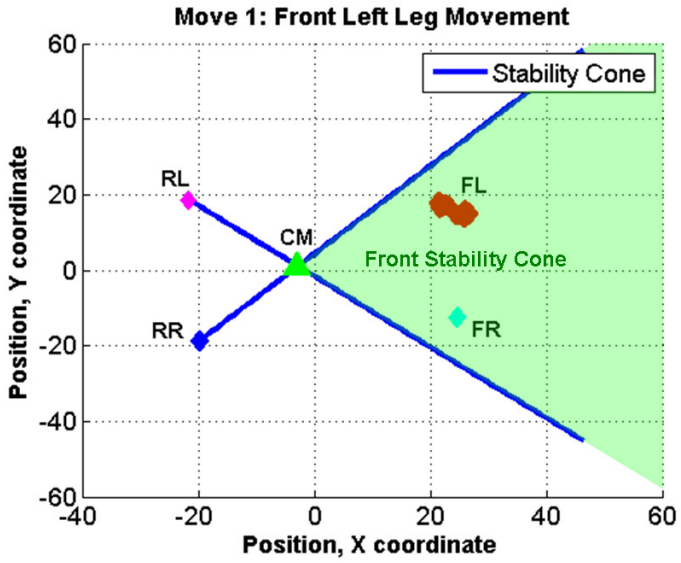


Figure 29: Front left leg moving freely within the stability cone limits as commanded by the operator. FL = Front Left, FR = Front Right, RL = Rear Left, RR = Rear Right, CM = Center of Mass.

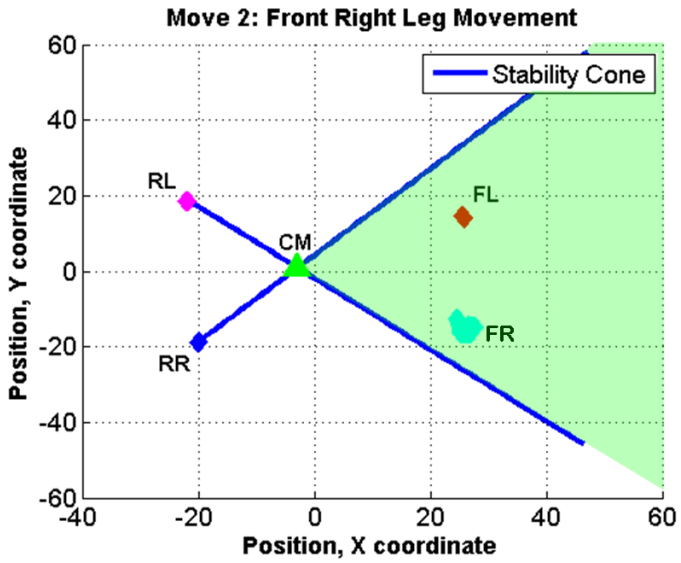


Figure 30: Robot in a statically stable position while the front right leg is moved freely within the stability cone limits as commanded by the operator.

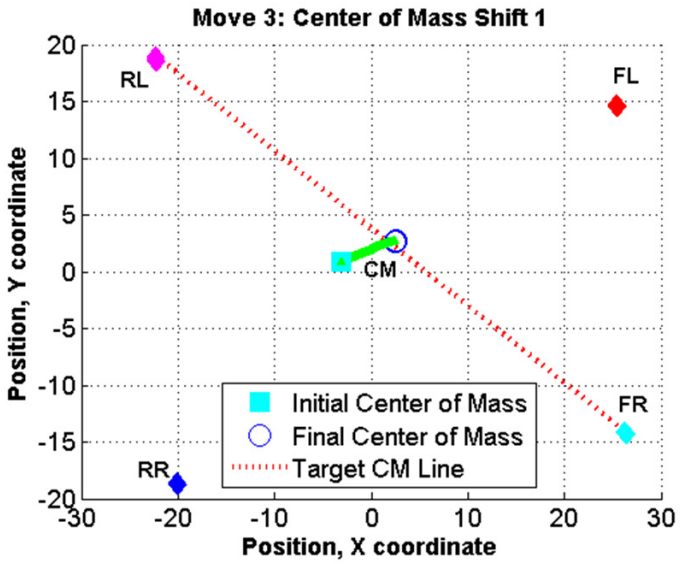


Figure 31: First center of mass shift ensures that the right leg can moved while maintaing static stability.

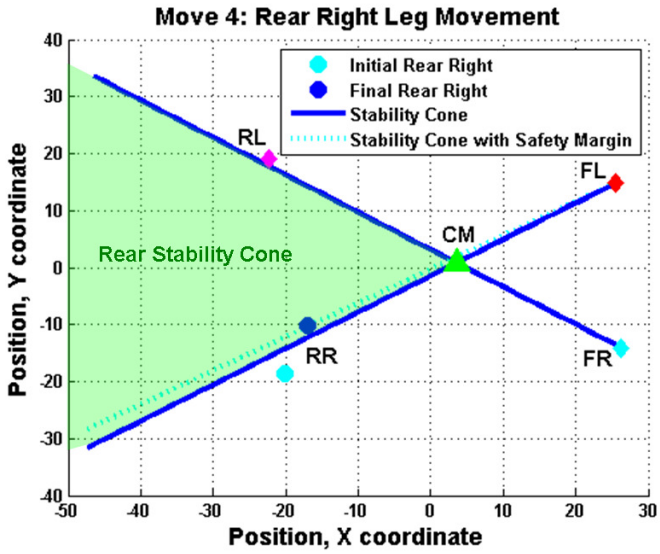


Figure 32: Rear right leg motion results in a statically stable configuration that allows the left leg to be moved.

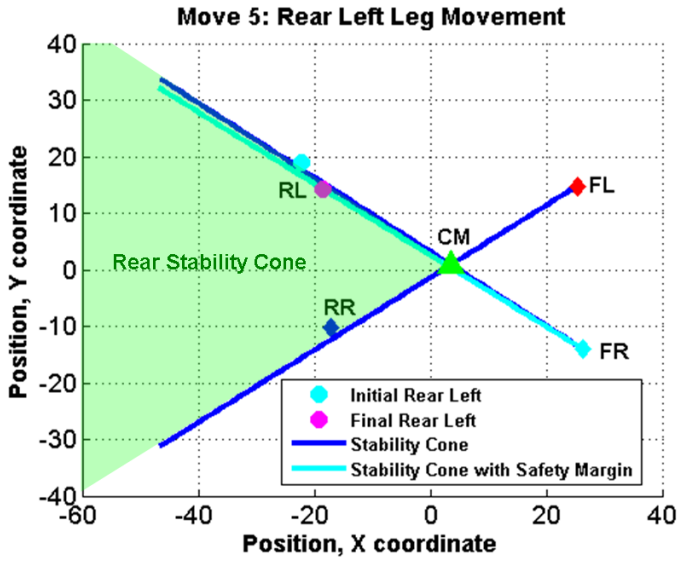


Figure 33: Rear left leg is moved inside the cone.

designed based on general feasibility, and have no measure for dealing with joint angle limits or other predefined obstacles. This could be improved for field implementation by using planners to determine the paths taken between the start and end positions.

6.5 Experimental Results

A complete gait cycle, starting from a statically stable position, was run, and the results were visually documented using the top-down labelled views shown in Figures 30 - 35. The goal here was to demonstrate that the gait developed in this chapter can be successfully implemented, ensuring that the user's motion was appropriately constrained and that the robot always maintained a statically stable position, while also contributing towards forward motion. The proposed front leg controller was implemented with a control horizon of $N = 5$ for a 0.002 second sampling rate.

Figure 29 depicts the initial position, as well as Move 1: the user's motion of the front left leg is constrained by the front stability cone, which is represented by the shaded green area within the blue lines. Physically, the user is able to feel these limits through the use of haptic joysticks in the form of moving into a soft wall.

After placing the front left leg, the user cycles to Move 2 and places the front right leg. Figure 30 shows that placement of this leg is again constrained by the stability cone.

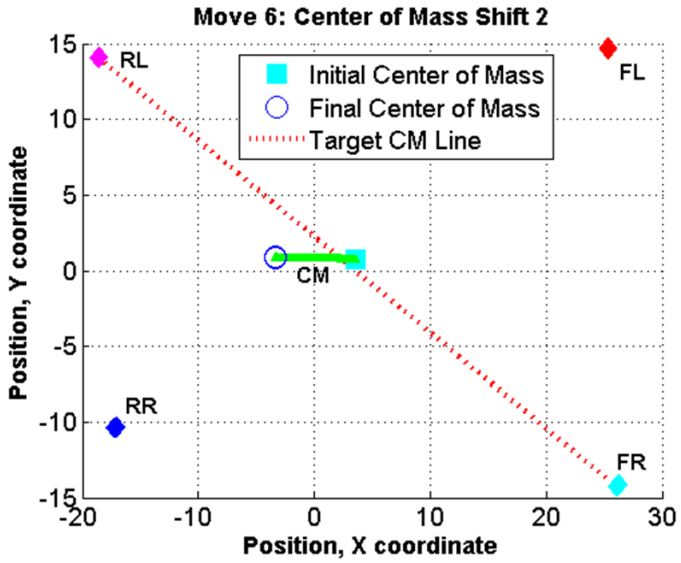


Figure 34: Second center of mass shift widens the cone for freer motion constraints on the front legs.

Move 3, shown in Figure 31, is the first center of mass shift. Here, the center of mass (labelled in its final position) is shown to be effectively shifted forward to a desired center of mass with a safety factor to ensure that the robot is not merely marginally stable.

Following the first center of mass shift, the rear legs are moved autonomously from their original point to a desired end location in Moves 4 and 5. Because of physical configuration constraints, extra measures were introduced in the rear right leg shift to ensure that the required motions would be made kinematically feasible. Specifically, instead of simply moving the leg from it's initial to final position, it was moved to a final 'safe' position, defined by the mechanical configuration of the robot, and then shifted the rest of the way to ensure that the final position ended inside the cone. While these extra measures are not ideal, they are an artifact resulting from reduced flexibility when working with an existing robot. Rather than demonstrating the entire motion, Figure 32 shows just the final position of the rear right leg following the right leg placement, which demonstrates that the leg is safely inside the cone. Here, the dashed cyan line again depicts the usage of a safety factor to ensure that the robot is not merely marginally stable.

