Optimization-based Assistive Controllers in Teleoperation of Mobile Robotic Manipulators

OPTIMIZATION-BASED ASSISTIVE CONTROLLERS IN TELEOPERATION OF MOBILE ROBOTIC MANIPULATORS

BY

SAMAN RAHNAMAEI, B.Sc.

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING
AND THE SCHOOL OF GRADUATE STUDIES
OF MCMASTER UNIVERSITY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

Master of Applied Science (2013) McMaster University
(Electrical & Computer Engineering) Hamilton, Ontario, Canada

TITLE:	Optimization-based Assistive Controllers in Teleoperation of Mobile Robotic Manipulators
AUTHOR:	Saman Rahnamaei B.Sc., (Electrical Engineering) University of Tehran, Tehran, Iran
SUPERVISOR:	Dr. S. Sirouspour
NUMBER OF PAGES:	xii, 92

To my beloved parents:

Fereshteh & M. Hossein

Abstract

This thesis investigates two significant problems in control and coordination of complex teleoperation systems as they relate to the operation of a mobile robotic manipulator. The first part of the thesis focuses on the design of a control framework to resolve kinematic redundancy in teleoperation of a mobile robotic manipulator. Apart from the redundancy, workspace considerations for the operator and robot and asymmetry of master and slave systems pose significant design challenges in such telerobotic systems . The second part of the thesis considers psychophysical aspects of teleoperation from the operator's perspective. This part presents a method for automatic *optimal* positioning of a single camera for a remotely navigated mobile robot in systems with a controllable camera platform. In each part, a constrained optimization problem is formulated and solved in real time. The solution of these optimization problems are integrated seamlessly into the teleoperation control framework in order to assist the operator in accomplishing the main task. The proposed control framework in the first part allows the operator to concentrate on the manipulation task while the mobile base and arm joint configurations are automatically *optimized* according to the needs of the task. Autonomous control subtasks are defined to guide the base and the arms towards this optimal configuration while the operator teleoperates the end-effector(s) of the mobile arm(s). The teleoperation and autonomous control tasks have adjustable relative priorities set by the system designer. The work in the second part enables the operator to focus mainly on navigation and manipulation while the camera viewpoint is automatically adjusted. The workspace and motion limits of the camera system and the location of the obstacles are taken into consideration in camera view planning. A head tracking system enables the operator to use his/her head movements as an extra control input to guide the camera placement, if and when necessary. Both proposed controllers have been implemented and evaluated in teleoperation experiments and user studies. The results of these experiments confirm the effectiveness of these controllers and demonstrate significant improvements compared to other existing controllers from the literature included in the studies.

Acknowledgements

First and foremost, I offer my sincerest gratitude to my supervisor, Dr. Shahin Sirouspour, who has supported me throughout my studies with his invaluable comments, knowledge, and patience whilst allowing me the room to work in my own way. I am deeply grateful to Dr. Tim Davidson for introducing me to the optimization topic and supporting me on the way.

My appreciation to my colleagues in the Haptics, Telerobotics and Computational Vision Lab for making this a fun and cheerful place to study. Special thanks to my dear friends Sajad Salmanipour and Behzad Iranpanah who were always willing to help and give their best suggestions.

Thanks are due to my friends in Hamilton who made the past two years enjoyable and unforgettable for me.

Very special thanks to my family for their endless love and for always believing in me and supporting me through the highs and lows of my life.

Lastly, my heartfelt appreciation goes to Atena for her kindness, help, encouragement, and for always cheering me up when I needed it.

Notation and abbreviations

SMSS	Single-Master/Single-Slave
DOM	Degrees Of Mobility
DOF	Degrees Of Freedom
HMI	Human Machine Interface
SQP	Sequential Quadratic Programming
GPS	Global Positioning System
HMD	Head Mounted Display
TCF	Teleoperation Control Frame
RFF	Robot's Fixed Frame
RPF	Robot's Pinned Frame
MF	Master's Frame
IHA	Integrated Haptic Actuator
TCT	Task Completion Time
LSD	Least Significant Difference
ANOVA	ANalysis Of VAriance
SEM	Standard Error of the Mean

Contents

List of Figures

Chapter 1

Introduction

The term teleoperation, which literally translates into *operating at a distance*, is defined by Hokayem and Spong (2006) as the means which "extends the human capability to manipulating objects remotely". Early applications of teleoperation emerged during 1950's and 1960's in space, underwater, and nuclear environments mainly because of the physical risk to the human life in those fields (Ferre *et al.*, 2007; Sheridan, 1989). Since then, experts in many other fields have used telerobotic systems in applications such as live-power line maintenance, hazardous material handling, mining, military operations, and search and rescue missions to increase the safety of human operators and to reduce their cost. Apart from safety and cost, telerobotic systems have been recently utilized to improve the capabilities and performance of humans in difficult tasks such as robotic-assisted surgery (Talamini *et al.*, 2003), nano-scale object manipulation (Sitti *et al.*, 2003), and education and training (Marín *et al.*, 2003)

Traditionally, the only feedback the operator used to get from the remote environment had been a single video feed. Over the years, researchers have added various other forms of feedback to the human interface. As for the visual interfaces, setups have been developed which use multi cameras instead of one. Furthermore, active vision systems in which the operator can control the camera have been used in many teleoperation systems. Other types of cameras such as stereo vision cameras and thermal cameras have also been utilized to give the operator additional information from the remote environment. Besides vision, audio and haptic interfaces have also been integrated to the human interface to engage more senses from the operator in the teleoperation task.

Haptic interfaces have been the subject of the majority of research in the teleoperation literature. From this point of view, teleoperation systems could be generally categorized as unilateral or bilateral. In unilateral teleoperation, the goal is for the remote robot (slave) to mimic the operator's controlling device (master) movements. In these systems, the information flow is one-directional from the master to the slave. In bilateral teleoperation, the additional goal is to reflect the forces between the slave robot and the environment back to the operator. To this end, contact position and/or force between the slave robot and the environment are measured and sent back to the master side. The master robot controller converts these information to mechanical signals and conveys them to the operator via a back-drivable master robot, also known as a *haptic interface.* The addition of the force feedback gives the operator the sensation of being present in the environment and handling the slave robot directly, which is referred to as *telepresence* in the literature. According to Hokayem and Spong (2006), the main goal of a teleoperation system is to achieve stability and telepresence. Fig. 1.1 shows a teleoperation setup with multiple haptic devices, multiple displays, and a gesture recognition device on the master side and a mobile manipulator in the remote environment. Different components of this system are going to be further discussed later in this chapter.

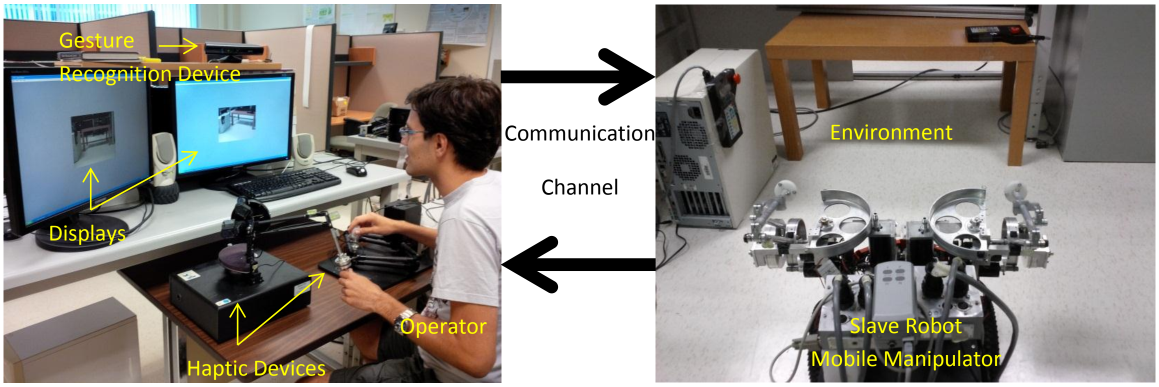


Figure 1.1: A modern teleoperation setup. Different components of the teleoperation system are marked.

1.1 Motivation

The most conventional teleoperation scenario studied in the literature is symmetric single-master/single-slave (SMSS) case in which the master and slave robots are similar to each other and have the same degrees-of-mobility (DOM). The diagram in Fig. 1.2 shows the basic elements of such system and how they interact with each other. Considering the symmetric SMSS architecture has allowed the researchers to focus on developing low-level bilateral teleoperation controls to achieve stability and transparency. The state of the art in this particular aspect of teleoperation control is quite advanced. In reality, however, most practical applications of teleoperation often involve complicated task scenarios and requirements that are beyond capabilities of symmetric SMSS. The challenging nature of some of these tasks requires complicated teleoperation architectures involving multiple master and/or slave robots (Sirouspour, 2005), dissimilar master and slave robots with kinematically redundant (Siciliano, 1990) or deficient (Romano *et al.*, 2007) slave robots.

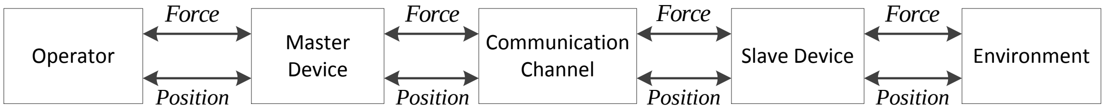
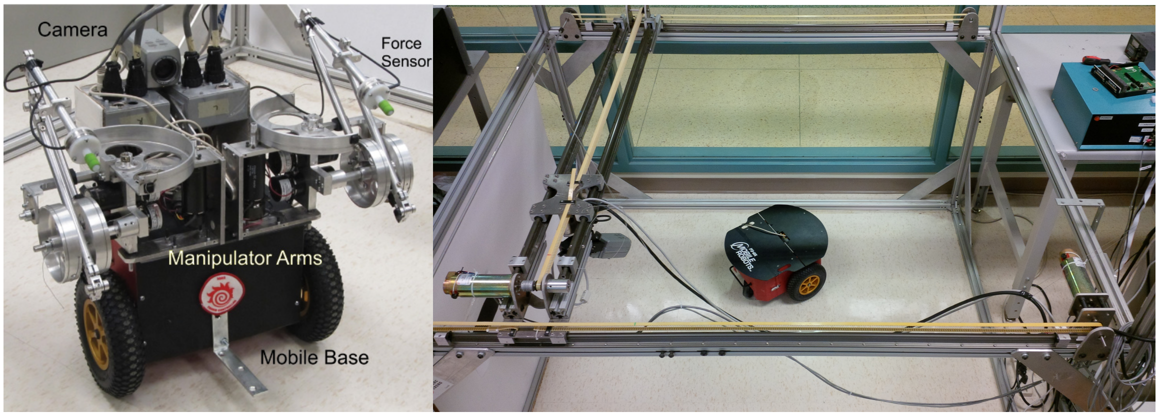


Figure 1.2: Block diagram of a symmetric SMSS teleoperation system. The master and slave blocks contain the respective device and its controller.

Apart from the individual complexities of master and slave devices, there are also other factors that contribute to the complexity of a teleoperation system as a whole. Existence of delay in the communication medium is one of the well known issues in stability of a bilateral teleoperation system (Shahdi and Sirouspour, 2009). Besides destabilizing the system, time delay also affects the operator performance as well (Corde Lane *et al.*, 2002). In a comprehensive survey of the literature, Chen *et al.* (2007) claim that there are many aspects of a teleoperation system, related to the Human Machine Interface (HMI), which could significantly affect the performance of the operator. Vision is the most important feedback the operator receives from the environment, and hence the majority of HMI design criteria are vision related. These include the field of view of the camera(s) used, positioning and control of the camera(s), depth perception from the video feed, motion of the camera, and quality of the video feed e.g. frame per second and resolution. Other issues related to HMI design include the use of audio and tactile displays, and also utilizing voice and gesture inputs.

Based on the above discussion, significant design issues for practical teleoperation systems beyond bilateral teleoperation control in symmetric SMSS can be categorized as

- Complexities arising from a large number of degrees of motion and kinematic dissimilarity/reduncy in master and slave robots e.g. see Fig. 1.3a. These lead to challenges in developing effective human-in-the-loop control strategies that involve a mixture of operator control and autonomous control.
- Providing effective visual feedback from the task environment to enhance operator's situation awareness and reduce his/her cognitive load. e.g. see Fig. 1.3b.



(a) (b)

Figure 1.3: A complex teleoperation system in McMaster University involving: (a) An 8-DOM mobile manipulator. (b) An active camera vision system.

Traditionally, the operator has been responsible for handling every aspect of teleoperation including all the complexities discussed above. In order to reduce the workload of the operator, some of these tasks could become automated so that the operator can concentrate on the more important aspects of teleoperation such as manipulation or navigation, instead of doing supplementary tasks, e.g., repositioning of camera or controlling redundant DOM of a slave robot.

1.2 Problem Statement and Thesis Contributions

Mobile manipulators are popular in many practical teleoperation applications such as nuclear/hazardous material handling, military operations, and search and rescue missions. Mounting one or two articulated arms on a mobile base vastly increase their reachable workspace, making them effective for operation in such environments. This thesis is concerned with the design of effective control and human machine interface strategies in teleoperation of mobile robotic manipulators. It specifically studies two major aspects of this challenging system design problem as briefly discussed below.

1.2.1 Semi-autonomous Control for Redundancy Resolution in Teleoperation of a Mobile Manipulator

While mounting a manipulator on a mobile base significantly increases its capabilities, it also makes its control challenging, whether using autonomous or teleoperation control. The integration of the base with the manipulator adds extra DOM creating a kinematically redundant robotic system. Teleoperation of such mobile manipulators is complicated by such redundancies and asymmetry in master/slave kinematics and workspace coverage.

Malysz (2011) has developed a general teleoperation framework which allows any arbitrary number of master and slave robots with potentially different DOM to be coordinated. The framework also involves up to three layers of autonomous control subtasks with different levels of priorities with respect to the teleoperation task; autonomous tasks can be of higher priority, weighted shared priority or lower priority. The work in this thesis employs this general framework for teleoperation of a mobile manipulator. It defines autonomous control subtasks which attempt to move the mobile base and the arm(s) to a desired *optimal* configuration. Instead of projecting the optimal configuration to the null-space of the teleoperation task, as is done previously (see Chapter 2 for a review of literature on this subject), autonomous subtasks have weighted shared priority with the teleoperation task in the proposed approach.

In formulating a constrained optimization problem, the objective is to keep the arm(s) joint positions in their mid-range and avoid singular configurations. This must be achieved while respecting the nonholonomic constraint on the motion of the base and by not affecting the position of the arm(s) end-effector(s) in a fixed global frame. The function of autonomous control subtasks is to guide the base and arms to the optimal configuration obtained from solving the optimization problem at each time step. A *dynamic* taskspace weighting matrix adjusts the relative dominance of the teleoperation and the autonomous configuration control subtasks. This yields a control strategy that at one extreme would strictly enforce the optimal configuration and at the other extreme would project the optimal configuration into the null-space of the teleoperation task. While the proposed formulation assumes a mobile base with nonholonomic motion constraint, it can be easily extended to a case without such constraint.

The optimization problem is solved in real time using sequential quadratic programming (SQP) explained in Boggs and Tolle (1995). The controller is also implemented and successfully tested on a mobile manipulator. Furthermore, human factor studies are conducted using two different camera viewpoints which will show the effectiveness of the proposed architecture and will reveal advantages and disadvantages of using each viewpoint.

1.2.2 Autonomous Viewpoint Planning in Teleoperation of a Mobile Robot

Video feed is arguably the primary source of information for the operator during a teleoperation task. The camera viewpoint can profoundly affect the operator's situation awareness and perception of the task environment. A clear occlusion-free view of the environment that provides sufficient details about the task elements can help the operator carry it out in a shorter time with greater accuracy, fewer mistakes, and reduced mental load. Yanco and Drury (2004) report that operators may spend up to 30% of their time trying to maintain situation awareness through the video feed and consequently ignore almost all other feedback channels in teleoperation. The vital role of visual feedback in enhancing situation awareness in teleoperation tasks e.g., see (Murphy and Burke, 2005) motivates the development of systematic methods for selecting camera viewpoint in teleoperation of robotic manipulators in order to facilitate task execution and reduce the operator's cognitive load.