The results of the rear left leg motion, Move 5, are shown in Figure 33. The left leg has successfully reached a statically stable final position along the dashed cyan line, which

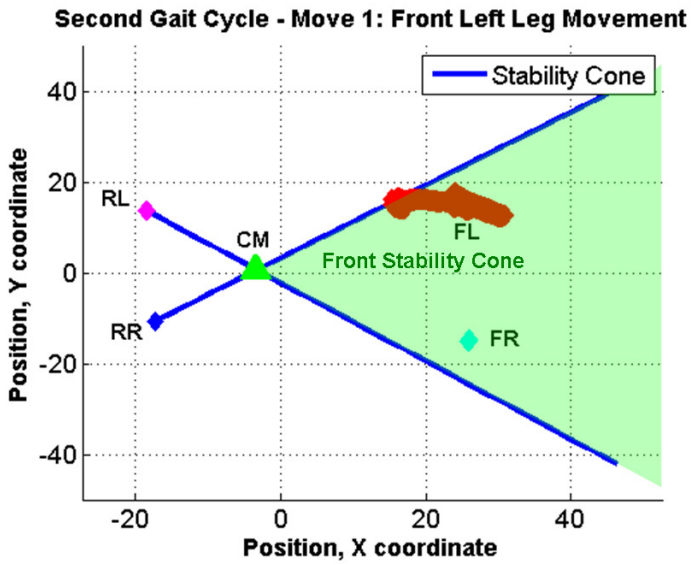


Figure 35: Second gait cycle, the front left leg can again be placed successfully be the user within the range defined by the cone.

again represents the desired point within the cone including a given safety margin.

Move 6 is shown in Figure 34. This second shift is intended to widen the cone for the front legs and ensure continued forward motion. It can also be observed that the forward and rear legs have advanced in the general robot direction; therefore, forward locomotion is achieved.

Finally, Figure 35 shows the start of the next gait cycle. The user is again constrained by the stability cone, and is free to continue moving forward.

Therefore, these results show that the human operator is able to carry out the higher-level task of placing the front legs in locations they deem appropriate (i.e in cluttered environments, the leg can be placed on relatively flat surfaces), while the controller and automatic gait ensure the lower-level task of ensuring static stability is completed.

6.6 Conclusions

In summary of this chapter, we applied the control framework to incorporate human input commands to control the front legs of a quadruped rescue robot as well as perform automated back leg and center of mass movements to ensure static stability. The controller

guides the user to place each of the front legs in a way that ensures the next front leg movement or any subsequent leg and center of mass movements will result in static stability for the robot. The viability of the controller was demonstrated on a hardware-in-the-loop simulation with real human input.

This application of the control allows a single human operator to cooperate with a single robot to determine the best actions for the robot in a USAR domain task. This control framework allows us to more easily control robotic systems that would otherwise be a challenge to control. The ability to offload a particularly difficult aspect of a human-robot task to a controller expands the types of problems humans and robots can cooperatively tackle.

In the next chapter, we expand the human-in-the-loop control framework to include multiple operators and multiple robots, which enables us to include a greater number of robots into the system. The human-in-the-loop control framework enables us to specify flexible formations for the robots to maintain, while human operators can manipulate the flexible formation to carry out high-level tasks. The result is a Multi-User Swarm-Level Interaction (MUSLI) where a small number of operators can control the shape of a robot swarm, and we apply this MUSLI control to a swarm manipulation task.

CHAPTER VII

MULTI-OPERATOR, MULTI-ROBOT CONTROL

In this chapter, we expand upon the single-operator, single-robot paradigm developed and studied in the previous chapters by addressing the problem of multiple operators controlling multiple robots and furthermore a large number of robots. Specifically, we employ the developed control framework to multi-operator control of a team of leader robots which enables the control of a large swarm of robots that follow these leader robots. In this way, we will create a Multi-User Swarm-Level Interaction (MUSLI) to allow multiple operators to control the shape of a robot swarm. We will apply MUSLI to the task of moving an object to a goal area and conduct multi-operator studies to determine the effectiveness of this form of human-swarm control.

7.1 Problem Formulation

We model the multi-agent interactions and formations as graphs with nodes representing agents and edges correspond to the existence of pairwise distance constraints. This approach allows us to take advantage of certain properties of graphs. In the following section, a brief overview of the graph notation used in this chapter is presented, followed by graph-based methods for defining formations. The formations will be described using inter-agent Euclidian distances, which will be defined as a state constraint on the system. This is constraint is used in the discussed control framework, and we present a solution to the associated optimal control problem. We will present a novel solution to the optimal control problem and a multi-operator experiment to study the efficacy of such a controller. Note that in this chapter, we return to the control framework notation used in Chapter 3.

7.1.1 Graph Notation

A graph, G, is defined by a node set, $V = \{1, 2, 3, \ldots, \mathbf{N}\}$ of $\mathbf{N}$ nodes and an edge set $E \subset V \times V$ of $\mathbf{M}$ unordered node pairs. We label each edge in E as $1, \ldots, \mathbf{M}$. Two nodes,

i and j, are adjacent, or neighbors, if $(i,j) \in E$. The neighborhood set of a node $i \in V$, $\mathbb{H}_i$, is the set of nodes $j \in V$ adjacent to node i. Edges can be given an orientation using $\sigma : E \to \{-1, 1\}$, resulting in a directed graph, G^σ, for which an associated incidence matrix, $D = [D_{ij}] \in \mathbb{R}^{\mathbf{N}\times\mathbf{M}}$, has elements given by

$$D_{ij} = \begin{cases} 1 & \text{if vertex } i \text{ is the tail of edge } j \\ -1 & \text{if vertex } i \text{ is the head of edge } j \\ 0 & \text{otherwise} \end{cases}, \tag{31}$$

7.1.2 Multi-agent Model and Dynamics

The multi-agent network in this thesis is modeled by a graph, $G = (V, E)$ where the $\mathbf{N}$ nodes correspond to $\mathbf{N}$ agents and each of the $\mathbf{M}$ edges of the network correspond to a pairwise distance constraint between two agents.

Each agent in the system has states that evolve with respect to discrete time, k, which we will denote as $x_{i,k} \in \mathbb{R}^n$ for agent i's state for all $i \in V$. In this chapter, each agent will be modeled as having linear dynamics.As such, we have a discrete-time dynamic system for $\mathbf{N}$ agents, where each agent $i \in V$ has discrete-time dynamics, $x_{i,k+1} = A_i x_{i,k} + B_i u_{i,k}$ where $u_{i,k} \in \mathbb{R}^m$ denotes the applied system input to agent i at time k and is calculated based on $v_{i,k} \in \mathbb{R}^m$ denoting human input. Note that the dynamics of each agent are decoupled from the state of other agents.

Hence, we can collect the states, $x_k = (x_{1,k}, \ldots, x_{\mathbf{N},k})$, control inputs, $u_k = (u_{1,k}, \ldots, u_{\mathbf{N},k})$, and human inputs, $v_k = (v_{1,k}, \ldots, v_{\mathbf{N},k})$ as vectors and describe the dynamics of the entire system of $\mathbf{N}$ agents as

$$x_{k+1} = Ax_k + Bu_k \tag{32}$$

with $A \in \mathbb{R}^{\mathbf{N}n\times\mathbf{N}n}$ and $B \in \mathbb{R}^{\mathbf{N}n\times\mathbf{N}m}$ where multiple human operators issue commands $v_k \in \mathbb{R}^{\mathbf{N}m}$.

7.1.3 Formations as State Constraints

The following discussion deals with how to specify formations based on inter-agent distances, i.e. a collection of pairwise constraints on states. Let an edge denote that an associated

pairwise distance constraint exists on agents i and j for all nodes $i, j \in V$ where $(i, j) \in E$, and let this edge be labeled as the lth edge. This pairwise constraint between agents i and j is defined as

$$||x_{i,k} - x_{j,k}||^2 = d_l, \tag{33}$$

with $d_l \in \mathbb{R}$ for $l \in \{1, \ldots, \mathbf{M}\}$ and $||.||$ representing the 2-norm (Euclidean). Thus, the specified inter-agent distance is $\sqrt{d_l}$. The collection of these constraints over the entire graph denotes the global constraint which we can write in terms of the well-known rigidity matrix, $R(x)$ with $\mathbf{M}$ rows and $\mathbf{N}n$ columns (for an example see [47]). Hence, the global constraint is given by

$$R(x_k)x_k = d \tag{34}$$

where $d = (d_1, \ldots, d_{\mathbf{M}})$ is the vector of distances desired for each edge. Note that this state constraint is a level-set of a quadratic function.

7.2 Multi-Operator, Multi-Robot Control

We have defined the dynamics of the multi-agent system, and we have defined the constraint set that specifies the low-level task (i.e. formation maintenance). Thus, we can apply the developed control framework and solve the resulting optimal control problem. We refer to this control as Multi-Operator-Multi-Robot (MOMR) control for the remainder of this chapter.

Given dynamics (32) and constraints (34), it is required that we solve $\mathcal{P}_{N_k}$ with $\mathbb{X}_f = \{x \mid R(x)x = d\}$ and linear system dynamics, $f(x_k, u_k) = Ax_k + Bu_k$. The distance constraints are level sets of a quadratic function, making $\mathbb{X}_f$ a non-convex set. The following is a discussion giving more intuition on why this set is non-convex. We know a Euclidean ball (e.g. $\{x \mid x^T x \leq d)\}$) is a convex set which includes the "inside" of the ball (i.e. the set of points $\{x \mid x^T x < d)\}$). However, we have a collection of distance constraints (e.g. $\{x \mid x^T x = d)\}$), where the inside of the ball is excluded from the set, so a convex combination of points in the set do not necessarily result in a point in the set. In other words, using the definition of non-convex sets, if we draw a line between two points in the set, all points on the line do not necessarily lie in the set. Thus, we treat $\mathbb{X}_f$ as a non-convex

set.

Given that the terminal state is constrained to a non-convex set , we only look for locally minimal solutions to the optimal control problem. In the following section, we present a method to solve $\mathcal{P}_{N_k}$ by splitting the problem into two parts. The first part consists of simulating the human input prediction forward to find a human-desired formation (that does not necessarily satisfy the formation constraint) and then finding the "closest" constraint-satisfying formation (denoted as $c \in \mathbb{R}^{\mathbf{N}n}$) to the human-desired formation. The second part of the method solves $\mathcal{P}_{N_k}$ for $\mathbb{X}_f = \{c\}$, which is essentially a point-to-point transfer optimal control problem where the cost penalizes deviations from the human input.

This form of control can be used with both rigid as well as non-rigid graphs (see [47, 59, 77] for a discussion on rigid and non-rigid graphs), however, such control of non-rigid graphs allow for a greater amount of formation-level freedom for the human to interact with. Rigid graphs describe graphs where *all* inter-agent distances in the graph are maintained not just the inter-agent distances specified by the edges. Rigid graphs only allow the human operators to rotate and translate the formation, where non-rigid graphs allow for a "deformation" of the structure while maintaining the desired inter-agent distances. In this way, there is enough high-level task freedom to necessitate multiple operators. An example of this is given in Section 7.3, where operators can "actuate" a gripper formed by a multi-agent system.

7.2.1 Control Strategy

Our approach to solving the optimal control problem is to first find the formation that satisfies the constraint $\mathbb{X}_f$, that is "closest" to the formation the human operators are predicted to drive the system to. Specifically, based on the predicted input sequence, $\mathcal{V}$, the human-desired formation at the end of the horizon, $k + N_k$, is given by simulating the state forward,

$$y = A^{N_k} x_k + \sum_{i=k}^{k+N_k-1} A^{k+N_k-1-i} B v_i \in \mathbb{R}^{\mathbf{N}n} \tag{35}$$

Hence, we wish to find the closest formation $c \in \mathbb{R}^{\mathbf{N}n}$ to y that lies in the set $\mathbb{X}_f$. This

closest formation is defined as the solution to the following quadratic program,

$$\min_{c} J(c) = \frac{1}{2}||c-y||^2 \tag{36}$$

$$\text{such that} \tag{37}$$

$$R(c)c = d \tag{38}$$

A gradient descent algorithm (presented in continuous time) is proposed to tackle this constrained optimization problem such that we can take advantage of the structure of the rigidity matrix. Namely, the continuous time derivative of $R(c)c = d$ is given by $R(c)\dot{c} = 0$, meaning that the null space of the rigidity matrix can be used to create an update rule to find the closest formation in the constraint space. In other words, by projecting the gradient of the cost function, we can "slide" along the constraint space until we get close to the human-desired formation.

The following descent algorithm updates the formation value (i.e. $\dot{c}$) by attempting to follow the negative gradient of the cost function $-\frac{\partial J}{\partial c}$ while ensuring this update will result in a formation that satisfies the formation constraint. Therefore, with an initial guess $c(0)$ such that $R(c(0))c(0) = d$, we update the guess (i.e. $\dot{c}$) with a projection of $-\frac{\partial J}{\partial c}$ onto the null space of the rigidity matrix, $R(c)\dot{c} = 0$.

The following derivation is based on Hilbert's projection theorem (for example see [90]), where a point in a Hilbert space (i.e. $s \in \mathbb{R}^{\mathbf{N}n}$) is projected onto a constraint space (the subspace given by the null space of the rigidity matrix). In order to find an expression for the projection, we first find a subspace orthogonal to the constraint space, then translate the orthogonal space to the point s. Then, the projection is the point w that lies in the intersection of the constraint space and the translated orthogonal space. This process is shown graphically in Figure 36. Let $s = -\frac{\partial J}{\partial c}^T = -(c-y)$. The null space (i.e. the constraint space) is represented by $\mathcal{N}(R(c)) = \{x \mid R(c)x = 0\}$, while the space orthogonal to the null space is represented by $\mathcal{N}(R(c))^\perp = \{z \mid z = R(c)^T p \text{ for some } p \in \mathbb{R}^{\mathbf{M}}\}$. As seen in Figure 36, we translate this orthogonal space to the point representing the negative of the gradient (i.e. s) by

$$\mathcal{N}(R(c))^\perp + s = \{w \mid w = R(c)^T p + s \text{ for some } p \in \mathbb{R}^{\mathbf{M}}\}. \tag{39}$$

Referring to Figure 36, the vector that lies in both $\mathcal{N}(R(c))$ and $\mathcal{N}(R(c))^{\perp} + s$ is sought or, in other terms, $w = R(c)^T p + s$ and $R(c)w = 0$. In order to find this point, we use both of these terms to find an expression for p,

$$\begin{aligned} 0 &= R(c)R(c)^T p + R(c)s \\ p &= -(R(c)R(c)^R)^{-1}R(c)s. \end{aligned} \tag{40}$$

This can be plugged back into $w = R(c)^T p + s$ to give,

$$\begin{aligned} w &= -R(c)^T(R(c)R(c)^T)^{-1}R(c)s + s \\ &= (I - R(c)^T(R(c)R(c)^T)^{-1}R(c))s, \end{aligned} \tag{41}$$

where I is the $(\mathbf{N}n) \times (\mathbf{N}n)$ identity matrix. Thus, w gives the projection of s onto the $\mathcal{N}(R(c))$. Note $R(c)^T(R(c)R(c)^T)^{-1}R(c)$ is the well-known projection matrix for $R(c)^T$ that we will denote as $Q(c)$. As such, $Q(c)$ projects vectors onto $\mathcal{N}(R(c))^{\perp}$, and as a consequence, $(I - Q(c))$ is a projection matrix onto the null space of the rigidity matrix (i.e. $\mathcal{N}(R(c))$). Rewriting (41) with $s = -(c - y)$, we now update the formation guess of the closest formation by,

$$\dot{c} = -(I - Q(c))(c - y). \tag{42}$$

The result of this gradient projection is a "sliding" of the initial guess along the null space until the gradient of the cost function is orthogonal to the null space, (i.e. the local minimizer of $J(c)$). Hence, our guess of the formation (c) slides along the constraint set ($\mathbb{X}_f$) until

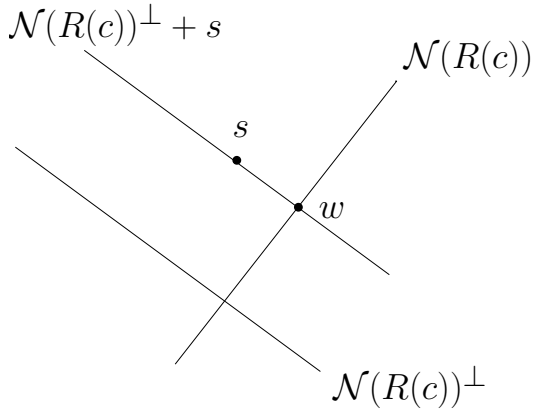


Figure 36: Diagram of Gradient Descent Projection

c is "close" to the human-desired formation. The following lemma shows that $(I - Q(c))$ is indeed a projection matrix, which will be then used in the following proof for gradient descent convergence to a local minimum.

Lemma 7.2.1 *$I - Q(c)$ is a projection matrix.*

Proof 3 *The matrix needs to satisfy $Q(c)^2 = Q(c)$ and $Q(c)^T = Q(c)$ to be a projection matrix. The first is satisfied by*

$$(I - Q(c))^2 = I^2 - 2Q(c) + Q(c)^2 = I - 2Q(c) + Q(c) = I - Q(c) \tag{43}$$

and

$$(I - Q(c))^T = I^T - Q(c)^T = I - Q(c). \tag{44}$$

Hence, $I - Q(c)$ is a projection matrix.

We state the update rule as a theorem and prove that the update rule results in a solution where the gradient is orthogonal to the null space.This solution implies that we cannot update our formation (i.e. $c \in \mathbb{X}_f$) without moving "away" from the human-desired formation (i.e. increasing the cost).

Theorem 7.2.2 *The gradient descent algorithm given by (42) converges to the closest local (with respect to initial guess $c(0)$) formation to y.*

Proof 4 *Let $\mathbb{V} = \frac{1}{2}||e||^2$ be a candidate Lyapunov function, where $e = (c - y)$. Then, we need to show $\dot{\mathbb{V}}$ is negative semi-definite for all $e \in \mathbb{R}^{\mathbf{N}n}$. Taking the time derivative of $\mathbb{V}$,*

$$\begin{aligned}
\dot{\mathbb{V}} &= \frac{\partial}{\partial t}(\frac{1}{2}e^T e) \\
&= e^T \dot{e} = e^T \dot{c} \\
&= e^T(-1)(I - Q(c))(c - y) \\
\dot{\mathbb{V}} &= -e^T(I - Q(c))e.
\end{aligned} \tag{45}$$

By Lemma (7.2.1), $(I - Q(c))$ is a projection matrix and we use the fact that projection matrices are positive semi-definite to show that $\dot{\mathbb{V}}$ is semi-negative definite by the definition of

semi-negative definite matrices. It remains to examine system behavior when $\dot{\mathbb{V}} = 0$. *Using LaSalle's principle, we show that* $\dot{\mathbb{V}} = 0$ *when* e *is orthogonal to* $\mathcal{N}(R(c))$ *or, equivalently,* $\dot{\mathbb{V}} = -e^T(I - Q(c))e = 0$ *when* $e \in \mathcal{N}(I - Q(c))$. *Examining* $\mathcal{N}(I - Q(c))$, *we see*

$$\begin{aligned} \mathcal{N}(I - Q(c)) &= \{z \mid (I - Q(c))z = 0\}, \\ \mathcal{N}(I - Q(c)) &= \{z \mid z - Q(c)z = 0\} \end{aligned}$$

while

$$Q(c)z \in \{q \mid q = R^T z\}.$$

Hence, we can rewrite the null space as

$$\begin{aligned} \mathcal{N}(I - Q(c)) &= \{z \mid z - R^T z = 0\}, \\ \mathcal{N}(I - Q(c)) &= \{z \mid z = R^T z\}. \end{aligned}$$

As a result, $e \in \{z \mid z = R^T z\} \subseteq \{w \mid w = R^T p, p \in \mathbb{R}^{\mathbf{M}}\}$. *Therefore,* $e \in \mathcal{R}(R(c)^T)$ *and orthogonal to* $\mathcal{N}(R(c))$. *In other words, the 2-norm error from the human desired formation cannot be minimized without increasing cost, so this is the "closest" formation to what the human desired while still in the constraint set (i.e. maintaining formation).*

This gradient descent technique converges to a point where the gradient of the cost function is orthogonal to the null space of the rigidity matrix, finding the closest formation to the desired predicted human formation that satisfies the formation constraint.

In implementation, our initial guess for $c(0)$ is the y found in the previous run of this gradient descent technique, thus we get local formation solutions that are similar to one another. In other words, we do not get jumps in formations which would make for choppy formation motion and cause confusion among the operators.

Finally in the second part of the control, this formation is used to solve $\mathcal{P}_N$ for $\mathcal{X}_f = \{c\}$. This simplification allows for the use of the closed-form solution, (4), where $M = I \in \mathbb{R}^{\mathbf{N}n \times \mathbf{N}n}$ and $b = c$ to drive the system to the desired formation while trying to minimize deviations from the predicted human input.