This thesis focuses on the development of a new control strategy for camera positioning in teleoperation of mobile robots. The control approach is based on a model-predictive optimization philosophy that takes into account performance objectives, operational constraints, and user commands via natural HMI to control the operator viewpoint. The proposed approach assumes that the camera viewpoint is controllable via a separate mechanism from the mobile robot. This could be a stationary robotic arm, a mobile ground or aerial vehicle, a gantry system, or a secondary arm mounted on the mobile robot itself. Camera positioning via stationary arms and gantry systems is feasible in controlled environments such as in teleoperation for nuclear waste handling, or robotic manufacturing applications. In underwater teleoperation or in tele-robotic search and rescue, the camera position can be controlled via a second arm on the mobile robot, or by mounting it on another mobile robot/vehicle. While in this thesis a gantry system is considered for camera positioning, the methodology presented here is general and can be expanded into other types of camera positioning systems.

A constrained optimization problem is formulated and solved to predict an optimal camera configuration for a fixed time frame into future, and design a smooth trajectory for the camera to reach there in time. This optimization-based approach considers workspace and velocity limitations of the camera system in finding the desired configuration and its corresponding trajectory. It also takes into account any stationary obstacle in the workspace to avoid occlusion of the operator's view by the obstacle, both at the end of the camera's movement and in its path to the desired configuration. Moreover, head tracking inputs are integrated into the optimization problem to give the operator an intuitive way to guide the camera to a particular configuration without using his/her hands or being distracted from the main teleoperation task, if needed.

This optimization problem is also solved in real time using SQP methods. The viewpoint controller is implemented and tested on an experimental setup. The results of a teleoperation experiment and a user study comparing this approach to two other common viewpoint control strategies indicate that the proposed viewpoint controller works as expected and significantly better than a number of previous strategies from the literature.

1.3 Organization of the Thesis

The rest of the thesis is organized as follows. Chapter 2 reviews the literature on the control of mobile manipulators and kinematically redundant robots and also design of teleoperation HMI with emphasize on different camera configurations and control methods. Chapter 3 briefly reviews the control framework proposed by Malysz (2011). It provides the reader with useful background information on the control approach and the terminology, which are extensively utilized in the Chapter 4. Semi-autonomous control of mobile robots is addressed in Chapter 4. This chapter presents the optimization problem formulation and how its solution is integrated into the general control framework of Chapter 3. Experimental results and a human factor study with the proposed control strategy are discussed at the end of this chapter. Autonomous viewpoint planning is investigated in Chapter 5 where the optimization problem formulation and the results of experiment and user study with the proposed camera view planner are presented. The thesis is concluded in Chapter 6 with a discussion about possible directions for future work.

1.4 Related Publication

Rahnamaei, S. and Sirouspour, S. (2013). Automatic viewpoint planning in teleoperation of a mobile robot. Submitted for publication to *IEEE Transactions on Cybernetics.*

Chapter 2

Literature Review

This chapter is divided into three sections. The first section briefly visits the concept of using autonomous subtasks in order to reduce the complexities of a teleoperation system from the operator's perspective. The second section reviews prior work in autonomous and teleoperation control of mobile manipulators. Since a mobile manipulator is considered as a general redundant robot in this thesis, control of such robotic devices are also included in the review. The third section gives an overview of previous research in HMI design for teleoperation systems. Different viewpoint selections and control methods used in teleoperation tasks are reviewed and the advantages and disadvantages of each one are discussed. The section continues with a review of alternative HMI designs and natural control methods utilized in teleoperation systems. A few examples of studies in image stabilization systems and their applications in teleoperation systems are also provided towards the end of the last section.

2.1 Incorporating Autonomous Subtasks in Teleoperation Systems

The idea of defining autonomous subtasks in order to make teleoperation easier for the operator has been considered before. There is an abundance of studies in the literature where an autonomous subtask has been defined to control the redundant DOM of the slave robot. Some of these studies are mentioned in the next section. Beside from redundancy resolution, autonomous subtasks have been used to guarantee safety in many teleoperation systems as well as to assist the operator in many difficult tasks. Abbott *et al.* (2007) introduced two types of "virtual fixtures"; guidance virtual fixtures are used to assist the operator in moving on desired paths, and forbidden-region virtual fixtures are defined to repel the operator from certain areas of the workspace. Funda *et al.* (1996) formulated an optimization problem in order to impose constraints on some particular movements of a teleoperated surgical robot, and maintain the best possible tracking as well. Sanchez *et al.* (2002) used virtual springs to push the operator away from singular configurations of the master and slave robots in bilateral teleoperation. Another popular application of autonomous assistance in teleoperation is obstacle avoidance for mobile robots (Diolaiti and Melchiorri, 2002). Lee *et al.* (2005) conducted user studies in both virtual and real environments and showed the effectiveness of assistive controllers in reducing the number of collisions.

2.2 Control of Mobile Manipulators and Kinematically Redundant Robots

Prior work on control of mobile manipulators can be categorized as autonomous control and teleoperation control of these devices. Although the approaches for resolving redundancy of such robots have been similar in both categories, it is worthwhile to review studies from both perspectives.

2.2.1 Autonomous Motion Control

Nagatani *et al.* (2002) used two independent controllers for the base and the manipulator to control a mobile manipulator drawing large shapes on a wall. Although the controllers are independent, they have to communicate with each other in order to carry out the task successfully. In order to control a mobile manipulator with imprecise locomotion, Shin *et al.* (2003) proposed a system in which the base only moves discretely when the desired trajectory reaches out of the workspace of the manipulator. The solution provided by Arimoto *et al.* (2005) only utilizes a virtual spring and damper in the task-space to control the motion of a human-like 7 degrees of freedom (DOF) robotic arm.

Although these approaches provide a simple solution for control of mobile manipulators, they do not utilize the potentially beneficial attributes of a redundant robot. A more popular approach used by many researchers is to utilize the extra DOM to optimize some user-defined criterion by moving the robot in the null-space of the redundant kinematics. Using this approach, Khatib (1999) minimized the deviation from the mid-range joint position of the manipulator. Similarly, Shimizu *et al.* (2008)

used the extra DOM in a 7 DOM manipulator to avoid joint limits. Seraji (1998) also proposed a general framework for any kind of configuration control within the null-space of the redundant robot kinematics. The reader is also referred to Nakanishi *et al.* (2005), where the authors implement and test various existing redundancy resolution algorithms on a 7 DOF anthropomorphic robot arm under the influence of modelling errors and sensory noise. In conclusion, they suggest that simplified controllers are more robust, hence perform better when applied to complex robots where modelling errors are inevitable. In particular, they highlight the performance of a simplified variation of the acceleration-based controller provided in Hsu *et al.* (1989) as the best.

2.2.2 Teleoperation Control

In teleoperation systems, the operator usually controls the motion of one point on the slave robot, called the end effector, via the master robot. Having a kinematically redundant slave robot means that it can maintain the particular end effector position (and orientation), set by the operator, with infinite internal configurations. The challenge is to uniquely determine the joint configuration of the slave robot without asking the operator to control it directly. In teleoperation systems, similar to automatically controlled systems, redundant robots are used to increase dexterity, avoid collisions, etc.

Goel *et al.* (2003) utilized kinematically redundant manipulators in failure tolerant teleoperation; they specifically studied the case of joint “locking up” for a 3 DOF redundant manipulator. Dai and Li (2010) developed a teleoperation motion controller for an 8-DOM mobile manipulator while the end-effector’s workspace was confined by a secondary task. Buss *et al.* (2007) developed a multi-modal telepresence system which allows dexterous free space motion for the operator; this motion is then used to control the base separately. Frejek and Nokleby (2013) proposed an algorithm for simplified teleoperation of mobile manipulators by autonomously moving the base only when the manipulator becomes singular, i.e., the manipulator reaches the boundaries of its workspace.

Similar to the work in autonomous control, many researchers have preferred to utilize the null-space of the redundant robot to define an autonomous subtask. Nath *et al.* (2008) used this idea for both master and slave redundant robots to achieve predefined subtask objectives. Hishinuma and Nenchev (2006) imposed an additional constraint on a redundant manipulator mounted on a flexible base in order to suppress the base vibrations. Rubi *et al.* (2002) autonomously controlled motion in the null-space of a redundant manipulator to avoid singularities and to keep the arm in the most manipulable configuration. The same idea was used by Nanayakkara *et al.* (2001) to prevent collisions in teleoperation of a 7-DOM industrial arm. Hwang and Hannaford (1998) utilized the two redundant DOFs in a 5-DOF arm to avoid joint limits during teleoperation. Park and Khatib (2006) exploited the extra DOM in mobile manipulators to autonomously control the internal posture of the robot. Malysz and Sirouspour (2011a) considered a dual-master trilateral configuration to resolve the kinematic redundancy of a single slave teleoperation system. Instead of prioritizing one master device over the other, this work focuses on the cooperation of the master devices with shared priorities. The method is further generalized in Malysz (2011), where any arbitrary number of master and slave robots and autonomous subtasks with different priorities can be incorporated in the control framework. This work will be further discussed in Chapter 3 as it provides the foundation for this thesis.

2.3 HMI Design and Viewpoint Control

The literature review on HMI design is divided into three subsections. Each of these subsections presents one critical aspect of HMI design for teleoperation systems that could heavily influence the operator's ability to perform the desired task.

2.3.1 Camera Positioning and Viewpoint Selection

As mentioned in the previous chapter, one of the most important aspects of HMI for a teleoperation system is the camera viewpoint selection. Freedman *et al.* (1977) and Das *et al.* (1989) were among the first researchers to study and highlight the effect of viewpoint selection on the performance of the operator. A search of the literature reveals various strategies for camera control including fixed and controllable egocentric, overhead view, exocentric, and also combinations of these viewpoints. Previous studies have compared and contrasted egocentric and exocentric viewpoints (Olmos *et al.*, 2000; Hollands and Wickens, 1999; Chen *et al.*, 2007). The emerging consensus from this past research is that the egocentric viewpoint helps with local navigation whereas an exocentric viewpoint usually provides the operator with greater global situation awareness. These characteristics have led to the concept of tethered viewpoint in which the camera moves with the robot, but instead of being fixed on the body, it is placed behind and above the robot.

Hollands and Lamb (2011) compared egocentric, tethered, and exocentric viewpoints in navigation and spatial awareness within a virtual reality environment. They concluded that the tethered view, while not the best in each individual metric, yielded the best overall trade-off by offering combined advantages of the other two approaches. In order to implement the idea of tethered viewpoint in a practical setting, Ricks *et al.* (2004) fused information from camera images, laser range finder and sonar readings into a single video feed to provide a tethered-like display. Keyes *et al.* (2006) argued that using an overhead camera which includes the chassis of the robot in the video feed can enhance the situation awareness. Essentially, an overhead camera is an alternative method to apply tethered viewpoints to teleoperation systems without using video processing to enhance the video.

Alternatively, researchers working on robotics search and rescue systems have attempted to overcome the shortcomings of an egocentric viewpoint by allowing the operator to freely turn the camera (Scholtz *et al.*, 2004). Hughes and Lewis (2005) showed the benefits of using two views, one from a camera fixed on the robot and another from a camera controlled by the operator, based on a study in a virtual reality environment.

Most other approaches in controlling the viewpoint autonomously are heavily reliant on the model of the environment which effectively makes them unusable in real life teleoperation situations. Brooks and McKee (2001) describe an architecture for autonomous camera placement in a teleoperation task, but it relies on a geometric model of the task-space, the camera positioning system, and the task itself. Furthermore, there are many autonomous tracking camera controllers aimed at working in structured environments such as classrooms for the purpose of automated lecture capturing, e.g., see Rui *et al.* (2004) and Mukhopadhyay and Smith (1999).

2.3.2 Alternative Designs and Natural Control Methods

A user-controlled camera may help the operator explore the task environment with more flexibility. However, it would also burden him/her with extra cognitive and manipulation load, increasing the probability of errors and mistakes. Many teleoperation tasks require use of both arms, which means that the operator would have to pause the main task and switch to a different hand-controller to adjust the viewpoint. Even in cases that the main task could be carried out with one arm, the operator would likely have to stop it and focus on the camera positioning task. In order to address this problem, Cândido *et al.* (2008) proposed a shared control system for a pan-tilt camera mounted on a mobile robot. In their control framework, the operator manually sets the desired azimuth and elevation of the camera or the GPS location of the point of interest, and the camera autonomously maintains the desired direction or fixates on the point of interest.

Natural human-machine interfaces such as head (face) and gaze trackers allow for hand-free control of camera viewpoint. Face tracking is already being used in video games (Wang *et al.*, 2006) and is set to become a turning point in the gaming industry (Laviola, 2008). There are also examples of applications of such interfaces in robotic teleoperation. For instance, Martins and Ventura (2009) utilized a pair of stereo cameras along with a head mounted display (HMD) and a head tracker to let the operator turn a robot by rotating his/her head. Fournier *et al.* (2011) developed a virtual reality immersion champer called CAVE in which the operator can interact with a 3D avatar of the robot; head tracking is employed in the CAVE to control the viewpoint of the operator.

Another alternative control method considered for teleoperation is eye tracking. Mitsugami *et al.* (2005) proposed a system in which the operator gazes at the robot to select it, then gazes at its desired destination in the task-space to move the robot. Eye tracking has also been considered as computer user interface, especially for the physically disabled (Kaufman *et al.*, 1993). Zhu *et al.* (2011) used a combined gaze and head tracking system and compared it to manual and autonomous control for the camera viewpoint in an experimental setup mimicking a rock breaker. Their results revealed performance advantages for the natural interface compared to manual and autonomous control.

2.3.3 Camera Motion and Image Stabilization

As stated in Chapter 1, one of the other issues related to the video feedback which could affect the performance of the operator is the motion of the camera. Rapid movements of the camera can induce vibration in the video feed which is known to cause motion sickness for the operator (Shiroma *et al.*, 2009; Hayashi *et al.*, 2005). Cancelling these vibrations through image stabilization has been a subject of research in the computer vision and teleoperation research communities for long. There are several examples in the literature where researchers have attempted to smooth out unwanted vibrations of the camera images in moving vehicles (Jin *et al.*, 2000; Hsu *et al.*, 2007). Shiroma *et al.* (2009) developed a compact image stabilization system to be used in teleoperation of small-sized humanoid robots. Hayashi *et al.* (2005) proposed an image stabilization system for search and rescue robots in order to reduce the operator's fatigue and possibility of motion sickness. These studies emphasize the importance of a smooth video feedback in a teleoperation system, which is going to be further discussed in Chapter 5.

Chapter 3

Summary of the General Teleoperation Control Framework

3.1 Overview

The general teleoperation control method of Malysz (2011) is briefly summarized in this chapter. A conceptual block diagram of this hierarchical control strategy is given in Fig. 3.1. This teleoperation control framework consists of four sets of high-level teleoperation-specific subtasks on top of low-level joint velocity controllers for each robot to ensure velocity tracking. The first set of high-level subtask(s) are related to human-in-the-loop teleoperation control. As mentioned in the previous chapter, any arbitrary number of master and slave robots can be incorporated in this control framework. To this end, each master device is assigned to control one *teleoperation control frame* (TCF). A TCF may involve all or a subset of the DOF of a slave robot. The other high-level subtasks are three sets of velocity-based autonomously controlled subtasks with different levels of priorities with respect to the teleoperation task. The *prioritized autonomous control* subtask(s) has the highest priority in the control hierarchy and the rest of the control subtasks are projected to its null-space. The second set of subtasks, denoted as *concatenated autonomous control*, usually have the same priority as the teleoperation subtask, but their relative strength can be adjusted via a dynamic weighting matrix. The last one is null-space autonomous control subtask which has the least priority and is designed to control any remaining null-space velocities.