7.3 MUSLI: Swarm Gripper Control

In order to highlight the strengths of cooperative human-robot formation control, we apply the control to a Multi-User Swarm-level Interaction (MUSLI) scenario in which multiple human operators can influence large numbers of swarm agents through a smaller group of robots. There are numerous benefits to controlling swarms through a smaller group of robots. One such benefit is that centralized control is only required for the small group of robots, while the swarm robots can be controlled by decentralized algorithms. This presents clear computational and communication bandwidth benefits.

Optimal control of a large swarm of robots would become computationally intractable as the number of robots increases. On the other hand, optimal control of a small group of leader agents is feasible and practical, and this centralized control of small groups only requires communication of state and human input information on the leader agents. For a particular swarm robot, control can be implemented using only the states of other robots within a specified distance, and potential function-based swarm control can be used that is trivial to compute. This swarm control scheme can be implemented in a decentralized fashion, where each agent can compute their own control using only information communicated by neighboring agents. Hence, this control allows us to add swarm robots without affecting the tractability of the leader agent optimal control problem. In addition, this control allows a small number of operators to control the shape of a swarm of robots through the shared control of leader agents. As will be shown in this chapter, the task of swarm shape control is difficult for operators without the assistance of the developed control framework.

We apply the control framework to a task where human operators must drive a swarm of robots in order to manipulate an object in the simulation environment discussed in this section. The collective of robots is characterized as two separate types of robots. We refer to the first type as leader robots as these agents will be controlled in a centralized fashion (i.e. state information for all agents is known to the controller) to produce a "skeleton" formation for the second type of robot to follow. This second type will be referred to as swarm robots as they can only interact with other agents within a certain radius of itself. Specifically, these swarm agents will be attracted to leader agents and a combination of

attraction and repulsion to other swarm agents to produce a swarming behavior that "fill in" the skeleton structure created by the leader agents. Human operators will be able to directly send commands to the leader agents and therefore indirectly control the swarm agents.

Part of the overall task is to organize the leader agents in a "gripper" skeleton and have the swarm agents fill in this gripper to produce a "swarm gripper" as shown in Figure 42). In this scenario, human operators issue control commands to the ends of the gripper such that the operators can translate, rotate, and open/close(i.e. move the ends apart or together) the gripper structure. The defined gripper shape is a non-rigid formation in that the distance between the two ends of the gripper may vary (see [47] for a treatment on rigid formation structures) while the specified inter-agent distances are maintained. The low-level aspect of this task will be to maintain the gripper structure by maintaining select inter-agent distances as shown in Figure 37. Specifically, the operators will issue commands to the gripper end agents. These commands will then be used to generate commands for all the leader agents in order to maintain the specified inter-agent distances. The leader agent movements will induce swarm robot movement.

Operator interaction with the leader group will consist of two operators, where one operator issues commands to one end of the gripper (agent 6 in the graph) and the other operator issues commands to the other end (agent 7). Through just these two agents, the operators will have the freedom to translate, rotate, and open/close the gripper in order to move a ball (shown as a red circle in Figure 42) into the goal area (shown as a green square). The ball only interacts with the swarm agents so it is imperative to maintain the gripper shape so as to space the swarm agents along the gripper. Thus, the high-level aspect of the control is to move the ball into the goal area.

Clearly, only the leader agents at the gripper ends are receiving commands from the operators, so the operators require assistance in providing commands to the other 5 leader agents so as to maintain the gripper formation. The following sections detail a control method in addition to the method presented in this chapter as means of comparison. In the first approach, we utilize the MOMR cooperative control developed in this thesis to provide

formation assistance, and in the second approach, we utilize a common leader-follower control approach which we refer to as manual control. The leader agents (i.e. agents 1 to 5) try to assist in maintaining inter-agent distances, but the operators will have full control over the gripper ends. So, this method of control is more akin to manual control. In addition to controlling leader agents 1 to 5, the MOMR cooperative approach will modify operator commands to the gripper ends, as discussed throughout this thesis.

7.3.1 MOMR Control of Leaders

We apply the control developed earlier in this chapter, and the specific parameter values used follow. The leader agents are modeled by the graph shown in Figure 37 and the associated rigidity matrix is used to calculate the control. A fixed horizon with FOH human input prediction is used in this experiment as it was deemed effective in Chapter 5. Each robot's position is given in planar coordinates so $n, m = 2$. The dynamics matrices, distance vector, and control horizon used in the experiment are $A = B = I \in \mathbb{R}^{7n \times 7n}$, $d_i = 10$ (for $i = 1, \ldots, 8$), $d_9 = d_{10} = 20$, and $N_k = 5$ with $k = 0.2$ sec. Human inputs $v_1, \ldots, v_5$ in this case are all zero, while the human operators generate the inputs v_6 and v_7. We compare operator performance and workload using this method with that of manual control described in the following.

7.3.2 Manual Control of Leaders

We wish to compare our proposed control framework to manual control, however, with only two operators and 7 leader agents, having the two operators manually controlling all 7 leaders agents was deemed infeasible. Therefore, the manual task will allow the two operators to issue commands to the gripper ends and then utilize a common leader-follower network control method (see [59] for example) to drive the remaining leader agents to maintain the inter-agent distances required.

Specifically, we utilize the following control law for the leader agents. In this subsection, we drop the discrete time k subscript in the state for convenience (i.e. x_i to denote $x_{i,k}$).

We construct a weight matrix, $W(x) = diag(w_1, \ldots, w_{10})$, where

$$\begin{aligned}
w_1 &= ||x_1 - x_2||^2 - d_1,\ w_2 = ||x_1 - x_3||^2 - d_2 \\
w_3 &= ||x_2 - x_3||^2 - d_3,\ w_4 = ||x_1 - x_4||^2 - d_4 \\
w_5 &= ||x_1 - x_5||^2 - d_5,\ w_6 = ||x_4 - x_5||^2 - d_6 \\
w_7 &= ||x_5 - x_7||^2 - d_7,\ w_8 = ||x_3 - x_6||^2 - d_8 \\
w_9 &= ||x_2 - x_6||^2 - d_9,\ w_{10} = ||x_4 - x_7||^2 - d_{10}.
\end{aligned}$$

This matrix, along with the incidence matrix (31), is used to derive the Laplacian matrix (see [59] for details) for the graph defined as

$$L_w = DW(x)D^T,$$

which we use to express the control inputs to the follower agents. Agents 6 and 7 are leaders that are completely controlled by the human operators and agents $1-5$ are followers that simply attempt to maintain the inter-agent distances specified in the graph. As such, the control inputs to the 7 agents are given by

$$u_k = \begin{bmatrix} L_f \\ 0_{2m \times 7Nn} \end{bmatrix} x_k + \begin{bmatrix} 0_{5m \times 2m} \\ I_{2m \times 2m} \end{bmatrix} \begin{bmatrix} v_6 \\ v_7 \end{bmatrix} \tag{46}$$

where $L_f \in \mathbb{R}^{5m \times 7Nn}$ denotes the first 5 rows of L_w and $0_{g \times h}$ denotes a matrix in $\mathbb{R}^{g \times h}$ with all elements being 0. $I_{g \times h}$ denotes the identity matrix in $\mathbb{R}^{g \times h}$. What results is a system where the ends of the gripper (agents 6 and 7) are completely controlled by the human operators and each of the other agents attempt to maintain inter-agent distances with agents that share an edge.

Although, the 5 follower agents will automatically attempt to maintain the required inter-agent distance, the human operators have control over the two gripper end agents and can drive these agents to positions that make the desired formation constraint infeasible to achieve. Therefore, in this manual mode, the human operators are still required to drive the gripper ends in a way such that the rest of the leaders can achieve the formation described by the formation constraint.

As will be shown in the following results, despite the remaining leader agents automatically following the operator-driven agents to maintain inter-agent distance, maintaining the desired formation is still very difficult and the control proposed in this paper is meant to assist in this task. Before describing the operator study details, we present the control laws governing the swarm agents.

7.3.3 Swarm Control

The swarm agents behave locally in that they only interact with leader agents and other swarm agents within a specified radius (called a δ disk). The state of each swarm agent is given by $s_i \in \mathbb{R}^2$ for $i = 1, \ldots, 150$. They interact with leader and swarm agents using the following consensus-based control laws that govern how the state evolves in time,

$$\dot{s_i} = \sum_{j \in \mathbb{H}_l} -(s_i - s_j)(a_l - b_l e^{-\frac{||s_i - s_j||^2}{c_l}}) + \sum_{j \in \mathbb{H}_s} -(s_i - s_j)(a_s - b_s e^{-\frac{||s_i - s_j||^2}{c_s}}) \qquad (47)$$

for $\mathbb{H}_l$ being the set of all leader agents within a distance of 1.0 units from swarm agent i and $\mathbb{H}_s$ being the set of all swarm agents within a distance of 0.1 units from swarm agent i. More details on this type of behavior for swarm agents are discussed in [33].

The parameters used in the following simulation are $a_s = 10$, $b_s = 100$, $c_s = 0.1$, where the neighborhood of any agent includes agents within the δ disk of 0.1 and $a_l = 100$, $b_l = 10$, $c_l = 0.5$ for a δ disk of 1.0.

7.4 Operator Study: Swarm Gripper

In order to show that multiple human operators can carry out high-level tasks while the low-level controller completes a lower-level task, a human operator study was conducted. Specifically, two operators issue commands to the two ends of the gripper as in Figure 42. These two agents along with the other five agents (shown in varying shades of grey/purple) are to be driven in formation (i.e. they are to maintain the distances shown in the blue solid lines between the leader agents). This formation is flexible in that there are a set of positions the ends of the gripper can be in such that all the agents maintain the required inter-agents distances.

The overall task has two parts in which a high-level task requires the operators to cooperate to drive the swarm gripper to collect a red ball and deposit in a goal square. The low-level task is to drive the gripper ends such that the leader agents maintain the inter-agent distances shown by the blue lines in Figure 42. Maintaining these distances will ensure that the leaders form the appropriate skeleton shape for the swarm agents to fill in. If the leader agents are too far or too close, the swarms agents will clump together and fail to maintain the shape of the gripper (see Figures 39 and 40 for examples of this).

Therefore, the presented human operator studies address the question of whether the proposed control method results in high-level task completion, low-level task completion (measured as penalty time where the error in any one of the required distances exceeds 10%), increased task performance (measured in time to place the ball in the goal square) over Manual control, and decreased operator workload (as measured by the NASA TLX workload survey [38]) over Manual control. For the rest of this chapter, we will refer to the developed MOMR controller as the Mixed Initiative(MI) controller and the manual controller as Manual control. Based on pilot studies, it is hypothesized:

Hypothesis: The MI controller will result in both low and high level task completion with the shortest total completion times (completion time plus penalty time) as well as the lowest operator workload when compared to Manual control.

To effectively assess performance of the human operators with and without the presented control framework, a counter-balanced 2x1 repeated measures experiment was conducted.

7.4.1 Participants

10 operators were recruited for the experiment from the Georgia Institute of Technology community. None of the operators have had previous experience with mobile robot control. Particpants were between the ages 20-60 with 7 female participants and 3 male participants.

7.4.2 Task

For each trial, the two participants were asked to drive the gripper-end agents as seen in Figure 42. The object is to move the swarm gripper in order to drive the ball into the goal square while maintaining the required inter-agent distances, shown as blue lines in the

figure. These lines turn red when the error from desired distance exceeds 10% of the desired distance and a time penalty is accrued for every time period any one of these edges is red. Both the task time (time to drive the ball into the goal) and the time penalty are displayed to the participants, and participants are instructed that performance will be evaluated by the sum of these two times.

The ball only interacts with the swarm agents, hence the operators must manipulate the swarm in order to move the ball into the goal area. The users interface with the simulation via a standard gamepad joystick as shown in Figure 41 and the state of the task is shown via the simulation graphical display shown in Figure 42.

7.4.3 Procedure

Participants read standard written instructions on the task and then are shown the task environment with verbal instructions on how to complete the task (i.e. they must move the ball into the goal square and maintain all required inter-agent distances). A time penalty will accumulate for any time that any one of the inter-agent distance lines are red and this will be added to the time it takes to move the ball to the goal square. The operators are instructed to push a gamepad button once the operators deem the ball has reached the square, which stops the timer.

The participants were allowed to practice the task with each controller (MI or Manual) before the participants conducted the recorded data trial. Operators were also allowed to verbally communicate with the each other in any way they chose. Each recorded run was followed by a NASA TLX workload survey. The order the participants used the two controllers were counter-balanced to account for any ordering effects. After the two recorded trials, the particpants were given an exit survey comparing the two controllers where controller preference, frustration, and trust were judged.

7.4.4 Results

In all trials, two human operators cooperatively working together were able to drive the ball into the goal square. This successfully shows that the operators were able to complete the high-level task for both the manual control and mixed-initiative control. But more

importantly, human operator commands were modified in the MI control and, yet, the operators were able to complete the high-level task. However, as shown in Figure 43, the penalty times for the Manual Control are very large and almost as much as the time required to complete the task, essentially doubling the time required to complete the high-level task. Also, the penalty times when using the MI control are much lower than with the Manual controller for every pair of operators. High-level task completion times are also shown in Figure 43 and as the reader can see, the task completion times are lower for each pair of operators with the MI controller than with the Manual controller. As such, the combined task completion times and penalty times are significantly lower for the MI controller than the Manual controller. With these results, the MI control leads to faster completion times for the high-level ball moving task while also maintaining the low-level task with less deviation from the formation. The non-zero penalty time for the MI control results from large/abrupt changes in operator input (which result in a poor prediction of future human input over the time horizon). This is further addressed in the discussion section. Mean task completion times were 86.3 seconds for the MI control and 214.9 seconds for the Manual control while the mean penalty times for each were 19.9 seconds and 214.3 seconds, respectively.

The mean total trial times (task completion plus time penalty) for the MI control was 106.2 seconds while the manual control task on average took 429.2 seconds. A repeated measures ANOVA was run in SPSS on the total trial times, resulting in a statistically significant difference between the controllers with a large effect size and large power ($p = 0.026$, power 0.737, partial $\eta^2 = 0.750$). Thus, users were able to complete the high-level task much quicker with the MI Controller than with the Manual control.

As for operator workload, Figure 44 shows that the NASA TLX workload scores tend to be lower with the MI controller than with the Manual controller. However, this is not the case for every operator and we utilize statistical analysis to draw conclusions about the workload scores. The mean NASA TLX workload score for the MI controller was 61.7 while the mean NASA TLX workload score for the Manual control was 78.0. A repeated measures ANOVA was conducted in SPSS showing a rejection of the null hypothesis with a medium effect size and large power ($p = 0.015$, power 0.755, partial $\eta^2 = 0.497$).

In a final survey, operators were asked to choose which controller they preferred to use again, which controller was the most frustrating to use, and which controller they trusted to help them complete the task. All 10 participants chose the MI controller as the preferred and trusted controller, while the Manual controller was chosen as frustrating.

7.4.5 Discussion

The majority of operators chose the technique of opening ends of the gripper around the ball, then rotating the formation so the open side of the gripper faced the goal. Finally, with ball within gripper shape, they would push the ball into the goal area. As such, the operator studies show that a pair of operators are able to complete the high-level manipulation task. However, the operators only successfully maintained the desired gripper formation with the MI control. With Manual control, it took the operators a significantly longer period of time to complete the high-level task, while also failing to adequately maintain the desired formation for almost all of the task time.

One significant reason the Manual Control task completion times are longer is that it was very difficult to maintain the gripper formation. This caused the swarm agents to clump together in round masses (like in Figure 40) in which case they no longer resembled the gripper. In these cases, the ball easily slipped through the gaps in the swarm and operators had to find creative solutions to move the ball. Clearly, maintaining multi-agent formations is a difficult task and nearly impossible to complete in addition to the high-level task. Without the assistance of the MI controller, the operators can place the gripper ends in positions where reaching the desired formation is infeasible (they require a violation of the required inter-agent distances). However, with the MI controller, the gripper ends are pushed away from such configurations while giving the operator the freedom to impart operator intention to the agents.

The advantage of the MI controller over the Manual control was obvious in both the task trial times as well as the penalty times. Additionally, we get a statistically significant workload advantage. Operators were clearly frustrated with the Manual control early in the trial and almost gave up in trying ensure the agents met the distance constraints. On the

other hand, the MI control trials clearly reduced the frustration with this low-level aspect of the task. Operators did comment that the trade-off for low-level task assistance was that the system responded more slowly with the MI control than with the Manual control. This follows intuition in that the operator commands were being modified to ensure that the formation constraint was being met. However, the authors speculate that, with the Manual control, the task response would feel more sluggish if the operators more carefully moved the agents in order to fulfill the lower-level task, thus moving the formation as slowly as the MI control does.

The penalty times for the MI control trials are non-zero, implying that the operators were able to drive the away from the desired formation (i.e. leave the constraint set) for some amount of time. This behavior emerges from the fact that the control is calculated using a predicted human input over a future time horizon. The time horizon allows for the possibility of temporarily leaving the constraint set depending on operator input. Also, the system will leave the constraint in the face of large/abrupt changes in human operator command, stemming from future human input predictions that do not adequately model future human input. However, the convergence results that come with the proposed MPC control framework guarantee that the system will be driven back to the constraint set if ever the state ever leaves the set. Thus, we see that time outside of the constraint set is very small with the MI control in contrast to Manual control. The large difference in penalty times from MI control to Manual control supports the advantages of the MI control approach.

In order to prevent these goal set departures, the time horizon would have to be shrunk to one time step when the system has reached the goal set, which would ensure the system never leaves the goal set; however, this *severly* limits the freedom the operator has to impart his/her high-level intentions for the system.

It should also be noted that this particular system could be used by one operator with two joysticks, but the operator workload would be high as saw that the workload for two operators using the Manual control is not insignificant. We were able to show that the presented control framework is effective for systems with more than one operator. We

postulate that one operator with the MI Control would, however, be a feasible scenario thanks to the low-level task assistance. Note that the presented example can be used for up to 7 operators, and, for any more operators, we would need to define an alternate set of dynamics for the optimal control solved at each time instant to incorporate more inputs.

7.5 Conclusions

In this chapter, we have extended the control framework developed in this thesis to a multi-operator, multi-robot setting, which facilitated the control of a large swarm of robots. This swarm gripper application exemplifies the advantages of the multi-operator/multi-robot paradigm in that a small group of operators (two) can control many agents (e.g. 157 agents: 7 leader agents, 150 swarm agents) to complete a manipulation task. The control affords the human operators the ability to complete a higher level-task while driving the system to states that satisfy the low-level task (i.e. maintain gripper formation). Human operator studies confirm that the presented control framework has advantages over more manual -like methods of control and that the operators are given sufficient control over the system to complete high-level tasks. The MOMR control paradigm required a novel control algorithm to solve the formation maintenance problem while allowing the operators high-level task freedom for a swarm manipulation task.

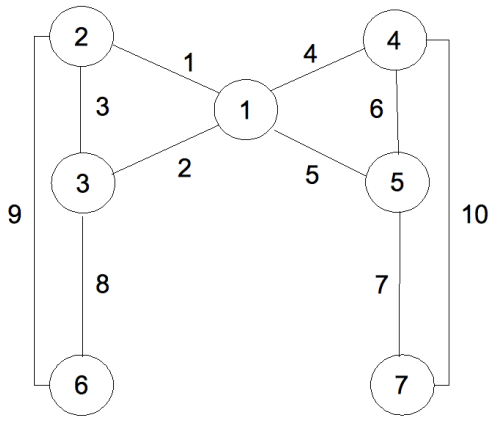


Figure 37: Underlying graph model of the leader agents and their distance-based constraints

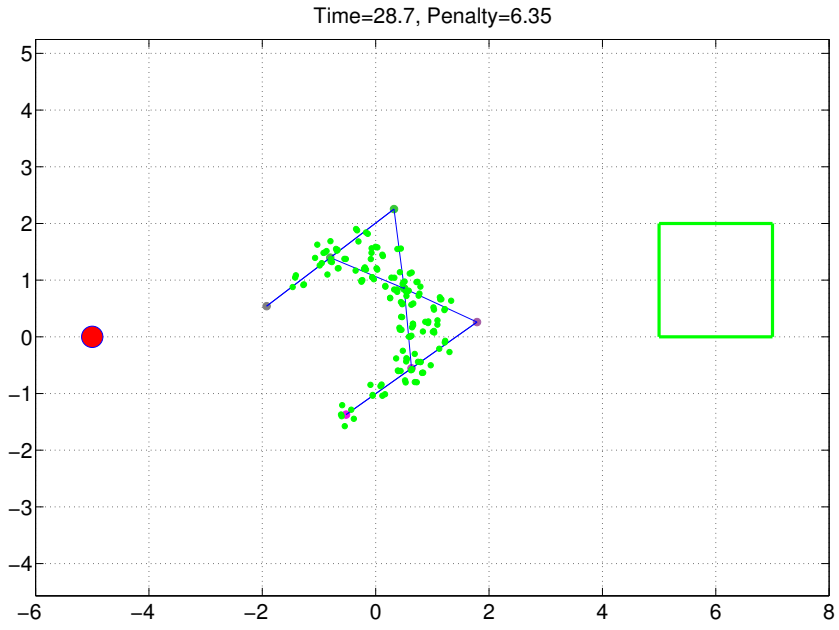


Figure 38: Graphical User Interface of swarm skeleton with swarm, goal square, and ball that only interacts with swarm agents