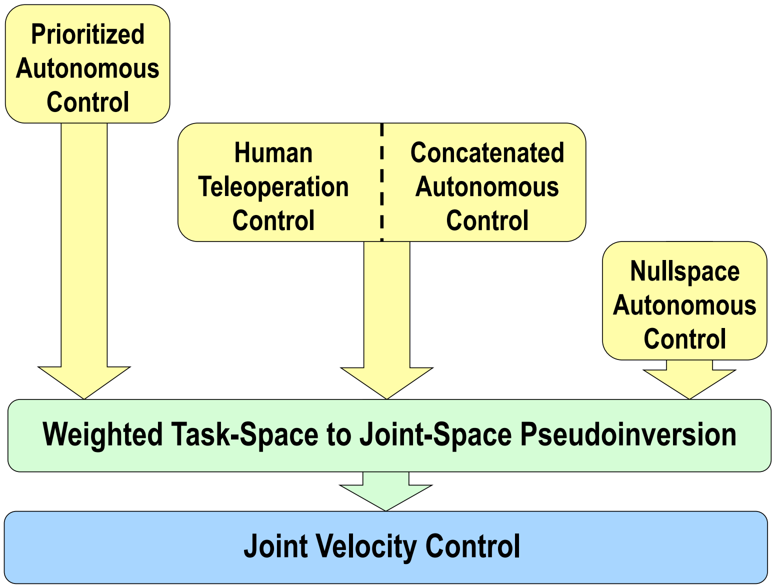

Prioritized Autonomous Control
Human Teleoperation Control
Concatenated Autonomous Control
Nullspace Autonomous Control
Weighted Task-Space to Joint-Space Pseudoinversion
Joint Velocity Control

Figure 3.1: Conceptual overview of the hierarchical teleoperation control framework proposed in Malysz (2011). The yellow boxes represent the high-level teleoperation subtasks, and their height indicate their respective control command priority in the teleoperation framework. (Malysz and Sirouspour, 2013)

Throughout this chapter, $\dot{x}_m^d$ and $\dot{x}_s^d$ denote the teleoperation control commands for the master and slave robots, respectively. The concatenated autonomous control command is represented with $\dot{c}^d$, and the other two commands are not going to be discussed any further since they are not utilized in this thesis.

3.2 Summary of the Control Design

This section briefly defines all the control commands in the teleoperation control framework which are used later in Chapter 4. Note that two sets of autonomous control commands are omitted from the framework since they are not used in this thesis. Hence, the following presentation is a simplified version of the teleoperation control framework in Malysz (2011).

The master and slave robot dynamics could be represented in the following nonlinear form (Sciavicco and Siciliano, 2000):

$$M_\gamma(q_\gamma)\ddot{q}_\gamma + C_\gamma(q_\gamma, \dot{q}_\gamma)\dot{q}_\gamma + D_\gamma\dot{q}_\gamma + g_\gamma(q_\gamma) = \tau_\gamma + J_\gamma^T(q_\gamma)f_{e,\gamma} \tag{3.1}$$

where γ stands for "is", $i \in [1, N_s]$ or "jm", $j \in [1, N_m]$; "is" and "jm" represent the i^{th} slave and j^{th} master robot, respectively. Moreover, N_m and N_s are the total number of master and slave robots, respectively. q_γ is the vector of the joint variables, τ_γ is the joint torques vector, $f_{e,\gamma}$ is the external user/environment force vector, and $J_\gamma(q_\gamma)$ is a configuration-dependent Jacobian matrix mapping workspace forces to joint-space torques. Furthermore, $M_\gamma(q_\gamma)$ and D_γ are positive definite mass and damping matrices, $C_\gamma(q_\gamma, \dot{q}_\gamma)$ represents the centrifugal and Coriolis effects, and $g_\gamma(q_\gamma)$ contains position-dependent forces, e.g., gravity.

The master and slave task-space velocity signals as well as the concatenated autonomous subtask (hereafter referred to as the autonomous subtask) velocity-like signals are defined below:

$$\dot{\bar{q}}_m = \bar{J}_m^\dagger \dot{\bar{x}}_m \tag{3.2}$$

$$\dot{\bar{q}}_m \triangleq [\dot{q}_{1m}^T \dot{q}_{2m}^T \cdots \dot{q}_{N_m m}^T]^T, \quad \bar{J}_m \triangleq [J_{1m}^T J_{2m}^T \cdots J_{N_m m}^T]^T, \quad \dot{\bar{x}}_m \triangleq [\dot{x}_{1m}^T \dot{x}_{2m}^T \cdots \dot{x}_{N_m m}^T]^T$$

$$\dot{x}_{is} = J_{is}\dot{q}_s, \qquad i \in [1, N_s] \tag{3.3}$$

$$\dot{c}_j = J_{jc}\dot{q}_s, \qquad j \in [1, N_c] \tag{3.4}$$

where N_c is the total number of autonomous subtasks. J_{is}, J_{jc}, and J_{km} $(k \in [1, N_m])$ are all Jacobian matrices which map velocities from joint-space to workspace. Furthermore, the dagger † superscript indicates the generalized pseudo-inverse, defined below for full rank matrices:

$$J^\dagger \triangleq \begin{cases} J^{-1}, & W_x J W_q \text{ square and full rank} \\ W_q^2 J^T (J W_q^2 J^T)^{-1}, & W_x J W_q \text{ fat and full rank} \\ (J^T W_x^2 J)^{-1} J^T W_x^2, & W_x J W_q \text{ tall and full rank} \end{cases} \tag{3.5}$$

In the above equations, $J \in \Re^{n_x \times n_q}$ is a Jacobian matrix. $W_q \in \Re^{n_q \times n_q}$ and $W_x \in \Re^{n_x \times n_x}$ are symmetric positive semi-definite weighting matrices in the joint-space and task-space, respectively. Defining the following concatenated task-space vector allows mixed autonomous/human teleoperation:

$$\dot{\bar{x}}_{sc} = \begin{pmatrix} \dot{\bar{x}}_s \\ \dot{\bar{c}} \end{pmatrix} = \begin{pmatrix} \bar{J}_s \\ \bar{J}_c \end{pmatrix} \dot{\bar{q}}_s = \bar{J}_{sc}\dot{\bar{q}}_s \tag{3.6}$$

$$\dot{\bar{c}} \triangleq [\dot{c}_1^T \dot{c}_2^T \cdots \dot{c}_{N_c}^T]^T, \quad \bar{J}_c \triangleq [J_{1c}^T J_{2c}^T \cdots J_{N_c c}^T]^T$$

$$\dot{\bar{x}}_s \triangleq [\dot{x}_{1s}^T \dot{x}_{2s}^T \cdots \dot{x}_{N_s s}^T]^T, \quad \bar{J}_s \triangleq [J_{1s}^T J_{2s}^T \cdots J_{N_s s}^T]^T, \quad \dot{\bar{q}}_s \triangleq [\dot{q}_{1s}^T \dot{q}_{2s}^T \cdots \dot{q}_{N_s s}^T]^T$$

The joint-space reference signals are defined based on the task-space desired velocities below:

$$\dot{\bar{q}}_m^d \triangleq \bar{J}_m^\dagger \dot{\bar{x}}_m^d \tag{3.7}$$

$$\dot{\bar{q}}_s^d \triangleq \bar{J}_{sc}^\dagger \dot{\bar{x}}_{sc}^d \tag{3.8}$$

where the superscript d denotes a desired reference signal.

The task-space reference signals are designed as follows:

$$\begin{aligned} \dot{x}_{is}^d \triangleq & A_i(K_{ih}\tilde{f}_{im} + K_{if}\tilde{f}_{is}) + \Lambda_i(K_{ip}x_{im} - x_{is}) + \\ & (I - \Omega_i)((I - \Sigma_i)\dot{\tilde{x}}_{is} + \Sigma_i K_{ip}\dot{\tilde{x}}_{im}) \end{aligned} \tag{3.9}$$

$$\begin{aligned} \dot{\bar{x}}_m^d = & [\dot{x}_{1m}^{d^T} \dot{x}_{2m}^{d^T} \cdots \dot{x}_{N_x m}^{d^T}]^T \\ \triangleq & \bar{K}_p^{-1}\bar{P}_{u_{xx}}[\alpha_1^T \alpha_2^T \cdots \alpha_{N_x}^T]^T + \bar{K}_p^{-1}\bar{P}_{u_{xc}}\dot{\bar{c}}^d + \bar{K}_p^{-1}\bar{P}_{n_{xx}}[\xi_1^T \xi_2^T \cdots \xi_{N_x}^T]^T \end{aligned} \tag{3.10}$$

where

$$
\begin{aligned}
\bar{J}_{sc}\bar{J}_{sc}^{\dagger} \triangleq& \bar{P}_u = \begin{bmatrix} \bar{P}_{u_{xx}} & \bar{P}_{u_{xc}} \\ \bar{P}_{u_{cx}} & \bar{P}_{u_{cc}} \end{bmatrix} && (3.11) \\
I - \bar{P}_u \triangleq& \bar{P}_n = \begin{bmatrix} \bar{P}_{n_{xx}} & \bar{P}_{n_{xc}} \\ \bar{P}_{n_{cx}} & \bar{P}_{n_{cc}} \end{bmatrix} && (3.12) \\
\bar{K}_p =& \mathrm{diag}(K_{1p}I_{n_{1x}}, K_{2p}I_{n_{2x}}, \cdots, K_{N_xp}I_{n_{N_x}}) && (3.13) \\
\alpha_i \triangleq& A_i(K_{ih}\tilde{f}_{im} + K_{if}\tilde{f}_{is}) + \Lambda_i(x_{is} - K_{ip}x_{im}) \\
&+ (I - \Omega_i)(\Sigma_i\dot{\tilde{x}}_{is} + (I - \Sigma_i)K_{ip}\dot{\tilde{x}}_{im}) && (3.14) \\
\xi_i \triangleq& (I - 2\Sigma_i)(I - \Omega_i)(K_{ip}\dot{\tilde{x}}_{im} - \dot{\tilde{x}}_{is}) + 2\Lambda_i(x_{is} - K_{ip}x_{im}) && (3.15) \\
\tilde{\chi}(s) =& \frac{C}{s + C}\chi(s) && (3.16)
\end{aligned}
$$

In the above equations, $i \in [1 : N_x]$; N_x is the total number of TCFs. Λ_i, Ω_i, Σ_i are positive diagonal matrices and A_i, K_{ip}, K_{if}, K_{ih} are scalar motion/force gains. Moreover, the *tilde* filter is defined in the frequency domain according to (3.16). Note that the above formulation concerns high-level velocity commands. There is also a low-level adaptive joint velocity controller in place to enforce velocity tracking (Malysz, 2011).

The block diagram in Fig. 3.2 is provided in the original work (Malysz, 2011) for the symmetric SMSS system with concatenated autonomous control subtasks. In the case of a kinematically redundant slave robot, the only difference is that $\bar{J}_s$ will be a fat matrix instead of a square matrix, which does not change the overall diagram. As illustrated in the diagram, first, the high-level teleoperation controller defines the task-space velocity commands, $\dot{x}_m^d$ and $\dot{x}_s^d$, in Eqs (3.9) and (3.10). Next, the autonomous control command, $\dot{c}^d$, is concatenated to $\dot{x}_s^d$ to form $\dot{x}_{sc}^d$ which enables the desired mixed autonomous/human teleoperation. Using the generalized pseudo-inverse of the Jacobian in Eqs. (3.7) and (3.8), these commands are converted to joint-space velocity commands. Note that the task-space weighting matrix, $\bar{W}_{xc}$, affects $\bar{J}_{sc}^{\dagger}$ and consequently $\bar{P}_u$ and $\bar{P}_n$ (Malysz, 2011; Malysz and Sirouspour, 2011b). Finally, the joint-space velocity commands are fed into the low-level joint velocity controllers to ensure velocity tracking.

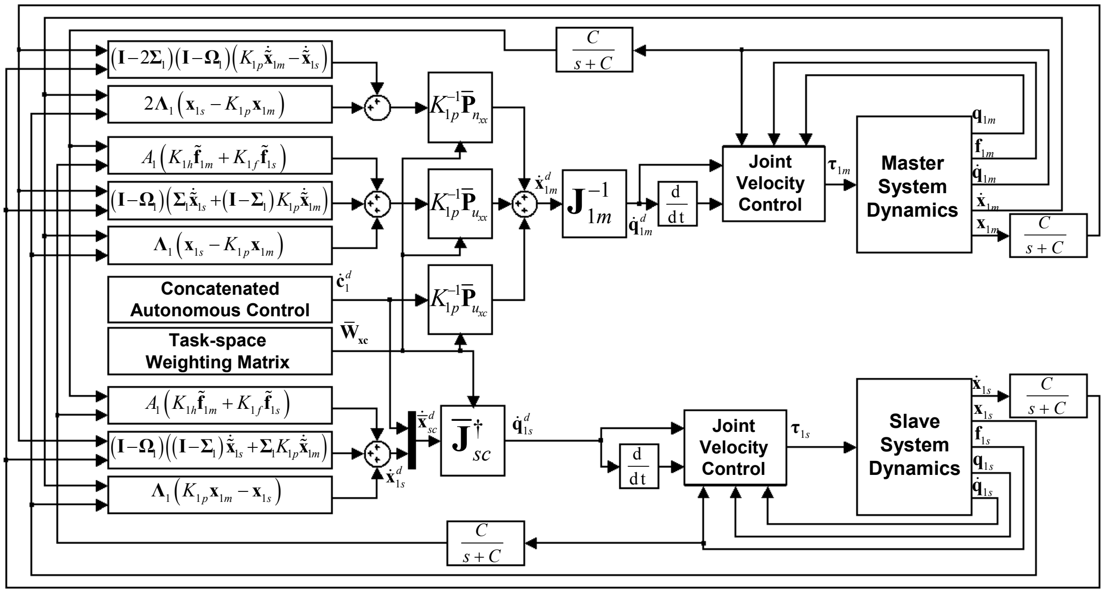

$(\mathbf{I}-2\mathbf{\Sigma}_1)(\mathbf{I}-\mathbf{\Omega}_1)(K_{1p}\dot{\tilde{\mathbf{x}}}_{1m}-\dot{\tilde{\mathbf{x}}}_{1s})$
$2\mathbf{\Lambda}_1(\mathbf{x}_{1s}-K_{1p}\mathbf{x}_{1m})$
$A_1(K_{1h}\tilde{\mathbf{f}}_{1m}+K_{1f}\tilde{\mathbf{f}}_{1s})$
$(\mathbf{I}-\mathbf{\Omega}_1)(\mathbf{\Sigma}_1\dot{\tilde{\mathbf{x}}}_{1s}+(\mathbf{I}-\mathbf{\Sigma}_1)K_{1p}\dot{\tilde{\mathbf{x}}}_{1m})$
$\mathbf{\Lambda}_1(\mathbf{x}_{1s}-K_{1p}\mathbf{x}_{1m})$
Concatenated Autonomous Control
$\dot{\mathbf{c}}_1^d$
Task-space Weighting Matrix
$\bar{\mathbf{W}}_{xc}$
$A_1(K_{1h}\tilde{\mathbf{f}}_{1m}+K_{1f}\tilde{\mathbf{f}}_{1s})$
$(\mathbf{I}-\mathbf{\Omega}_1)((\mathbf{I}-\mathbf{\Sigma}_1)\dot{\tilde{\mathbf{x}}}_{1s}+\mathbf{\Sigma}_1K_{1p}\dot{\tilde{\mathbf{x}}}_{1m})$
$\mathbf{\Lambda}_1(K_{1p}\mathbf{x}_{1m}-\mathbf{x}_{1s})$
$K_{1p}^{-1}\bar{\mathbf{P}}_{n_{xx}}$
$K_{1p}^{-1}\bar{\mathbf{P}}_{u_{xx}}$
$K_{1p}^{-1}\bar{\mathbf{P}}_{u_{xc}}$
$\frac{C}{s+C}$
$\dot{\mathbf{x}}_{1m}^d$
$\mathbf{J}_{1m}^{-1}$
$\dot{\mathbf{q}}_{1m}^d$
$\frac{d}{dt}$
Joint Velocity Control
$\boldsymbol{\tau}_{1m}$
Master System Dynamics
$\mathbf{q}_{1m}$
$\mathbf{f}_{1m}$
$\dot{\mathbf{q}}_{1m}$
$\dot{\mathbf{x}}_{1m}$
$\mathbf{x}_{1m}$
$\frac{C}{s+C}$
$\dot{\bar{\mathbf{x}}}_{sc}^d$
$\dot{\mathbf{x}}_{1s}^d$
$\bar{\mathbf{J}}_{sc}^{\dagger}$
$\dot{\mathbf{q}}_{1s}^d$
$\frac{d}{dt}$
Joint Velocity Control
$\boldsymbol{\tau}_{1s}$
Slave System Dynamics
$\dot{\mathbf{x}}_{1s}$
$\mathbf{x}_{1s}$
$\mathbf{f}_{1s}$
$\mathbf{q}_{1s}$
$\dot{\mathbf{q}}_{1s}$
$\frac{C}{s+C}$
$\frac{C}{s+C}$

Figure 3.2: Block diagram of the general teleoperation control framework for the simplified case of SMSS without prioritized and null-space autonomous subtasks. (Malysz, 2011)

Using the above high-level teleoperation joint velocity commands combined with suitable low-level joint velocity controllers, e.g., the adaptive controller introduced in Malysz (2011), it can be shown that the haptic forces conveyed to the operator satisfy:

$$K_{ih}\tilde{f}_{im} + K_{if}\tilde{f}_{is} + f_c = M_{it}\ddot{x}_{im} + B_{it}\dot{x}_{im} \tag{3.17}$$

where

$$f_c = A_i^{-1}\bar{P}_{u_{xx}}^{-1}\bar{P}_{u_{xc}}\dot{c}^d \tag{3.18}$$

$$M_{it} = (I + \bar{P}_{u_{xx}}^{-1}\bar{P}_{n_{xx}})A_i^{-1}C^{-1}K_{ip} \tag{3.19}$$

$$B_{it} = (\Omega_i + \bar{P}_{u_{xx}}^{-1}\bar{P}_{n_{xx}})A_i^{-1}K_{ip} \tag{3.20}$$

In (3.17), $i \in [1 : N_x]$, f_{im} and f_{is} are the user and environment forces. f_c is the force arising from one autonomous control subtask. Moreover, M_{it} and B_{it} are the virtual tool mass and damper matrices. Note that since the autonomous subtask has the same priority with the teleoperation commands, the user can feel its force and interact with it.