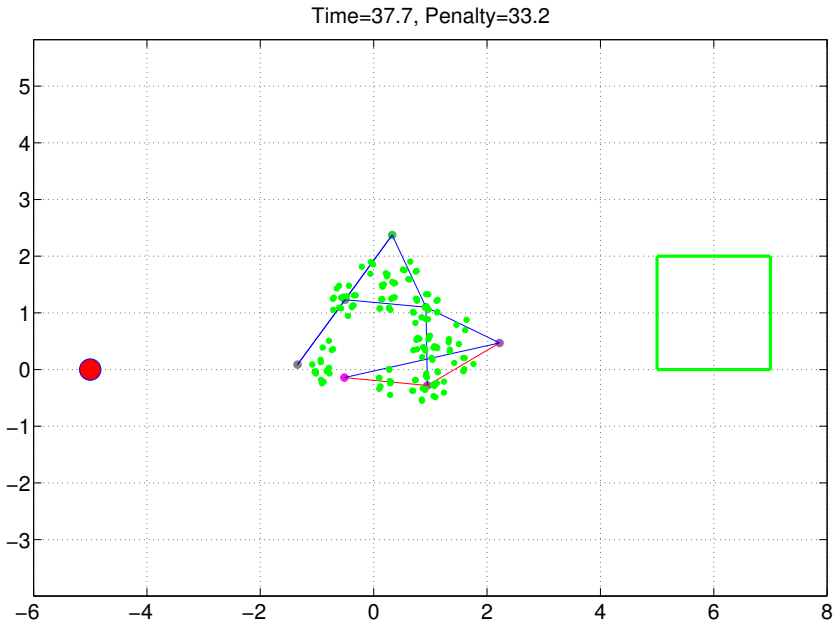


Figure 39: Example of gripper end placement where satisfying all required inter-agent distances is infeasible

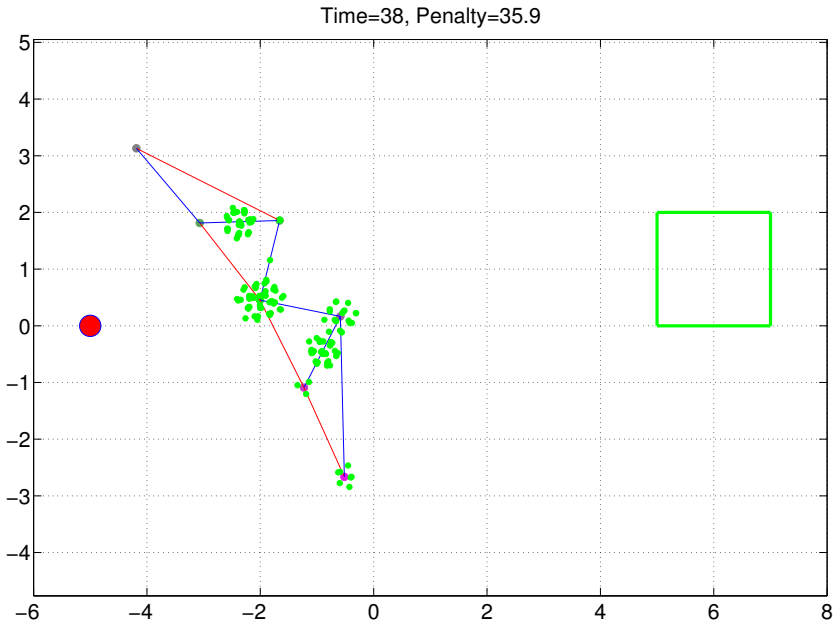


Figure 40: Example of swarm agent clumping that results from failure to maintain the required inter-agent distances.

Figure 41: Joystick used for Operator study

Figure 42: Operator study environment. Two operators utilize two joysticks to interact with the MUSLI simulation.

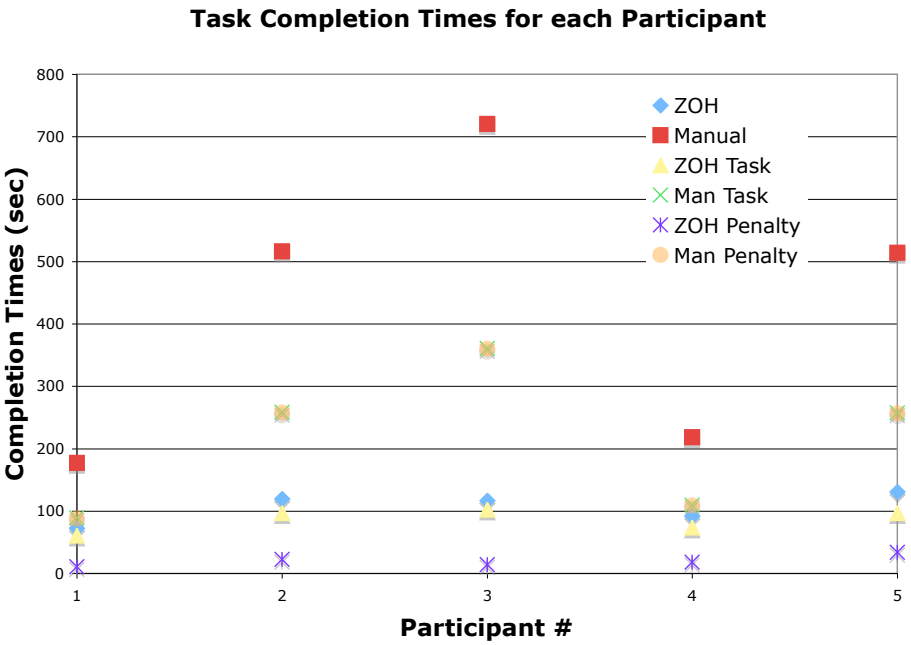


Figure 43: Task completion, penalty, and total times are shown for every participant pair with each of the MI and Manual controllers. The total Manual times are much longer than the MI times across all participants. Task completion times are also less for the MI control than the Manual control for all trials, and we see the Manual penalty times are significantly higher than the penalty times for the MI control. In fact, Manual penalty times are nearly the same as the Manual task completion times. Statistical analysis confirms the statistical significance of these performance advantages.

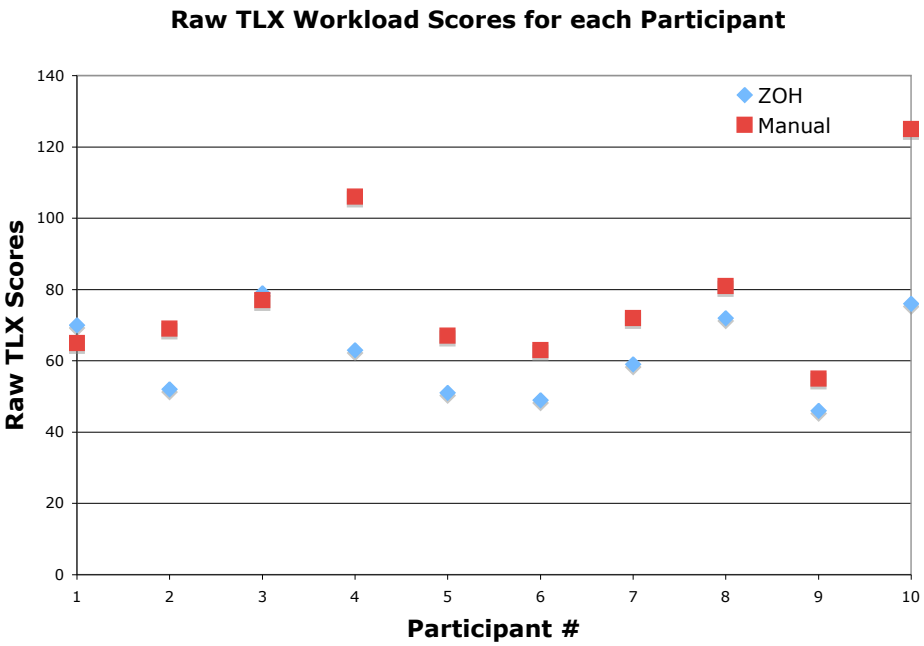


Figure 44: NASA TLX operator workload scores are shown for each participant with each of the two controllers. The operator workload scores tend to be higher for the Manual control over the MI control. Statistical analysis confirms the statistical significance of the operator workload advantages of MI control over Manual control.

CHAPTER VIII

CONCLUSIONS AND FUTURE WORK

8.1 Conclusions

To conclude, this thesis has presented a method for incorporating human operator input with an automatic controller that guarantees that a low-level task will be completed, while affording the operator freedom to complete a high-level task. This method also results in sliding autonomy where human and robot share control commensurate with human operator performance of the low-level task. Previous work in algorithms that combine user input with automatic control lacked guarantees of task completion for the automatic control in conjunction with sliding autonomy. The work in this thesis developed methods to address these gaps.

Specifically, this thesis's main contributions were:

- *Theoretical guarantees of low-level task completion*: This was proven theoretically for the presented model-predictive control framework using Lyapunov analysis, and the control provides robustness on par with infinite horizon optimal control.

- *High-level task completion affordance to human operators*: Through example tasks, human operator studies, and applications, the control framework provided human operators the freedom to carry out high-level aspects of the task despite sharing control of the system. Operators were able to carry out shared control tasks for single as well as multiple robot and swarm robot scenarios.

- *Low-level task performance-based sliding autonomy*: Human control influence was weighted higher when the operator adequately performs the low-level task and weighted lower when the operator has difficulty. In this way, we have developed a cooperative human-robot control framework where the automatic control complements the skills of the human operator when necessary.

- *Task performance and operator workload benefits*: These benefits were displayed for both the single-operator, single-mobile-robot navigation task as well as the MUSLI control task. In the navigation task, operators were able to complete both high and low level aspects of the task faster and with less workload. In the MUSLI scenario, the operators were able to carry out the high level aspect of the task as well as the low-level with workload benefits. Without the human-in-the-loop control framework, the operators performed poorly in the low-level aspect of the task.

- *Applications to a variety of cooperative human-robot tasks*: These applications include USAR robot tasks and multi-operator, multi-robot swarm control tasks. The developed controller was applied to an Urban Search and Rescue quadruped robot where human operators give commands to place the robot's feet in cluttered disaster environments while the controller guaranteed robot stability. The controller allows the operator the freedom to choose suitable feet placement in challenging terrain while the automatic control ensures that these feet placement will lead to stability in the next gait motion. The control framework was then extended to the multi-operator, multi-robot control paradigm. Human operators could manipulate flexible formations while the control algorithm assisted in ensuring the formation was maintained. The ability to effectively compose high and low level task commands enabled us to incorporate large numbers of robots resulting in human-swarm interaction. This human-swarm interaction allowed a small number of operators to control large swarms in a manipulation task.