In order to utilize the autonomous control capabilities of this framework, one only needs to define proper velocity-like reference signals, $\dot{c}_j^d$, and their corresponding Jacobian, J_{jc}, according to equation (3.4). The present work defines these signals in Eqs. (4.12) and (4.14) based on the solution to an optimization problem.

Chapter 4

Semi-autonomous Control of a Kinematically Redundant Mobile Manipulator

In this chapter, a new controller for assistive teleoperation of a kinematically redundant mobile manipulator is presented. While the combined mobile base/slave arm(s) system is kinematically redundant, the slave arm itself is non-redundant and kinematically similar to the master arm. Our goal is to develop a controller that enables the operator to focus on symmetric SMSS teleoperation of the manipulator arm, while the base automatically repositions itself according to the needs of the teleoperation task. To this end, first a constrained optimization problem is formulated. The solution to this problem determines an optimal configuration for the mobile manipulator in the workspace for a given manipulator end-effector position(s), controlled by the operator through teleoperation. Next, the solution of this problem is used to define two sets of concatenated autonomous subtask reference commands and Jacobian matrices: $\dot{c}_b^d$,

J_{c_b}, $\dot{c}^d_{ff}$, and $J_{c_{ff}}$ (see Eq. (3.4)). Note that these autonomous subtasks have the same priority as the teleoperation control commands (see Fig.3.1). Finally, the autonomous subtasks are integrated into the general teleoperation control framework of Chapter 3 to form the overall controller. Experimental results with the proposed semi-autonomous teleoperation controller are also given in the last section of this chapter.

4.1 Formulation of an Optimization Problem for Redundancy Resolution

The problem formulation is based on the generic schematic in Fig. 4.1. In this figure:

- MF and WF are two coordinate frames fixed with respect to the master device and the workspace, respectively.
- RFF is the Robot's Fixed Frame. This coordinate frame is fixed on the mobile robot and is used to represent variables when the operator is using the egocentric viewpoint.
- RPF is the Robot's Pinned Frame. The origin of this coordinate frame is fixed on the mobile robot, but its orientation is always aligned with WF. This frame is used to represent variables when the operator is using the bird's eye view camera.
- T_f is the time in which the robot is expected to reach the optimal configuration.
- P_{ee} is the manipulator's end effector position. P_{b_0} and P_b are the current and optimal desired position of the base, respectively.

- Δx, Δy, and $\Delta\theta$ are relative displacements and rotation of the base from 0 to T_f with respect to RFF at time 0.

The mobile manipulator depicted in the schematic of Fig. 4.1 has a nonholonomic mobile base and two articulated arms mounted on top of it. Note that this is simply one example of a mobile manipulator and the scope of this work is not restricted to this example. In this figure, the master side also consists of two arms kinematically similar to the ones on the mobile robot. Hence, the operator can easily teleoperate the two arms on the mobile robot (without the base) because of the symmetric nature of the master/slave configuration. Our goal is to reposition the base automatically so that the operator can teleoperate the arms without getting involved in the redundancy resolution of the whole mobile manipulator (including the base). Another challenging aspect of the problem is that the workspace of the master device is limited but the workspace of the mobile manipulator is potentially infinitely large. This is addressed later in this chapter.

Fig. 4.1 depicts a scenario where the operator has stretched the arms of the mobile manipulator near the boundaries of their workspace. In such case, the mobile base has to move in order to enlarge the workspace of the arms; the dashed blue shape in Fig. 4.1 represents a possible optimal configuration for the mobile manipulator in this scenario. Once the optimal configuration is determined by the optimization problem, autonomous subtasks in the form of (3.4) are defined and incorporated into the general teleoperation control framework to move the base autonomously and to guide the arms towards the optimal configuration. Finally, the high-level joint-space velocity commands are produced using Eqs. (3.7) and (3.8).

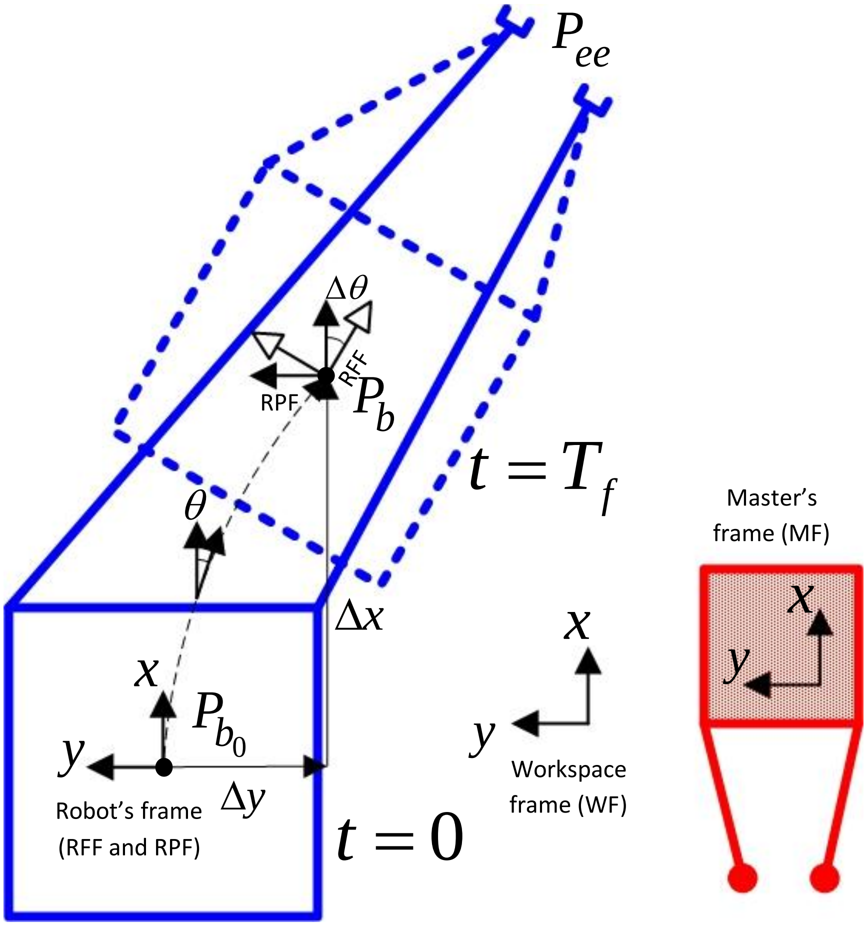


Figure 4.1: Top view of the robot and the master device. The solid and the dashed blue lines represent a possible initial and optimal configurations, respectively. Variables Δx, Δy, $\Delta\theta$, and θ are shown in solid black while the path of the robot is demonstrated in dashed black. The master device is also shown in red. The robot's, master's, and workspace frames are shown in their respective positions. Note how RFF and RPF are always aligned with the robot and the workspace frame, respectively.

The mobile manipulator's desired configuration is determined by solving the optimization problem presented below. For certain mobile manipulators some of the constraints might be redundant or need minor adjustments.

$$\min_{q_i,\Delta x,\Delta y,\Delta\theta} \quad \frac{1}{n}\sum_{i=1}^{n} k_i \frac{q_i-\bar{q}_i}{U_i-L_i} \tag{4.1}$$

subject to

$$L_i \leq q_i \leq U_i \tag{4.2}$$

$$R^{WF}_{R*F} P^{R*F}_{ee}(q) + P^{WF}_{b} = P^{WF}_{ee} \in \Re^3 \tag{4.3}$$

$$O(q) = O_0 \in \Re^3 \tag{4.4}$$

$$|\det J(q)| \geq M, \quad J(q) \in \Re^{n\times n} \tag{4.5}$$

$$\Delta x \sin\Delta\theta = \Delta y(1+\cos\Delta\theta) \tag{4.6}$$

The objective function (4.1) is chosen to move the arm joints as far as possible from their limits; here q_i is the i^{th} joint angle, $\bar{q}_i$, U_i, and L_i are the midpoint, maximum, and minimum value of that joint angle, respectively. Moreover, k_i is the relative weight of the joint angle q_i in the cost function and n is the total number of joints whether there is one or more than one arm mounted on the mobile base.

The first constraint in (4.2) expresses the joint limitations. An important aspect of autonomous subtask design is not to interfere with the operator teleoperation commands. As a result, the constraints in (4.3) and (4.4) require that the mobile robot end-effector position/orientation(s) in the workspace not change due to internal repositioning of its base and joints. In (4.3), $R * F$ stands for either RFF or RPF. The values with superscript $R * F$ and WF are represented in the robot's coordinate frame and the workspace coordinate frame, respectively (see Fig. 4.1). Furthermore,

R_{R*F}^{WF} is the rotation matrix between the workspace and robot coordinate frames. The constraint in (4.4) is only applicable to the cases where the operator explicitly controls the orientation of the end effector, as well as its position. $O(q)$ is the orientation of the end effector in the desired configuration as a function of joint angles and O_0 is the current orientation of the end effector. These vectors are consist of a set of three "Euler angles" which describe the orientation of the end effector using a sequence of three elementary rotations (about x, y, or z axes) from the workspace frame.

The constraint in (4.5) is to ensure the robot arm(s) never ends up in a singular configuration; here $J(q)$ is the Jacobian matrix of the arm defined below, and M is a predefined threshold.

$$\dot{x}_{ee} = J(q)\dot{q} \tag{4.7}$$

where $q \in \Re^n$ is the vector of manipulator joint variables and $\dot{x}_{ee} \in \Re^n$ is the task-space velocity vector of the manipulator's end-effector. The Jacobian matrix maps velocities from joint-space to workspace. If $\det J(q) = 0$ the manipulator would become singular which means it loses the ability to move in certain directions (Sciavicco and Siciliano, 2000). Note that for mobile manipulators with more than one arm, constraints (4.3), (4.4), and (4.5) have to be written for each arm separately.

The last constraint in (4.6) is for mobile manipulators with a nonholonomic base robot. In such case, the three base motion variables Δx, Δy, and $\Delta\theta$, defined in Fig. 4.1, are not independent of each other. Assuming constant base translational and angular velocities until the base reaches its optimal position, one could show:

$$\Delta x = \int_0^{T_f} v \cos\theta dt = \int_0^{\Delta\theta} \frac{v}{\omega} \cos\theta d\theta = \frac{v}{\omega} \sin\Delta\theta \tag{4.8}$$

$$\Delta y = \int_0^{T_f} v \sin\theta dt = \int_0^{\Delta\theta} \frac{v}{\omega} \sin\theta d\theta = \frac{v}{\omega}(1 - \cos\Delta\theta) \tag{4.9}$$

where T_f is the time it takes for the robot to get to the optimal position and θ is the angle between the robot and the initial heading direction of the robot i.e. initial x direction of RFF. A simple interpretation of (4.8) and (4.9) is that the robot moves on a circular path segment to reach its optimal configuration. Using these two equations one could come up with the constraint in (4.6).

In most cases, this formulation will result in a nonconvex nonlinear optimization problem. However since the problem only involves a few variables it can be easily solved using SQP in real time. Note that the above formulation will almost always result in one unique position for the base. In the case of a nonholonomic base, there are three variables associated with the base while there are two equality constraints for every arm in (4.3) (for x and y directions) and one nonholonomic constraint in (4.6). So even with just one arm, there are enough equality constraints to uniquely identify the base optimal position.

4.2 Autonomous Subtask Design

In this section, the reference signals in the control architecture described in Chapter 3 are determined. On the master side, the master arm(s) joint space variables are $\dot{q}_m$. On the slave side, $\dot{q}_s$ includes the slave arm joint variables and the base motion variables (in case of a nonholonomic robot v and ω). It is clear that the slave mobile manipulator is kinematically redundant. Two autonomous subtasks with shared priority with the teleoperation task, introduced in Eq. (3.4), are designed. One deals with the redundant DOM and controls the motion of the base and the other works in parallel to the teleoperation and guides the manipulator (and in turn the operator) to the optimal configuration.

The teleoperation reference commands are designed in Malysz (2011) with $\dot{x}_m^d$ from Eq. (3.10) being represented in the master's coordinate frame and $\dot{x}_s^d$ from Eq. (3.9) in RFF if using egocentric camera, or RPF if using bird's eye view camera (see Fig. 4.1). Using the robot attached frames solves the issue with the bounded and unbounded workspace of the master and slave robots. The workspace of the slave arm(s) becomes bounded in these coordinate frames which enables symmetric SMSS teleoperation between master and slave arm(s). An alternative interpretation of this coordinate frame selection is that the master robot is being moved with the mobile robot, which effectively makes its workspace unbounded.

One autonomous subtask needs to be defined to resolve the redundancy of the mobile manipulator and control the position of the base. Keeping in mind that the reference command should be defined such that the base reaches the optimal position in T_f seconds, the following controller is used to define the desired base velocities:

$$\omega^d = \frac{\Delta\theta}{T_f} \tag{4.10}$$

$$v^d = \begin{cases} \omega^d \frac{\Delta x}{sin\Delta\theta} & \text{if } \Delta\theta \neq 0 \\ \frac{\Delta x}{T_f} & \text{if } \Delta\theta = 0 \end{cases} \tag{4.11}$$

Note that (4.11) is obtained by combining (4.8) and (4.10). Using (4.10) and (4.11), the following autonomous subtask reference command, $\dot{c}_b$, is designed for the control of the mobile base:

$$\dot{c}_b^d = \begin{pmatrix} v^d \\ \omega^d \end{pmatrix} \tag{4.12}$$

$$J_{c_b} = \begin{bmatrix} 0_{N_b \times N_s} & I_{N_b} \end{bmatrix} \tag{4.13}$$

where N_s and N_b represent the total DOM of the slave manipulator and the base, respectively.

Another autonomous subtask, $\dot{c}_{\text{ff}}$, is defined to give force feedback to operator and guide him/her to the optimal arm configuration. Note that the force feedback should not be too strong to interfere with the operator teleoperation task. This subtask has shared weighted priority with the teleoperation task and its relative strength can be controlled with the weighting matrix W_c (Malysz and Sirouspour, 2011b), which affects $\bar{J}_{sc}^{\dagger}$. A proportional controller is used to define the reference signal for the autonomous subtask which guides the arm (and effectively the operator) to the optimal configuration, i.e.,

$$\dot{c}_{\text{ff}}^d = K_{\text{ff}}(q_s^* - q_s) \tag{4.14}$$

where K_{ff} is a positive scalar. Also, q_s^* and q_s are the optimal and current arm joint variables, respectively. Note that these reference signals are already in the joint space, therefore

$$J_{c_{\text{ff}}} = \begin{bmatrix} I_{N_s \times N_s} & 0_{N_s \times N_b} \end{bmatrix} \tag{4.15}$$

Finally, concatenating (4.13) and (4.15) and also (4.12) and (4.14), $\bar{J}_c$ and $\dot{\bar{c}}^d$ from Eq. (3.6) are obtained:

$$\dot{\bar{c}}^d = \begin{pmatrix} \dot{c}_b^d \\ \dot{c}_{\text{ff}}^d \end{pmatrix}, \qquad \bar{J}_c = \begin{pmatrix} J_b \\ J_{\text{ff}} \end{pmatrix} \tag{4.16}$$

Fig. 4.2 shows an overview of the complete control architecture. As shown in this block diagram, the optimization provides the desired manipulator joint configuration and base movements. Using the result of the optimization, $\dot{\bar{c}}^d$ and $\bar{J}_c$ are calculated and used in the general teleoperation control framework of Chapter 3 in order to move the base autonomously and guide the manipulator towards the desired configuration.