These developments in human-robot cooperative control has resulted in following publications: [15, 14, 12, 13, 17, 16].

8.2 Future Work

The work in this thesis provides a basis for human-robot cooperative control with guarantees of task completion for both single-operator, single-robot and multi-operator, multi-robot scenarios. Additional advancements in the work presented in this thesis could lead to human-robot cooperation for a higher complexity class of low-level tasks and levels of human

input. One direction of future work would be to have human operators be able to specify more abstract levels input in addition to signal level inputs (as we have presented) depending on the operating environment. For example, operators could specify desired behaviors to the robots such as "open gripper" which would then generate a human input sequence for use in the control framework, but still allow the human to directly intervene when more detailed control is needed by issuing joystick commands.

Another future direction of work could be to investigate alternate methods for varying the control horizon. The method proposed in this thesis sought to vary horizons based on human input prediction performance, but alternate methods could vary the horizon based on high or low-level task performance since the horizon length had an effect on human-to-robot control influence weighting (recall that short horizons tended to weight robot control over human input). However, a balance between large and small horizons must be maintained in that larger control horizons could lead to degraded control performance due to poor performance of human input prediction schemes.

This leads to another future area of work in that the human operator studies hinted that more complex methods of human input prediction lead to poor operator performance. Further human operators studies would be required to confirm this notion along with some additional human input prediction techniques as developed in other HRI literature. Examples of other human input prediction techniques include machine learning, fuzzy-logic, and genetic algorithms.

The area of multi-operator, multi-robot control is still a growing area of research, and the work presented in this thesis could serve as a foundation to expand upon. Particularly, further analysis is required on swarm behaviors that follow multi-agent skeleton formations. Methods for specifying parameter values of the swarm control to guarantee skeleton coverage would lead to effective human-swarm interaction.

REFERENCES

[1] Aicardi, M., Casalino, G., Bicchi, A., and Balestrino, A., "Closed loop steering of unicycle like vehicles via lyapunov techniques," *Robotics & Automation Magazine, IEEE*, vol. 2, no. 1, pp. 27–35, 1995.

[2] Allen, J. E., Guinn, C. I., and Horvtz, E., "Mixed-initiative interaction," *IEEE Intelligent Systems and their Applications*, vol. 14, no. 5, pp. 14–23, 1999.

[3] Anderson, S., Peters, S., Pilutti, T., Tseng, H., and Iagnemma, K., "Semi-autonomous avoidance of moving hazards for passenger vehicles," in *Proceedings of the 2010 ASME Dynamic Systems and Controls Conference*, 2010.

[4] Arkin, R. and Ali, K., "Integration of reactive and telerobotic control in multi-agent robotic systems," in *From animals to animats 3: Proc. Third International Conference on Simulation of Adaptive Behavior*, pp. 473–478, 1994.

[5] Balch, T. and Arkin, R., "Behavior-based formation control for multirobot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926–939, 1998.

[6] Birk, A. and Kenn, H., "A control architecture for a rescue robot ensuring safe semi-autonomous operation," in *RoboCup-02: Robot Soccer World Cup VI* (Kaminka, G., Lima, P., and Rojas, R., eds.), LNAI, Springer, 2002.

[7] Bruemmer, D. J., Boring, R. L., Few, D. A., Marble, J. L., and Walton, M. C., "i call shotgun!: An evaluation of mixed-initiative control for novice users of a search and rescue robot," in *In Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, pp. 2847–2852, 2004.

[8] Bruemmer, D., Few, D., Boring, R., Marble, J., Walton, M., and Nielsen, C., "Shared understanding for collaborative control," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 35, no. 4, pp. 494–504, 2005.

[9] Bruemmer, D., Marble, J., Dudenhoeffer, D., Anderson, M., and McKay, M., "Mixed-initiative control for remote characterization of hazardous environments," in *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*, p. 9, IEEE, 2003.

[10] Burstein, M., Ferguson, G., and Allen, J., "Integrating agent-based mixed-initiative control with an existing multi-agent planning system," in *Proc. Fourth International Conference on MultiAgent Systems*, pp. 389–390, 2000.

[11] Burstein, M., Ferguson, G., and Allen, J., "Integrating agent-based mixed-initiative control with an existing multi-agent planning system," in *MultiAgent Systems, 2000. Proceedings. Fourth International Conference on*, pp. 389–390, IEEE, 2000.

[12] Chipalkatty, R., Daepp, H., Egerstedt, M., and Book, W., "Human-in-the-loop: Mpc for shared control of a quadruped rescue robot," in *Intelligent Robots and*

Systems (IROS), 2011 IEEE/RSJ International Conference on, pp. 4556–4561, IEEE, 2011.

[13] Chipalkatty, R. and Egerstedt, M., "Human-in-the-loop: Variable horizon mpc with human inputs," *Robotics and Automation, IEEE Transactions on*.

[14] Chipalkatty, R. and Egerstedt, M., "Human-in-the-loop: Terminal constraint receding horizon control with human inputs," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 2712–2717, IEEE, 2010.

[15] Chipalkatty, R. and Egerstedt, M., "Multi-operator-multi-robot control of swarms," in *Workshop on Algorithms for Robotics*, 2012. Under Review.

[16] Chipalkatty, R. and Egerstedt, M., "Constrained agreemet protocols for tree graph topologies," *International Journal of Control*, To Appear.

[17] Chipalkatty, R., Egerstedt, M., and Azuma, S., "Multi-pendulum synchronization using constrained agreement protocols," in *International Conference on Robot Communication and Coordination*, 2009.

[18] Connell, J. and Viola, P., "Cooperative control of a semi-autonomous mobile robot," in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, pp. 1118–1121, IEEE, 2002.

[19] Cummings, M., "Human supervisory control of swarming networks," in *2nd Annual Swarming: Autonomous Intelligent Networked Systems Conference*, 2004.

[20] De Santos, P. G., Estremera, J., Garcia, E., and Armada, M., "Including joint torques and power consumption in the stability margin of walking robots," *Auton. Robots*, vol. 18, no. 1, pp. 43–57, 2005.

[21] Desai, J., Ostrowski, J., and Kumar, V., "Controlling formations of multiple mobile robots," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 4, pp. 2864–2869, IEEE, 2002.

[22] Desai, M. and Yanco, H., "Blending human and robot inputs for sliding scale autonomy," in *Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on*, pp. 537–542, IEEE, 2005.

[23] Diftler, M., Culbert, C., Ambrose, R., Platt Jr, R., and Bluethmann, W., "Evolution of the nasa/darpa robonaut control system," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 2, pp. 2543–2548, IEEE, 2003.

[24] Ding, X., Powers, M., Egerstedt, M., Young, S., and Balch, T., "Executive decision support," *Robotics & Automation Magazine, IEEE*, vol. 16, no. 2, pp. 73–81, 2009.

[25] Droge, G. and Egerstedt, M., "Adaptive look-ahead for robotic navigation in unknown environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, To appear IEEE, 2011.

[26] Dunbar, W. and Murray, R., "Model predictive control of coordinated multi-vehicle formations," in *IEEE Conference on Decision and Control*, vol. 4, pp. 4631–4636, Citeseer, 2002.

[27] Enes, A., *Shared control of hydraulic manipulators to decrease cycle time*. PhD thesis, Georgia Institute of Technology, 2010.

[28] Estremera, J. and de Santos, P., "Generating continuous free crab gaits for quadruped robots on irregular terrain," *Robotics, IEEE Transactions on*, vol. 21, pp. 1067 – 1076, Dec. 2005.

[29] Estremera, J., Garcia, E., and Gonzalez De Santos, P., "A multi-modal and collaborative human–machine interface for a walking robot," *J. Intell. Robotics Syst.*, vol. 35, no. 4, pp. 397–425, 2002.

[30] Findeisen, R. and Allgaower, F., "An introduction to nonlinear model predictive control," in *In 21st Benelux Meeting on Systems and Control*, 2002.

[31] Finzi, A., "A mixed-initiative approach to human-robot interaction in rescue scenarios," in *in International Conference on Automated Planning and Scheduling (ICAPS), Printed Notes of Workshop on Mixed-Initiative Planning and Scheduling*, pp. 36–43, 2005.

[32] Fong, T., Thorpe, C., and Baur, C., "Multi-robot remote driving with collaborative control," *Industrial Electronics, IEEE Transactions on*, vol. 50, no. 4, pp. 699–704, 2003.

[33] Gazi, V. and Passino, K., "Stability analysis of swarms," *Automatic Control, IEEE Transactions on*, vol. 48, no. 4, pp. 692–697, 2003.

[34] Goodrich, M. A. and Schultz, A. C., "Human robot interaction: A survey," *Foundations and Trends in HumanComputer Interactio*, vol. Vol. 1: No 3, pp. pp 203–275, 2007.

[35] Goodwin, G., Seron, M., and Dona, J. D., *Constrained Control and Estimation*. Springer, 2005.

[36] Griffin, W., Provancher, W., and Cutkosky, M., "Feedback strategies for telemanipulation with shared control of object handling forces," *Presence: Teleoperators & Virtual Environments*, vol. 14, no. 6, pp. 720–731, 2005.

[37] Griffiths, P. and Gillespie, R. B., "Shared control between human and machine: haptic display of automation during manual control of vehicle heading," in *Proc. 12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems HAPTICS '04*, pp. 358–366, 27–28 March 2004.

[38] Hart, S. and Staveland, L., "Development of nasa-tlx (task load index): Results of empirical and theoretical research," *Human mental workload*, vol. 1, pp. 139–183, 1988.

[39] Heger, F. and Singh, S., "Sliding autonomy for complex coordinated multi-robot tasks: Analysis & experiments," *Proceedings, Robotics: Systems and Science, Philadelphia*, 2006.

[40] Hong, S.-G., Kim, B. S., Kim, S., and Lee, J.-J., "Artificial force reflection control for teleoperated mobile robots," *Mechatronics*, vol. 8, no. 6, pp. 707 – 717, 1998.

[41] Huang, Q.-J. and Nonami, K., "Humanitarian mine detecting six-legged walking robot and hybrid neuro walking control with position/force control," *Mechatronics*, vol. 13, no. 8-9, pp. 773 – 790, 2003. Computational Intelligence in Mechatronic Systems.