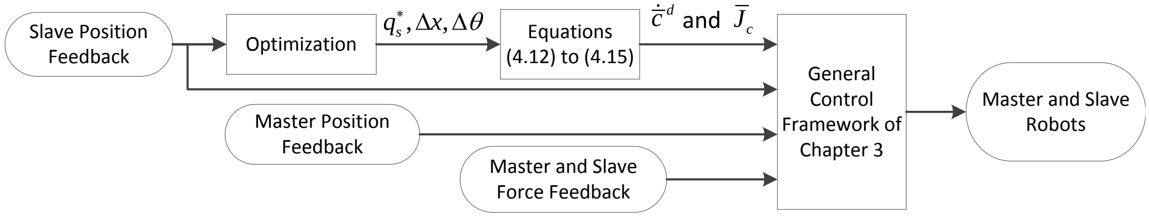

Slave Position Feedback
Optimization
$q_s^*, \Delta x, \Delta\theta$
Equations (4.12) to (4.15)
$\dot{\bar{c}}^d$ and $\bar{J}_c$
General Control Framework of Chapter 3
Master and Slave Robots
Master Position Feedback
Master and Slave Force Feedback

Figure 4.2: Block diagram of the complete mobile manipulator controller.

4.3 Experimental Results

In this section, the results of two sets of experiments are presented. First plots of position and force signals from a single operation of the robot are presented to show the low-level behavior of the control system. Then the results of a human factors study are given to compare operator performance using the proposed asymmetric teleoperation controller with autonomous subtasks to that of a conventional controller.

4.3.1 Control Experiment

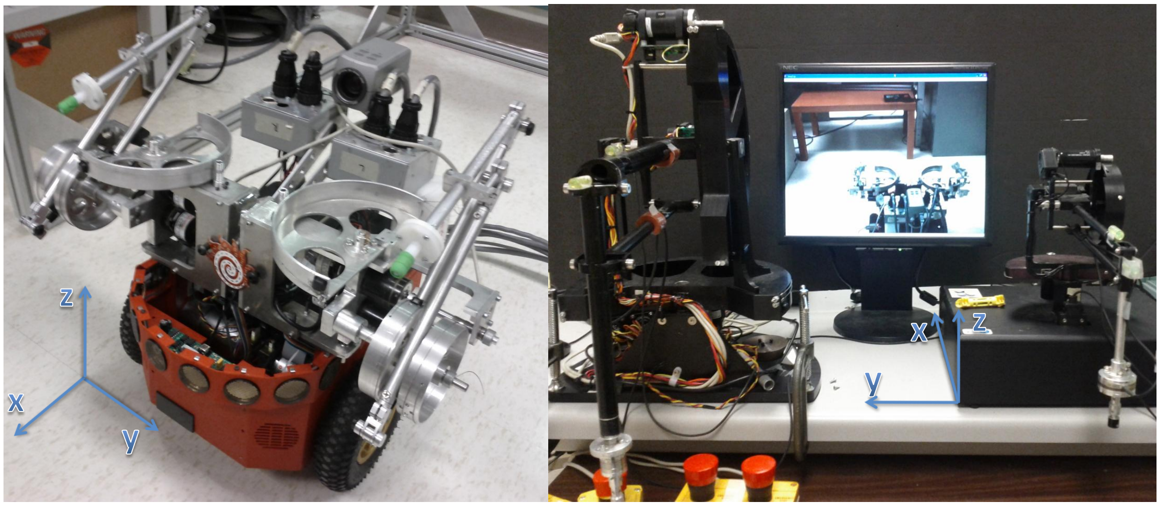
The mobile manipulator used in the experiment is based on a Pioneer P3-DX mobile platform with two custom-built 3 DOF robotic arms mounted on it (see Fig. 4.3a). The Pioneer P3-DX has its own on-board velocity controller. The two arms are kinematically similar to a Sensable Phantom and are equipped with Maxon RE-35 DC motors with optical encoders that have resolution of 20000 counts/rev. They employ a capstan design to provide low friction, zero backlash and back-drivability. On the master side, a Sensable Phantom and a Quanser Integrated Haptic Actuator (IHA) were employed. The Phantom and IHA were equipped with ATI Mini40 and Nano25 F/T sensors to measure the end effector forces, respectively. A fixed Sony DFWVL500 digital camera provided visual feedback of the task. Also, the operator is provided with a clutch to control whether the base can move or not. Fig. 4.3b shows the master console with two haptic devices.

The real-time control code ran under Matlab RTW/Simulink with Quarc 2.0, utilizing a custom interface between the API of the Pioneer P3-DX and Simulink. The Quarc control software operated at a sampling rate of 1 kHz. The optimization routine was run on a separate computer, using *fmincon* in MATLAB at a rate of more than 20 Hz. The two computers communicated through the TCP/IP protocol. The control parameters are given in Table 5.1.

Table 4.1: Control parameters. All units are in SI.

A	0.002	Ω	0.01	Λ	15	Σ	0.5
K_p	I	K_h	I	K_f	I	K_{ff}	0.8
T_f	0.67	C	30π	M	0.015	w_c	0.15

The aim of the task was to demonstrate the robot's ability to manoeuvre in a



(a) (b)

Figure 4.3: The experimental setup used for the control experiment. (a) The mobile manipulator at its initial position in the task space. The task space coordinate frame is demonstrated at the bottom left corner. (b) The master arms and the visual feedback provided to the operator. Master coordinate frame is demonstrated at the bottom.

workspace under the proposed control architecture. The task lasted 30 seconds and the robot successfully moved according to the needs of the operator. Note that the operator used the bird's eye view camera in this task, hence the slave robot related signals are represented in RPF. Fig. 4.4 demonstrates the position of the arms controlled by the operator. The operator moves his arms back at the 7th second, which causes the base to move back as demonstrated in Fig. 4.5. The operator then moves his arms forward and to the right at the 12th second which causes the base to move forward and turn clockwise as in Fig. 4.5. Note that the clutch associated with the base movement is not active before the 7th second and after the 25th second, hence there is no movement seen in Fig. 4.5.

Fig. 4.6 shows the forces sensed by the force sensors at the master side. This

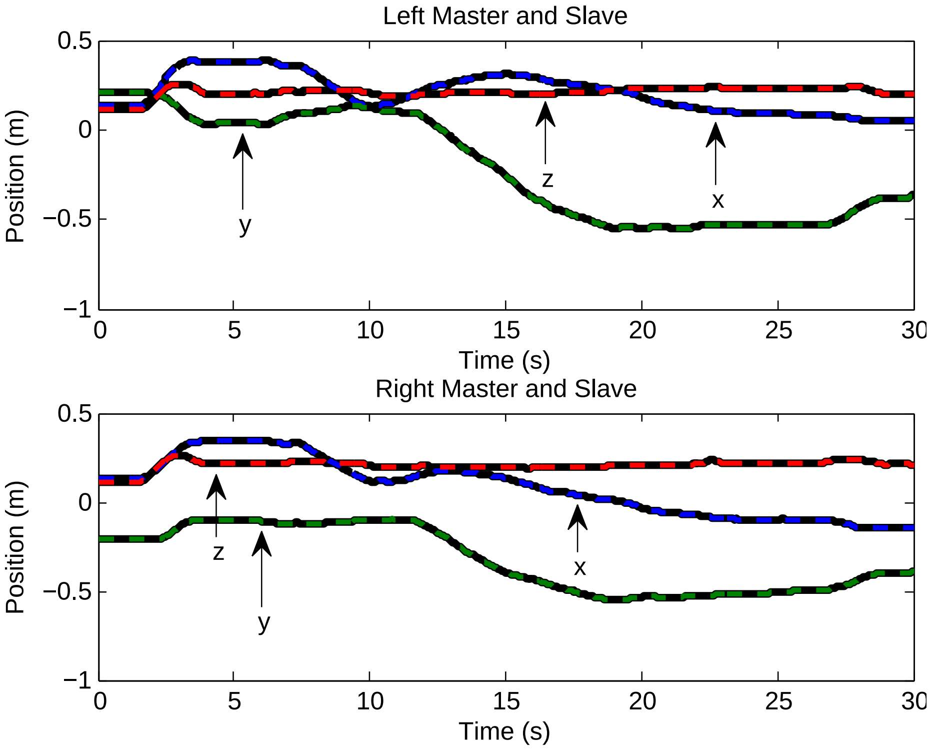


Figure 4.4: Position of master (represented in MF) and slave arms (represented in RPF) in dashed colored and solid black lines, respectively. Blue, green, and red colors represent x, y, and z components, respectively.

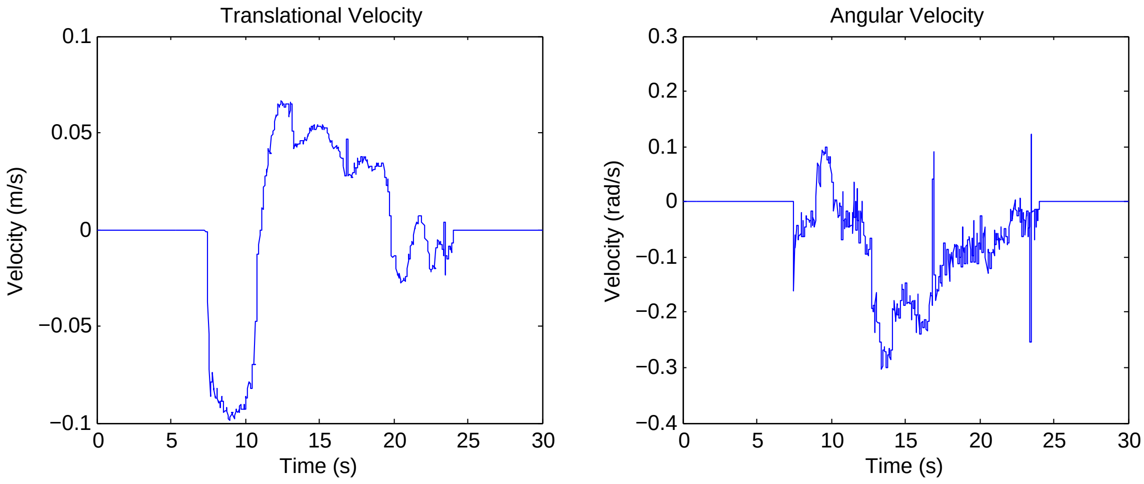


Figure 4.5: Translational and angular velocities of the mobile base in the control experiment.

figure shows how the base movement is reflected back to the operator through haptic feedback. The most significant parts where the movement of the base has made the operator insert extra force are marked with cyan circles. Finally, Fig. 4.7 displays the

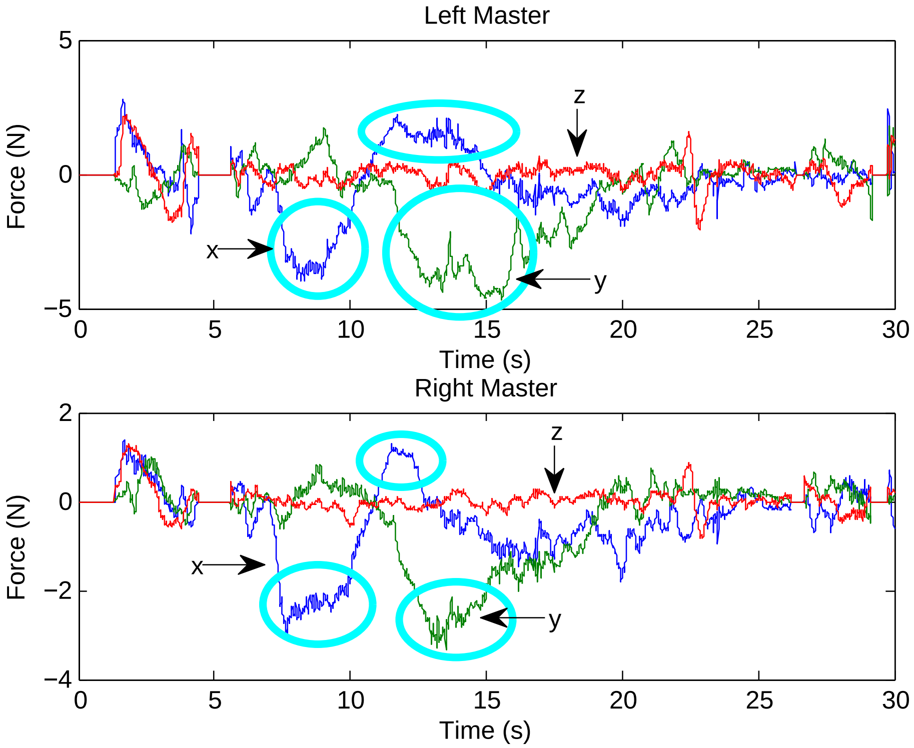


Figure 4.6: Operator input force profiles during the control experiment. Cyan circles highlight the parts where the autonomous subtask $\dot{c}_{ff}$ has inserted considerable amount of force to the operator. Blue, green, and red colors represent x, y, and z components, respectively.

motion of the base during the task in the workspace. It has started from point $[0, 0]$ and moved back to $[-0.29, 0.01]$, then has moved forward and turned right to reach $[-0.01, -0.17]$ and finally the task has ended with base being at $[-0.02, -0.13]$.

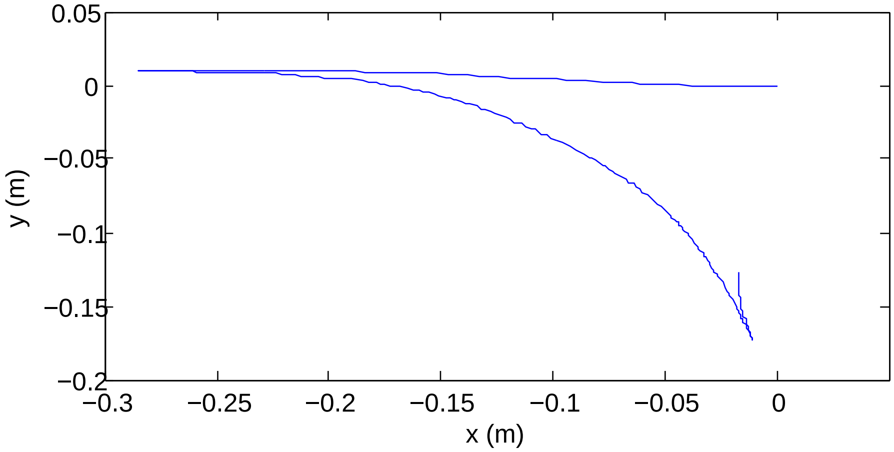


Figure 4.7: Trajectory of base movement in the workspace during the control experiment.

4.3.2 Human Factors Experiments

The goal of this experiment was to evaluate the effectiveness of the proposed control architecture from a system usability perspective. To this end, the participants of the experiment were asked to press two remote buttons in the workspace using the proposed Optimization Based control architecture (OB) and a Conventional architecture (C), in which a different master device is utilized to control the base movement. Both these experiments were modified to be carried out using either an egocentric (EC) viewpoint or a bird's eye (BE) viewpoint, so each participant completed the task four times. Fig. 4.8 shows a view of the task.

The experiment setup and control parameters were exactly the same as those described in the previous subsection. The new device used to control the base independently was a Quanser planar pantograph device with two translational and one rotational degrees of freedom. The movement was restricted in one translational direction and the remaining translational DOF and rotation were mapped to the base

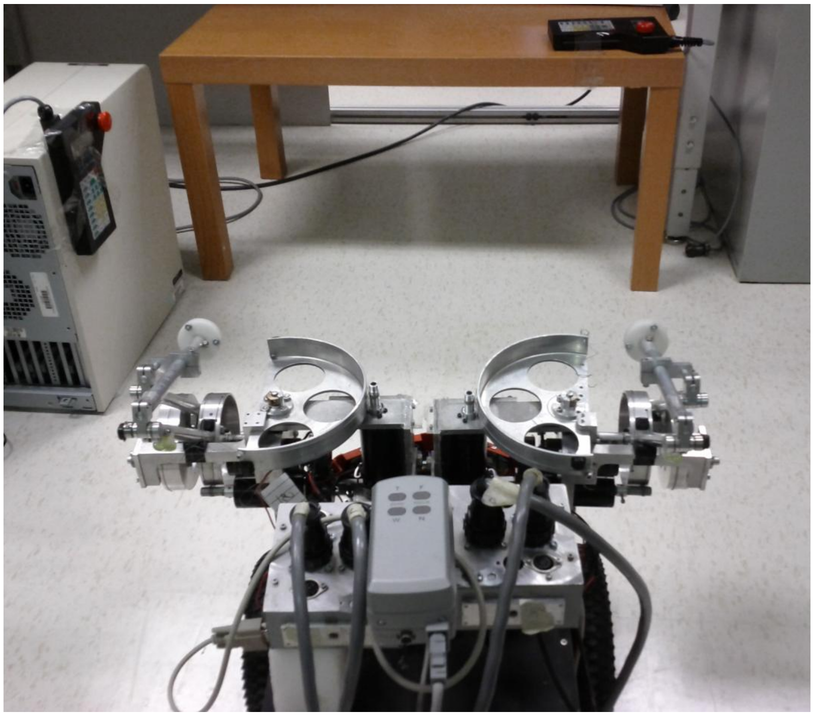


Figure 4.8: A bird's eye view of the task-space in human factors experiment.

translational and angular velocities, respectively.

Fourteen (14) participants were recruited for the experiment. The participants had little to no background in teleoperation and ranged between 22 to 29 years of age. The performance measure used was task completion time (TCT). Given their lack of experience in teleoperation, participants were shortly trained individually prior to performing the task. The results of the experiment are presented in Fig. 4.9. Statistical analysis was performed by the Fisher's Least Significant Difference (LSD) test using a Repeated Measures ANalysis Of VAriance (ANOVA) (Howell, 2004) with a significance level of $\alpha = 0.05$ for all cases. The statistical analysis indicated that the operators using the proposed control architecture had significantly better task completion time when using the EC camera (the protected t test result indicated a probability of $P = 0.000242$ for the null-hypothesis). In the BE viewpoint case, the differences were not statistically significant.

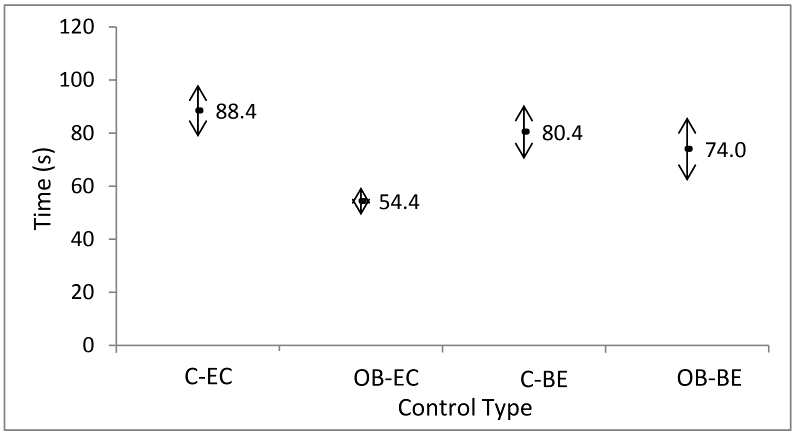


Figure 4.9: Test results of the volunteers performing the task using the proposed Optimization Based control architecture (OB) and a Conventional control architecture (C) while getting visual feedback through egocentric (EC) or bird's eye (BE) viewpoints. The arrows are the standard error of the mean (SEM).

Qualitatively, participants pointed out two major problems of the BE camera. One is the lack of hand-eye coordination when the robot has turned and is not aligned with the camera anymore and the other is poor visual feedback from the manipulation task because the camera was located far from the robot and no zoom function was provided. But on the other hand, most participants admitted that they were more comfortable using the BE camera when they wanted to move the robot around the field because they had a better sense of the whole environment. Since, the major shortcomings of the BE viewpoint are related to the visual feedback, our control approach could not improve the performance in this mode significantly. Another possible explanation for why our proposed approach did not perform better under this camera viewpoint is that the controller is more reliant on hand-eye coordination which this viewpoint lacks.

Chapter 5

Autonomous Viewpoint Planning in Teleoperation of a Mobile Robot

In this chapter a new viewpoint planning controller is presented for the use in teleoperation of a mobile robot. First, attributes of a good viewpoint are defined. Based on these attributes and operator's commands via a natural interface (head tracking), a constrained optimization problem is formulated to determine the best possible camera configuration and trajectory in teleoperation of a mobile robot. Experimental results that demonstrate the effectiveness of the proposed camera view planner are provided in the last section of this chapter.

5.1 Attributes of a Good Viewpoint

Before formulating an optimization problem for finding an optimal viewpoint, one needs to first establish the attributes of a desirable viewpoint in teleoperation. Obviously such viewpoint must cover all the points in the workspace which the operator might be interested in looking at during the task. After reviewing studies on eye fixation locations during visuo-motor tasks, Hayhoe and Ballard (2005) concluded that human eyes mostly fixate on task-specific points in the workspace. Furthermore in their study of eye-hand coordination, Johansson *et al.* (2001) discovered that eyes fixate on a particular point even before the hand reaches it. The results of these studies underscore the importance of determining critical task-specific points that must be included in the operators view of the workspace. Teleoperation tasks can be categorized as navigation or manipulation type. A discussion of critical task-specific points for each of these types of tasks and also other attributes of an optimal viewpoint follows next.

5.1.1 Navigation Tasks

Based on the conclusions of the study in Johansson *et al.* (2001), it can be argued that in navigation-type teleoperation tasks, at any point in time, the operator should have a view of (i) the body of the robotic manipulator at the present time (ii) the destination of robot in the near future, and (iii) any obstacle on the robot's path from its current location to its destination. These three requirements establish critical points that must be included in the operators view in a navigation-type task. It should be noted, however, that the requirement of including these critical points in the operator's view would not uniquely determine the camera position but rather impose constraints on acceptable positions. Fig. 5.1 illustrates a generic scenario in which several instances of camera configuration from an infinite solution space that satisfy the requirement for inclusion of critical points are shown.

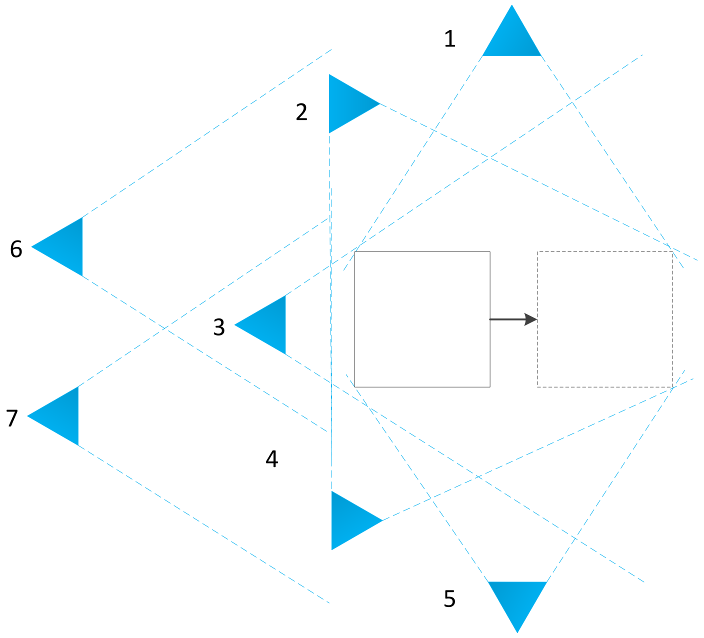


Figure 5.1: Top view of the robot and seven possible camera configurations that would cover critical points in the workspace. The solid rectangle represents the robot with the arrow aligned with its current velocity. Estimated future position of the robot is shown with the dashed rectangle. Seven valid camera configurations are represented with solid blue triangles with the boundaries of their field of view shown as dashed lines.

Human operators are naturally accustomed to navigation using an egocentric viewpoint and studies have shown that such viewpoint is the most effective for local navigation (Olmos *et al.*, 2000; Hollands and Wickens, 1999; Chen *et al.*, 2007). In teleoperation, an egocentric viewpoint would place the camera on the remote robot. The most intuitive camera orientation would be the one aligned with the robot heading direction. Maximizing egocentricity is an important objective in our viewpoint optimization approach. In this sense, Camera Position 3 would be preferable among the potential solutions shown in the example of Fig. 5.1, if such position satisfies the workspace/velocity limits of the camera positioning system. However in certain circumstances, such as when the robot is trapped or it attempts to move in a confined space (Murphy *et al.*, 2000), the operator may need to rotate the camera to look at a different direction in the workspace. In the proposed approach in this thesis, the operator would be able to rotate the camera using head movements tracked by a tracker device. Details concerning the integration of these commands into the optimization problem formulation will be presented in the next section.

As expressed in Chapter 2, ensuring that the visual feedback does not contain unwanted vibrations is a vital part of any system with a moving camera. The proposed optimization-based approach to camera positioning penalizes large camera movement acceleration when finding an optimal configuration. Minimizing the acceleration of the camera positioning system would reduce induced vibration in the video feed in the first place. Image stabilization could further reduce these vibrations, if needed.

5.1.2 Manipulation Tasks

Critical points for manipulation tasks are: (i) the manipulator end-effector position at current time, (ii) the manipulator end-effector position in near future, (iii) any obstacle in the end-effector path from the current to the future position, and (iv) at least part of the robot body so that the operator maintains a global spatial awareness. Determining the best orientation for the camera in manipulation tasks is highly task-dependent and should be left to the operator. In the proposed approach, the operator guides the camera orientation using the head tracking device. Another objective is to minimize the the distance of the camera from the tip(s) of the end effector(s) to give the operator as large view of the task as possible. Similar to the case of navigation task, the workspace/velocity limitations of the camera positioning system as well as the required acceleration to reach the desired configuration must be taken into account in view planning. However, these considerations are less critical in a manipulation task compared to navigation.

5.2 Optimization-based View Planning

In this section a constrained optimization problem is formulated that takes into account all the metrics discussed in the previous section. The body of the robot or the manipulator end-effector could be modelled by a single or multiple points, depending on its size and how important it is to see the whole body in the video feed. Modelling of the robot by a single point could result in partial visual feedback from the robot. On the other hand, modelling it by multiple points increases the number of constraints and the complexity of the objective function, resulting in increased computations. Throughout this thesis, the robot is modelled by a single point for simplicity. Moreover, obstacles are represented by bounding spheres to simplify the formulation of the constraints. Fig. 5.2 shows a schematic of the robot, camera system and the obstacle and serves as a reference for defining variables in the problem formulation. In this figure:

- T_m is the “camera planning horizon”. Due to velocity/acceleration limitations of the camera platform, it is impossible to move to the optimal configuration instantaneously; therefore the optimal configuration is determined for T_m seconds in future. This would give the positioning system sufficient time to move the camera to the optimal configuration.

- $T_f > T_m$ is “robot future position horizon”. As mentioned in the previous section, the position of the robot in near future is considered an important point that must be kept in the operator's field of view. In this sense, the predicted position of the robot in $T_f > T_m$ seconds in future must be taken into account in planning the viewpoint for T_m seconds in future.

- P_0, P_1 and P_2 are the current, and estimated positions of the robot at T_m and T_f seconds ahead, respectively.

- P_{I_1} to P_{I_3} are "internal points" at times T_{I_1} to T_{I_3}, respectively ($T_{I_1} < T_{I_2} < T_{I_3} < T_m < T_f$), which are used to design a smooth trajectory for the camera positioning system. The number of these internal points could vary from one to any arbitrary number depending on the task and the value of T_m; however for clarity of presentation only three points have been considered in the rest of the chapter.

- C_s and r_s are the centre and radius of an obstacle. In this particular example only one obstacle is considered, but there is no limit on the number of obstacles in general.

- P_{c_0}, $P_{c_{I1}}$ to $P_{c_{I3}}$, and P_c are the current position of the camera, its positions at the internal times T_{I_1} to T_{I_3}, and its final desired position at T_m in the 3D space, respectively.

- The superscripts 1 and 2 for $P_{c_{I1}}$ to $P_{c_{I3}}$ and P_c differentiate two possible solutions. These two solutions are compared later in this section.

- V_c and V_{heading} are unit vectors representing the orientation of the camera in its optimal position, and the robot heading direction, respectively. Note that

V_{heading} is defined for the robot T_m seconds into the future.

Based on the discussion in the previous section, in the absence of any constraints and inputs from the head tracker, the ideal configuration for the camera would be to reach position P_1 with the orientation V_c aligned with the robot heading direction V_{heading} after T_m seconds. In order to calculate P_1, assuming constant translational (v) and rotational velocity (ω) during the prediction horizon, one can write the following for a planar mobile robot:

$$\Delta x = \int_0^{T_m} v\cos(\omega t)\,dt = \frac{v}{\omega}\sin(\omega T_m) \tag{5.1}$$

$$\Delta y = \int_0^{T_m} v\sin(\omega t)\,dt = \frac{v}{\omega}\left(1-\cos(\omega T_m)\right) \tag{5.2}$$

where Δx and Δy are displacements in x and y directions from 0 to T_m with respect to the robot-attached coordinate frame at time 0, as shown in Fig. 5.2. Using (5.1) and (5.2), one could predict the position of the robot at time T_m as

$$P_1^W = P_0^W + R_R^W \frac{v}{\omega}\begin{bmatrix} \sin(\omega T_m) \\ 1-\cos(\omega T_m) \\ 0 \end{bmatrix} \tag{5.3}$$

where superscripts W and R indicate whether a variable is defined in the workspace or robot coordinate frame. P_0^W and R_R^W are the position and orientation of the robot coordinate frame with respect to the workspace frame at time 0. P_2^W could also be predicted from (5.3) by substituting T_m for T_f.

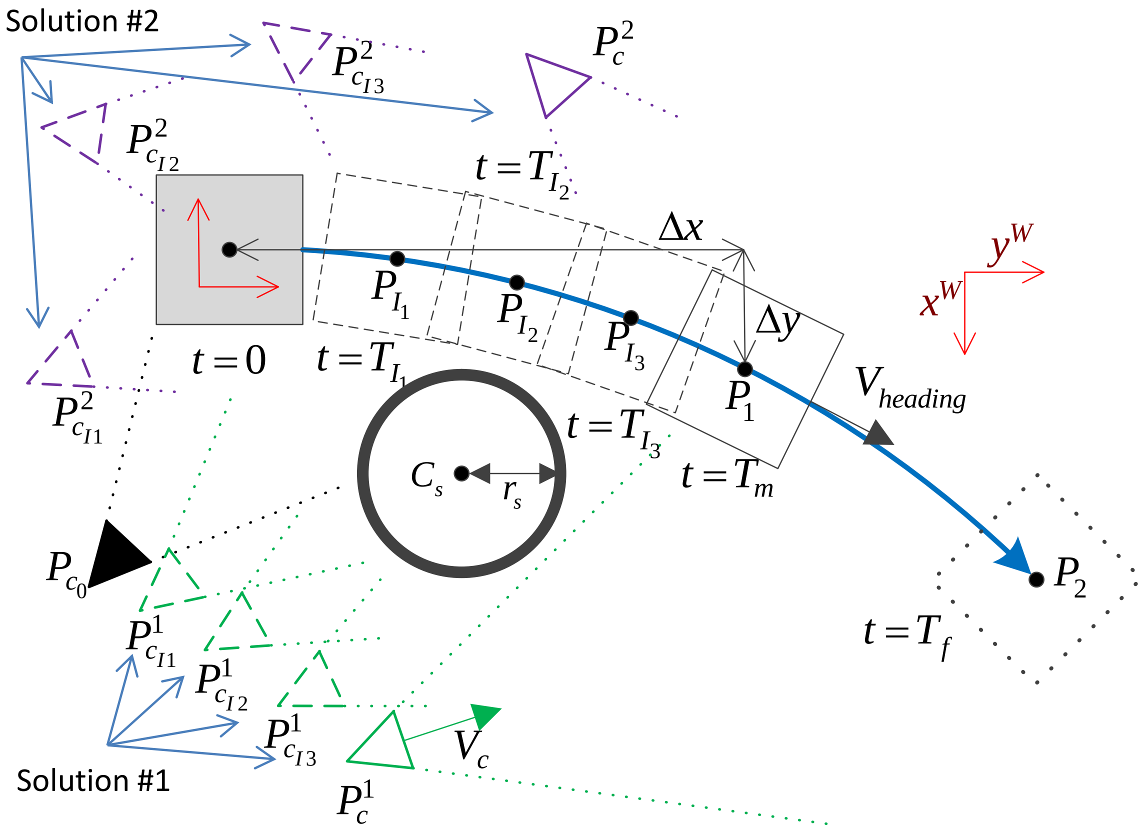


Figure 5.2: Top view of a possible robot path and two sets of alternative camera configurations and trajectories in a navigation task. An obstacle is shown by a black circle. The solid grey rectangle represents the current position of the robot moving on the curved blue line. The dashed, solid white, and dotted rectangles show the estimated positions of the robot at the internal time steps, T_m, and T_f seconds, respectively. The black filled triangle represents initial position of the camera. Solid and dashed triangles show the optimal position of the camera and its trajectory at internal time steps; the green and blue colours denote the possible solutions for camera positioning. The dotted lines extending from the triangles are the boundaries of the camera field of view. The robot and workspace coordinate frames are also shown in red. In case of manipulation task, the rectangle represents the manipulator end-effector.