[42] Kazerooni, H. and Mahoney, S., "Force augmentation in human-robot interaction," in *American Control Conference, 1990*, pp. 2821–2826, IEEE, 2009.

[43] Kim, D.-J. and Behal, A., "Human-in-the-loop control of an assistive robotic arm in unstructured environments for spinal cord injured users," in *HRI '09: Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, (New York, NY, USA), pp. 285–286, ACM, 2009.

[44] Kinugawa, K. and Noborio, H., "A shared autonomy of multiple mobile robots in teleoperation," in *Robot and Human Interactive Communication, 2001. Proceedings. 10th IEEE International Workshop on*, pp. 319–325, IEEE, 2001.

[45] Kira, Z. and Potter, M., "Exerting human control over decentralized robot swarms," in *Autonomous Robots and Agents, 2009. ICARA 2009. 4th International Conference on*, pp. 566–571, IEEE, 2009.

[46] Kolter, J. Z., Rodgers, M. P., and Ng, A. Y., "A control architecture for quadruped locomotion over rough terrain," in *IEEE International Conference on Robotics and Automation. The Half-Day Workshop on: Towards Autonomous Agriculture of Tomorrow, 19-23 May*, pp. 4814–21, 2008.

[47] Krick, L., Broucke, M., and Francis, B., "Stabilisation of infinitesimally rigid formations of multi-robot networks," *International Journal of Control*, vol. 82, no. 3, pp. 423–439, 2009.

[48] Kuwata, Y., Richards, A., Schouwenaars, T., and How, J., "Decentralized robust receding horizon control for multi-vehicle guidance," in *American Control Conference, 2006*, pp. 6–pp, IEEE, 2006.

[49] Lee, S. and Kim, G. J., "Effects of haptic feedback, stereoscopy, and image resolution on performance and presence in remote navigation," *Int. J. Hum.-Comput. Stud.*, vol. 66, no. 10, pp. 701–717, 2008.

[50] Lewis, B. and Sukthankar, G., "Two hands are better than one: Assisting users with multi-robot manipulation tasks," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 2590–2595, IEEE, 2011.

[51] Lewis, M. and Sycara, K., "Network-centric control for multirobot teams in urban search and rescue," in *System Sciences (HICSS), 2011 44th Hawaii International Conference on*, pp. 1–10, IEEE, 2011.

[52] Ljung, L. and Ljung, E., *System identification: theory for the user*. Prentice-Hall Englewood Cliffs, NJ, 1987.

[53] Loizou, S. G. and Kumar, V., "Mixed initiative control of autonomous vehicles," in *Proc. IEEE International Conference on Robotics and Automation*, pp. 1431–1436, Apr. 10–14, 2007.

[54] Mano, H., Kon, K., Sato, N., Ito, M., Mizumoto, H., Goto, K., Chatterjee, R., and Matsuno, F., "Treaded control system for rescue robots in indoor environment," in *Proc. IEEE International Conference on Robotics and Biomimetics ROBIO 2008*, pp. 1836–1843, 22–25 Feb. 2009.

[55] Martin, P., de la Croix, J., and Egerstedt, M., "Pancakes: A software framework for distributed robot and sensor network applications," in *10th International Symposium on Distributed Autonomous Robotics Systems*, 2010.

[56] Mayne, D., "Model predictive control: The challenge of uncertainty," in *Model Predictive Control: Techniques and Applications-Day 1 (Ref. No. 1999/095), IEE Two-Day Workshop on*, p. 6, IET, 2002.

[57] Mayne, D., Rawlings, J., Rao, C., and Scokaert, P., "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, p. 789814, 2000.

[58] McLachlan, S., Arblaster, J., Liu, O., Miro, J., and Chenoweth, L., "A multistage shared control method for an intelligent mobility assistant," in *Rehabilitation Robotics, 2005. ICORR 2005. 9th International Conference on*, pp. 426–429, IEEE, 2005.

[59] Mesbahi, M. and Egerstedt, M., *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

[60] Michalska, H. and Mayne, D., "Robust receding horizon control of constrained nonlinear systems," *Automatic Control, IEEE Transactions on*, vol. 38, no. 11, pp. 1623–1633, 1993.

[61] Michelman, P. and Allen, P., "Shared autonomy in a robot hand teleoperation system," in *Intelligent Robots and Systems' 94.'Advanced Robotic Systems and the Real World', IROS'94. Proceedings of the IEEE/RSJ/GI International Conference on*, vol. 1, pp. 253–259, IEEE, 2002.

[62] Moosavian, S. A. A., Kalantari, A., Semsarilar, H., Aboosaeedan, E., and Mihankhah, E., "Resquake: A tele-operative rescue robot," *Journal of Mechanical Design*, vol. 131, no. 8, p. 081005, 2009.

[63] Morris, A., Donamukkala, R., Kapuria, A., Steinfeld, A., Matthews, J., Dunbar-Jacob, J., and Thrun, S., "A robotic walker that provides guidance," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 1, pp. 25–30, IEEE, 2003.

[64] Murphy, R. R., "Human-robot interaction in rescue robotics," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 34, no. 2, pp. 138–153, 2004.

[65] Nourbakhsh, I. R., Sycara, K., Koes, M., Yong, M., Lewis, M., and Burion, S., "Human-robot teaming for search and rescue," 2005.

[66] Nudehi, S., Mukherjee, R., and Ghodoussi, M., "A shared-control approach to haptic interface design for minimally invasive telesurgical training," *Control Systems Technology, IEEE Transactions on*, vol. 13, no. 4, pp. 588–592, 2005.

[67] Olfati-Saber, R., "Near-identity diffeomorphisms and exponential epsilon-tracking and epsilon-stabilization of first-order nonholonomic se (2) vehicles," in *American Control Conference, 2002. Proceedings of the 2002*, pp. 4690–4695, 2002.

[68] Olfati-Saber, R. and Murray, R. M., "Agreement problems in networks with directed graphs and switching toplogy," in *Proc. 42nd IEEE Conf. Decision Control*, vol. 4, (Maui, HI), pp. 4126–4132, Dec 2003.

[69] Olfati-Saber, R. and Murray, R., "Graph rigidity and distributed formation stabilization of multi-vehicle systems," in *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol. 3, pp. 2965–2971, IEEE, 2002.

[70] Parikh, S., Grassi Jr, V., Kumar, V., and Okamoto Jr, J., "Incorporating user inputs in motion planning for a smart wheelchair," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 2, pp. 2043–2048, IEEE, 2005.

[71] Parker, L. and Howard, A., "Assistive formation maintenance for human-led multi-robot systems," in *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, pp. 2350–2355, IEEE, 2009.

[72] Poncela, A., Urdiales, C., Perez, E., and Sandoval, F., "A new efficiency-weighted strategy for continuous human/robot cooperation in navigation," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 39, no. 3, pp. 486–500, 2009.

[73] Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., and Ng, A., "Ros: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[74] Saber, R., Richard, S., and Murray, R., "Consensus protocols for networks of dynamic agents,"

[75] Scokaert, P. O. M., Mayne, D. Q., and Rawlings, J. B., "Suboptimal model predictive control (feasibility implies stability)," *IEEE Transactions on Automatic Control*, vol. 44, no. 3, pp. 648–654, 1999.

[76] Sengstacken, A., DeLaurentis, D., and Akbarzadeh-T, M., "Optimization of shared autonomy vehicle control architectures for swarm operations," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 40, no. 4, pp. 1145–1157, 2010.

[77] Singer, A. and Cucuringu, M., "Uniqueness of low-rank matrix completion by rigidity theory," *Arxiv preprint arXiv:0902.3846*, 2009.

[78] Steele, M. and Gillespie, R., "Shared control between human and machine: Using a haptic steering wheel to aid in land vehicle guidance," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 45, pp. 1671–1675, SAGE Publications, 2001.

[79] Tanner, H., Jadbabaie, A., and Pappas, G., "Coordination of multiple autonomous vehicles," in *Proceedings of the 11th IEEE Mediterranean Conference on Control and Automation*, (Rhodes, Greece), June 2000.

[80] Thomas, M. and Kardos, L., "Shrinking horizon model predictive control applied to autoclave curing of composite laminate materials," in *Proceedings of the American Control Conference*, 1994.

[81] Tzafestas, S. G. and Tzafestas, E. S., "Human–machine interaction in intelligent robotic systems: A unifying consideration with implementation examples," *J. Intell. Robotics Syst.*, vol. 32, no. 2, pp. 119–141, 2001.

[82] Usher, K., "Human-in-the-loop pose control of a non-holonomic vehicle," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 3948–3953, IEEE, 2006.

[83] Voos, H., "Model predictive collaborative motion planning and control of mobile robots including safety aspects," in *Advanced Robotics, 2009. ICAR 2009. International Conference on*, pp. 1–6, IEEE, 2009.

[84] Wang, C., Ma, H., and Cannon, D. J., "Human-machine collaboration in robotics: Integrating virtual tools with a collision avoidance concept using conglomerates of spheres," *J. Intell. Robotics Syst.*, vol. 18, no. 4, pp. 367–397, 1997.

[85] Wasson, G., Gunderson, J., Graves, S., and Felder, R., "Effective shared control in cooperative mobility aids," in *In Proceedings of the Fourteenth international Florida Artificial intelligence Research Society Conference (May 21 - 23*, pp. 509–513, AAAI Press, 2001.

[86] Wasson, G., Sheth, P., Alwan, M., Granata, K., Ledoux, A., and Huang, C., "User intent in a shared control framework for pedestrian mobility aids," in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 3, pp. 2962–2967, IEEE, 2004.

[87] Wegner, R. and Anderson, J., "Agent-based support for balancing teleoperation and autonomy in urban search and rescue," *Int. J. Robot. Autom.*, vol. 21, no. 2, pp. 120–128, 2006.

[88] Wesselowski, K. and Fierro, R., "A dual-mode model predictive controller for robot formations," in *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 4, pp. 3615–3620, IEEE, 2004.

[89] Whetten, J., Goodrich, M., and Guo, Y., "Beyond robot fan-out: Towards multi-operator supervisory control," in *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, pp. 2008–2015, IEEE, 2010.

[90] Zhou, Y., Egerstedt, M., and Martin, C., "Hilbert space methods for control theoretic splines: A unified treatment," *Communications in Information and Systems*, vol. 6, No. 1, pp. 55–62, 2006.

[91] Zigoris, P., Siu, J., Wang, O., and Hayes, A., "Balancing automated behavior and human control in multi-agent systems: a case study in roboflag," in *American Control Conference, 2003. Proceedings of the 2003*, vol. 1, pp. 667–671, IEEE, 2003.