5.2.1 Head Tracker Inputs

The variables P_{task} and V_{task} are determined by the head tracker inputs, which are interpreted as velocity commands. This is because the operator has to look at the video feed on a fixed display and therefore cannot rotate or move his/her head very much for too long. The block diagram in Fig. 5.3 shows how the input signals from the head tracking device are modified. These inputs are categorized as rotational and translational motion commands and are discussed separately below.

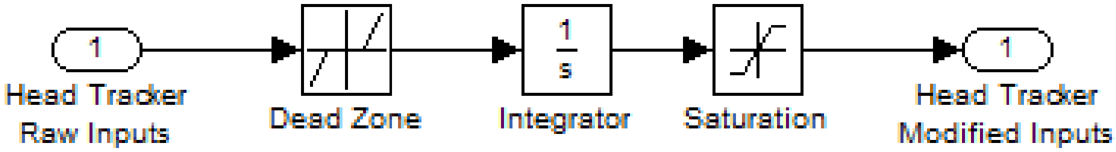


Figure 5.3: Inputs from the head tracker pass through a dead zone block first to eliminate unintentional motion commands from the operator. The output of the dead zone block is then integrated and capped at some maximum and minimum values.

Head Tracker Rotational Inputs: Since the camera rotation about its x axis (roll) does not really change the viewpoint of the operator, it is not considered in this work; however, if needed, this could be easily added to the proposed view planning framework in the same way as the other two rotations (pitch and yaw). By default, the camera's desired pointing direction, V_{task}, is equal to V_{heading}, but the operator can rotate this vector by rotating his/her head:

$$V_{\text{task}}(t) = \begin{cases} R_R^W R_z(\phi_{ht}(t)) R_y(\psi_{ht}(t)) V_{\text{heading}}(t) \\ \quad \text{if rotation commands are initiated.} \\ V_{\text{task}}(t-1) + K_{br}\left(V_{\text{heading}}(t-1) - V_{\text{task}}(t-1)\right) \\ \quad \text{if rotation commands are NOT initiated in a navigation task.} \\ V_{\text{task}}(t-1) \\ \quad \text{if rotation commands are NOT initiated in a manipulation task.} \end{cases} \tag{5.4}$$

In the above equation, R_R^W was already defined and $\phi_{ht}(t)$ and $\psi_{ht}(t)$ are the modified yaw and pitch inputs from the head tracker, respectively; $K_{br} < 1$ is a scaler. Note that V_{heading} is represented in the robot's coordinate frame. The first case in Eq. (5.4) shows how V_{heading} is rotated by the operator commands in the robots's coordinate frame and then is represented in the workspace frame. Note that the operator can turn the desired orientation of the camera in both manipulation and navigation tasks, but practical evidence shows that if the operator rotates the camera during navigation to see a particular point, he/she has to spend some time rotating back the camera to the default orientation once there is no more need to see that particular point (Hughes and Lewis, 2005). In order to reduce the cognitive load on the operator, the backward rotation is automatically carried out during navigation, according to the second case in (5.4). When the operator does not initiate a rotation command in a manipulation task, V_{task} maintains its previous value according to the third case in (5.4).

Head Tracker Translational Inputs: By default the camera's desired position, P_{task}, is equal to P_1 as stated before. In manipulation tasks there might be a need to move the camera around and inspect other areas. In the proposed approach, the translational movements of the operator's head can partially move P_{task}, as illustrated below:

$$P_{task} = P_1 + K_t R_C^W \begin{bmatrix} x_{ht}(t) \\ y_{ht}(t) \\ z_{ht}(t) \end{bmatrix} \tag{5.5}$$

where $x_{ht}(t)$, $y_{ht}(t)$, and $z_{ht}(t)$ are the x, y, and z translations of the operator's head represented in the camera's coordinate frame and are modified according to Fig. 5.3. K_t is a scaler and R_C^W is the rotation matrix between the workspace and camera coordinate frames. Note that the modified inputs $x_{ht}(t)$, $y_{ht}(t)$, and $z_{ht}(t)$ are saturated so P_{task} cannot move far away from P_1. Moreover, this can only be done during manipulation since moving the camera away from the robot could be too dangerous while navigating.

5.2.2 Optimization Problem Formulation

The camera configuration is determined by solving the following optimization problem:

$$\begin{aligned} \min_X \quad & \|P_c - P_{\text{task}}\|^2 - k_o \cos(\angle(V_c, V_{\text{task}})) \\ & + \sum_{j=1}^{m} k_{v_j} \text{OBV}(P_{I_j}, P_{c_{Ij}}) + K_a \int_0^{T_m} \|a\|^2 + \|\alpha\|^2 \, dt \end{aligned} \tag{5.6}$$

subject to:

i $L_{X_\gamma} \leq X_\gamma \leq U_{X_\gamma}$

ii $|y_i^c| \leq \tan(\frac{\alpha}{2})x_i^c$

iii $|z_i^c| \leq \tan(\frac{\beta}{2})x_i^c$

iv $|\overline{P_cC_{s_k}}|^2 - \frac{(\overline{P_cP_i}.\overline{P_cC_{s_k}})^2}{|\overline{P_cP_i}|^2} + r_{s_k}^2.(\tilde{u}(\lambda - 1) + \tilde{u}(-\lambda)) \geq r_{s_k}^2$

v $\Gamma_{c_{Im}} - v_{m,\Gamma}(T_m - T_{I_m}) \leq \Gamma_c \leq \Gamma_{c_{Im}} + v_{m,\Gamma}(T_m - T_{I_m})$

vi $\Gamma_{c_{I(j-1)}} - v_{m,\Gamma}(T_{I_j} - T_{I_{j-1}}) \leq \Gamma_{c_{Ij}} \leq \Gamma_{c_{I(j-1)}} + v_{m,\Gamma}(T_{I_j} - T_{I_{j-1}})$

vii $\Gamma_{c_0} - v_{m,\Gamma}T_{I_1} \leq \Gamma_{c_{I1}} \leq \Gamma_{c_0} + v_{m,\Gamma}T_{I_1}$

where

$$X = [x_c, y_c, z_c, \phi_c, \psi_c, x_{c_{Ij}}, y_{c_{Ij}}, z_{c_{Ij}}, \phi_{c_{Ij}}, \psi_{c_{Ij}}]^T \tag{5.7}$$

$$\begin{aligned} \mathrm{OBV}(P_I, P_{c_I}) = {} & \tilde{u}(\underset{k,j}{\tilde{\max}}(r_{s_k}^2 - (|\overline{P_{c_I}C_{s_k}}|^2 \\ & - \frac{(\overline{P_{c_I}P_I}.\overline{P_{c_I}C_{s_k}})^2}{|\overline{P_{c_I}P_I}|^2} + r_{s_k}^2.(\tilde{u}(\lambda - 1) + \tilde{u}(-\lambda))) \\ & , |y_I^c| - \tan(\frac{\alpha}{2})x_I^c, |z_I^c| - \tan(\frac{\beta}{2})x_I^c)) \end{aligned} \tag{5.8}$$

$$\lambda = \frac{(\overline{P_cP_i}.\overline{P_cC_{s_k}})}{|\overline{P_cP_i}|^2} \tag{5.9}$$

Here $i \in [1, n]$, $j \in [1, m]$, and $k \in [1, l]$; n, m, and l are the total number of important points, internal time steps, and obstacles, respectively, and X is the vector of the optimization variables defined in (5.7). Moreover, ϕ_c and ψ_c are the pan and tilt orientation of the camera at time T_m; x_c, y_c, and z_c are the coordinates of P_c; $x_{c_{Ij}}$, $y_{c_{Ij}}$, and $z_{c_{Ij}}$ are also defined similarly for $P_{c_{Ij}}$. It should be noted that the number of optimization variables depend on the number of internal points considered, e.g., choosing to have three internal points would yield 20 optimization variables. The role of each term in the cost function and the constraints of the above optimization problem are discussed below

∘ *Objective Function:* The objective function is a weighted sum of multiple costs, with the scalers k_o, k_{v_j}, and k_a determining the relative importance of each term. Ideally the camera should be positioned at P_{task} with its optical axis aligned with V_{task}. The first and second terms penalize the distance between the actual camera position P_c and P_{task} and the angle between the actual camera orientation V_c and V_{task}, respectively. The third term contains m terms in itself, utilizing OBV function which is defined in (5.8). For any internal point, this function would take a value of one if any obstacle is blocking the view of the operator, or the robot is not inside the field of view, otherwise it is zero. The actual form of this function is derived using terms similar to those used in Constraints ii to iv; investigating those constraints would help the reader better understand how the OBV function is formulated. The notations $\tilde{u}$ and $\tilde{\max}$ are smooth approximations of the step and max functions, introduced to facilitate solving the optimization problem. These approximations are given by:

$$\tilde{u}(x) = \frac{1}{2} + \frac{1}{\pi}\arctan(M.x) \text{ where } M \gg 1 \tag{5.10}$$

$$\tilde{\max}(x,y) = \frac{xe^{Mx} + ye^{My}}{e^{Mx} + e^{My}} \text{ where } M \gg 1 \tag{5.11}$$

The third term in the objective function is meant to encourage a camera path to the final configuration in which the operator's view of the robot is never blocked by an obstacle. However, it is possible that at certain internal points the view is blocked, but at the optimal camera configuration at T_m second the operator must be able to see the robot; this is strictly enforced by the Constraints ii to iv. Furthermore, the designer can decide whether it is more important for him/her to maintain the view in the early stages of the camera's movement or the final stages by choosing appropriate values for k_{v_j}.

The last term in the objective function is the integral of sum of the square of the translational and rotational accelerations of the camera from the current time until it reaches the desired configuration at T_m. Reducing these accelerations ensures smooth vibration-free movement of the camera system. It should be noted that the internal camera points are used to design a smooth trajectory for the camera to reach the desired configuration at time T_m. Parabolic spline interpolation (De Boor, 1978) has been used in this thesis to derive the camera trajectory based on the internal points, as discussed in the Appendix. This interpolation is computationally inexpensive and ensures first-order continuity at the internal points for a smooth camera movement. Since the designed trajectory is fed into the low-level controller of the camera positioning system, second-order continuity is not necessary. The acceleration terms (a and α) can be explicitly expressed as a function of time and the optimization variables as shown in the appendix.

The two possible solutions presented in Fig. 5.2 demonstrate an example of a trade off between the third and the fourth terms of the objective function. In the first solution, the operator's view is blocked by the obstacles at times T_{I_2} and T_{I_3} but the movement of the camera, hence its acceleration is relatively small. Conversely in the second solution, the operator's view is never obscured but the camera has to move very rapidly with high accelerations. One can expect that setting a high value for k_a would favour the first solution whereas setting large values for k_{v_j} would tend to yield a solution close to the second case.

An alternative formulation for the optimization problem would move the acceleration related term from the objective function to the constraints. This way, it can be guaranteed that the acceleration would never surpass a limit specified by the user.

∘ *Constraint i:* The first constraint expresses the workspace boundaries of the camera positioning system. X_γ represents the γ^{th} element of X, also L_{X_γ} and U_{X_γ} represent the lower and upper limits of the X_γ variable, respectively; these are the workspace limits of the camera positioning system.

∘ *Constraints ii and iii:* These constraints must be written for every important point. α and β stand for the camera's horizontal and vertical angle of view, respectively. x_i^c, y_i^c, and z_i^c are the corresponding coordinates of the i^{th} important point represented in the camera frame. Fig. 5.4 shows how these two constraints ensure each important point are inside the camera's field of view.

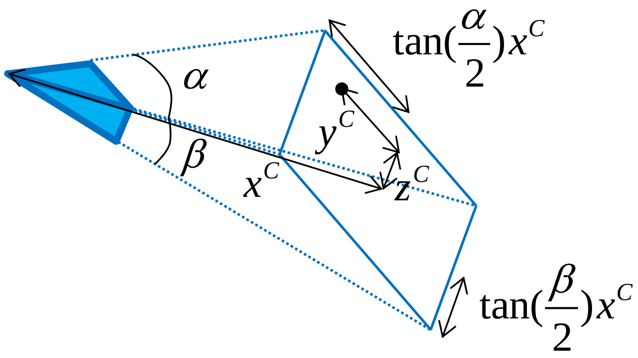


Figure 5.4: 3D view of the camera image. The camera and the image are shown by the blue pyramid and parallelogram, respectively.

∘ *Constraint iv:* This constraint ensures that the operator's view of the important points is not obscured by any obstacle; it must be written for every combination of important points and obstacles.

The variable λ is defined in (5.9) and represents the projection point of the centre of sphere onto the line connecting the camera and the point of interest. A $\lambda = 1$ means that the projection falls exactly on the point of interest while $\lambda = 0$ means it falls on the camera, i.e., $\overline{P_cP_i} \perp \overline{P_cC_{s_k}}$. The formula in Constraint (iv) is obtained from Held (1997) where a set of intersection tests are introduced. The test in Held (1997) is between an infinite line and a sphere and does not include the term $r_{s_k}^2 \left(\tilde{u}(\lambda - 1) + \tilde{u}(-\lambda)\right)$. Since we are only interested in finding the intersection between the line segment $\overline{P_cP_i}$ and the sphere, the last term is added in this thesis to ensure that the constraint is satisfied when the line intersects with the sphere outside the segment, i.e., $\lambda > 1$ or $\lambda < 0$. It is worth mentioning that $|\overline{P_cC_{s_k}}|^2 - \frac{(\overline{P_cP_i}.\overline{P_cC_{s_k}})^2}{|\overline{P_cP_i}|^2}$ is a non-negative number, thus the constraint is always satisfied for $\lambda > 1$ or $\lambda < 0$.

$\circ$ *Constraints v to vii:* These constraints guarantee that the camera positioning system is capable of moving the camera to the desired configuration. Γ stands for any of the variables x, y, z, ϕ, or ψ; $v_{m,\Gamma}$ is the maximum achievable velocity of the camera positioning device in the Γ direction. These constraints must be written for all the five DOF of the camera positioning system.

Similar to the one in Chapter 4, the optimization problem defined above is nonlinear and non-convex. Given the small size of the problem, it can be effectively solved using SQP methods, although there is no guarantee that a global solution would be found. A drawback of SQP algorithms is that they rely on the gradient of the objective function and constraints. Hence, functions such as **step** and **max** had to be approximated by smooth functions in (5.10) and (5.11). Using a large M in these approximations would result in steep slopes that could slow down or prevent the SQP algorithm convergence. To avoid this problem the optimization problem can be solved in multiple stages increasing the value of M gradually and using the result of each stage as the starting point in the next stage.

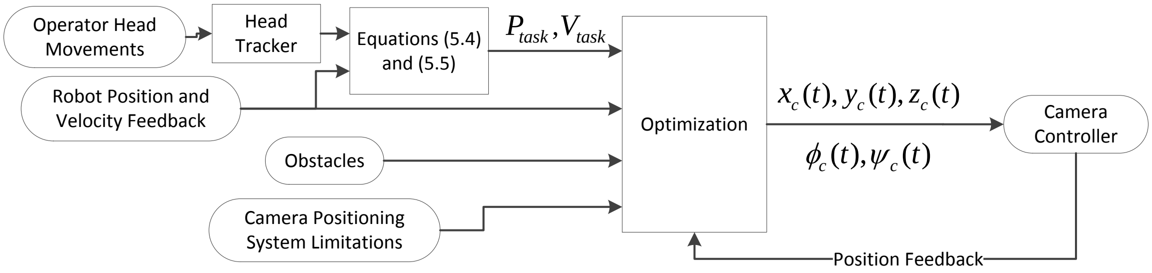


Figure 5.5: Block diagram of the proposed camera viewpoint planner.

Fig. 5.5 shows the block diagram of the proposed camera viewpoint planner. To ensure that the camera movements would not negatively impact the operators eye-hand coordination in carrying out the bilateral teleoperation task, the coordinate transformation between the master and slave robots coordinate frames must be adjusted accordingly. In the present work, the master coordinate frame always rotates with the camera pan and tilt angles. To this end, the master robot position, velocity, and force signals are passed through the following coordinate transformation:

$$T_C = \begin{bmatrix} cos(\phi) & -sin(\phi) & 0 \\ sin(\phi) & cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos(\psi) & 0 & sin(\psi) \\ 0 & 1 & 0 \\ -sin(\psi) & 0 & cos(\psi) \end{bmatrix} \tag{5.12}$$

where ϕ and ψ are the current pan and tilt rotations of the camera. Also, every signal sent to the master from the slave is transformed by T_C^{-1}. The impact of this compensation on the operators performance is investigated in the human factor study presented in the last section of this chapter.

5.2.3 Finding a Solution when the Original Problem Becomes Infeasible

Here a solution is proposed to address situations in which the original optimization problem would become infeasible. To this end, a challenging navigation scenario that could give rise to such cases is investigated below.

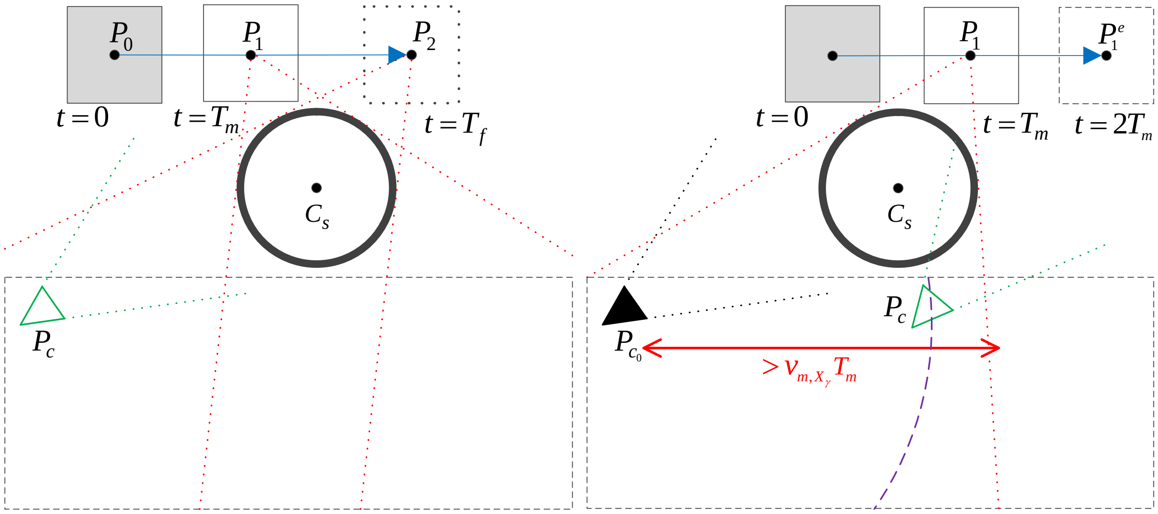
Assume that the mobile robot is passing by an obstacle as shown in Fig. 5.6a while the workspace limits of the camera positioning system prevent it from following the robot around the obstacle. At some point, especially if the robot is moving fast, it becomes impossible to see the two important points. The predicted position of the robot both at "camera planning horizon", P_1, and at "robot future position", P_2, as required by the original problem formulation. Following the tangent lines in Fig. 5.6a, one could easily observe that for the camera to see P_1, it has to move to the left half of the camera positioning system workspace; meanwhile, P_2 could only be viewed from the right half side of the workspace. In such cases a feasible solution to the optimization problem is found by excluding P_2 from the group of *important points*. It is reasonable to argue that seeing the robot at "camera planning horizon" has priority than being able to see its future position. The camera position P_c in Fig. 5.6a is a possible (not necessarily the optimal) solution after the omission of P_2 from the list of important points.

Fig. 5.6b depicts the scenario as in Fig. 5.6a after T_m seconds. The camera has moved to the optimal configuration of Fig. 5.6a and the robot is still moving on its original path. P_2 is still excluded from the optimization problem as decided in the previous step. Since the robot is moving past the obstacle, at some point the camera has to move to the right half of the workspace to catch up with the robot. As demonstrated in Fig. 5.6b, the velocity constraints of the camera positioning system might prevent the camera from being able to see the robot at the *camera planning horizon*; this would make the optimization problem infeasible due to *Constraints v to vii*. In such cases, the problem is revised to grant extra time to the camera positioning system to move to a desired configuration i.e. by doubling the value of T_m. If the optimization problem is still infeasible, T_m could be further increased.

The methods described above should address infeasible cases of the original optimization problem. Obviously, the optimization problem must return to its original form after such situation is resolved. In this thesis, the two cases discussed above are considered as two stages of resolving infeasible scenarios. Once the problem becomes infeasible, first P_2 is excluded from the formulation and if the situation still persists T_m would be increased until a feasible solution is found. The algorithm returns to the pervious stage formulation as soon as it becomes feasible again. This ensures that the optimization problem maintains its original form once the situation is resolved.

5.3 Experimental Results

In this section, the results of two sets of experiments with the proposed camera view planning system in teleoperation of a mobile robot are presented. First, the basic operation of the system is demonstrated by a set of graphs showing the trajectories of the robot and the camera system in a single experiment. Then, the results of a human factors study are given to compare the operator's performance using the proposed controller and two conventional methods for controlling camera viewpoint.



(a) (b)

Figure 5.6: Top view of a possible robot path in a challenging navigation task. The internal points are not displayed to avoid confusion. The grey square represents the current position of the robot moving on the blue line. The big dashed rectangle is the boundary of the camera workspace; the obstacle is shown by a circle. (a) The robot has just started going behind the obstacle. Solid and dotted squares show the estimated positions of the robot after T_m and T_f seconds, respectively. The red dotted lines are tangent to the obstacle indicating areas of the workspace in which P_1 or P_2 view will be blocked by the obstacle. The green triangle and dotted lines show a possible optimal position for the camera after excluding P_2 from the list of important points. (b) The robot continues its path past the obstacle. P_2 is excluded from the diagram because of the decision made in the earlier time step. The solid and dashed squares show the estimated positions of the robot T_m and $2T_m$ seconds into the future. The red dotted lines are tangent lines to the obstacle indicating the areas of the workspace in which P_1 is visible. The filled black triangle shows the current position of the camera which was optimized in the earlier time step and the dashed purple arc represents its maximum reachable workspace in T_m seconds. The green triangle represents a possible solution after extending the *camera planning horizon*, T_m, to $2T_m$.

5.3.1 System Operation Experiment

The experimental setup is similar to the one explained in Chapter 4. It consists of a P3-DX mobile robot on the slave side and a Quanser 3-DOF planar pantograph equipped with ATI Nano25 F/T sensor to measure the end effector force/torque on the master side, shown in Fig. 5.7. A Sony DFW-VL500 digital camera is mounted on a Directed Perception Pan-Tilt Unit D46-17 that is moved by a custom made planar gantry system. The Pan-Tilt unit provides two rotational DOF (pitch and yaw) and the gantry system adds two translational DOF (x^W and y^W in Fig. 5.7), yielding four DOFs for the camera positioning system. The gantry system has a workspace of 90 cm by 115 cm and is fixed at 90 cm above ground. A Microsoft Kinect for Windows tracks the operators head movements (as discussed in Section 5.2.1) for the use as additional commands to the viewpoint planner. Also, the operator is provided with two clutches; one to control whether a navigation task or a manipulation task is being carried out and another to distinguish whether the head movements are intentional commands to planner, or merely unintentional natural motion. The teleoperation control is based on the framework of Chapter 3. In the present work, this framework is utilized to coordinate a 3-DOF master arm with a kinematically deficient nonholonomic mobile robot. Fig. 5.7 shows the entire experimental setup.

The teleoperation and the gantry control systems run under Matlab RTW/Simulink with Quarc 2.0 at a sampling rate of 1 kHz. The optimization runs on a separate computer, using *fmincon* in MATLAB at a rate of 5 Hz, i.e., the optimal trajectory of the camera is updated after it reaches the first internal point from the previous trajectory. The two computers communicate through the TCP/IP protocol. Three internal points are considered in the formulation of the optimization problem; the

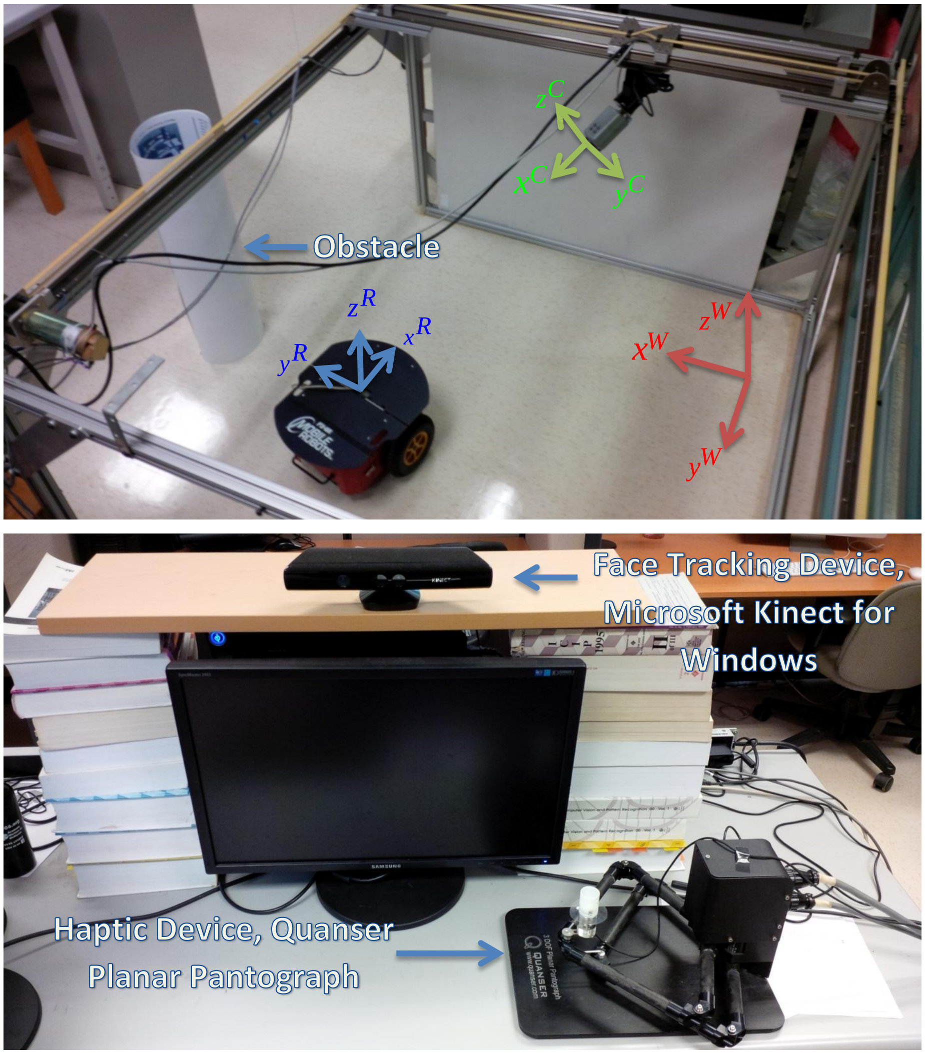


Figure 5.7: Top: The experimental setup. View of task field with the robot, camera, gantry system, and the cylindrical obstacle. Coordinate frames of the workspace, camera, and robot are shown in red, green, and blue, respectively. Bottom: The operation station consisting of monitor display, Microsoft Kinect for Windows on top of it, and Quanser planar pantograph.

control parameters are given in Table 5.1.

Table 5.1: Control parameters. All units are in SI.

T_m	T_f	T_{I_1}	T_{I_2}	T_{I_3}	K_t	K_{br}
0.8	2.8	0.2	0.4	0.6	1.5	0.33
K_o	K_a	K_{v_1}	K_{v_2}	K_{v_3}	α	β
3	0.1	0.5	0.75	1	0.41	0.31

The experiment starts with the robot and the camera resting at $[0.7 \quad 0.19]^T$ and $[0 \quad 0.19]^T$, respectively. The movement of the base and camera are demonstrated in the $x - y$ plane in Fig. 5.8. Some important points in the experiment have time stamps so that the reader can correspond them to Fig. 5.9 which shows the x and y positions of the robot and camera with respect to time; the z components of the positions are constant so they are omitted from the figures.

In the experiment, the operator first turns his head to the right to inspect the wall next to the robot. Then, he turns his head back to the left to restore a forward facing camera and start moving. These commands and their effect on the orientation of the camera can be clearly seen in Fig. 5.10. Next, the operator moves the robot forward and goes around the obstacle, which creates a situation similar to that of Fig. 5.6b. The camera follows the robot smoothly until it is about to move behind the obstacle at $t = 28.5s$. At this time, the original optimization problem becomes infeasible and T_m is temporarily extended so that the camera can move to the other side of the obstacle; the operator's view of the robot is restored at $t = 30s$. Note that the camera starts moving around the obstacle just before the view is about to be blocked by the obstacle, which highlights the predictive nature of the controller. The robot stops next to a new wall at $t = 34s$. The operator moves his head backward and turns it up to get a better view of the wall at $t = 38s$. As shown in Fig. 5.9 and

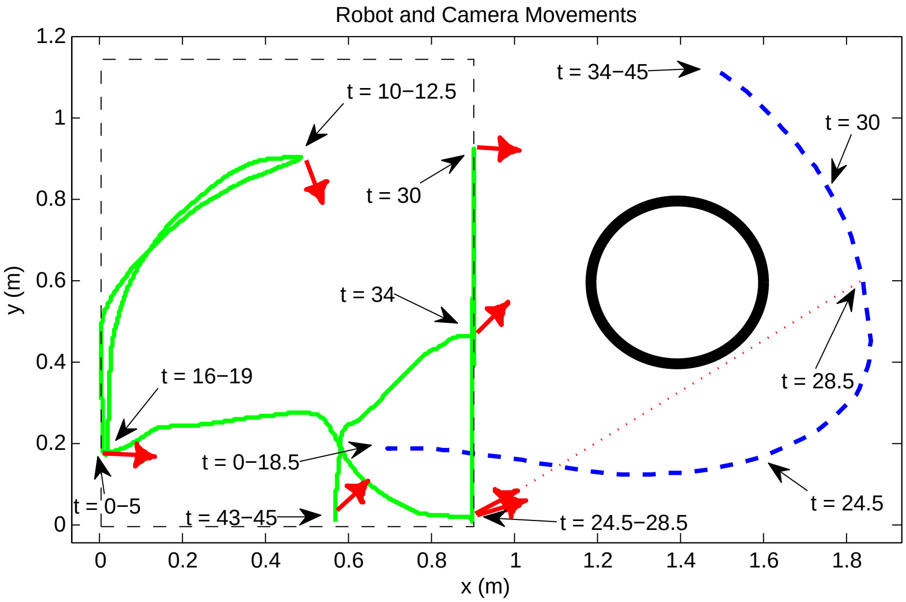


Figure 5.8: Movement of the robot and the camera in the $x - y$ plane. Robot and camera paths are depicted in dashed blue and solid green, respectively. The orientation of the camera is also shown with red arrows at some important times. The dashed rectangle is the boundary of the gantry system workspace and the dotted red line shows that the robot is still visible at $t = 28.5s$.

Fig. 5.10, these commands result in the camera moving backward and upward from $t = 40s$ to $t = 43s$. The experiment ends at $t = 45s$.

5.3.2 Human Factors Experiments

The goal of the next set of experiments was to evaluate the effectiveness of the camera view planner from a system usability perspective. The need for camera control and viewpoint alteration mostly arises in tasks which involve substantial movements on the slave side. Large robotic arms performing manipulation tasks and almost all mobile robots performing navigation tasks fall into this category. The experiments in this paper focused on a navigation task with a mobile robot. Fig. 5.11 shows the full

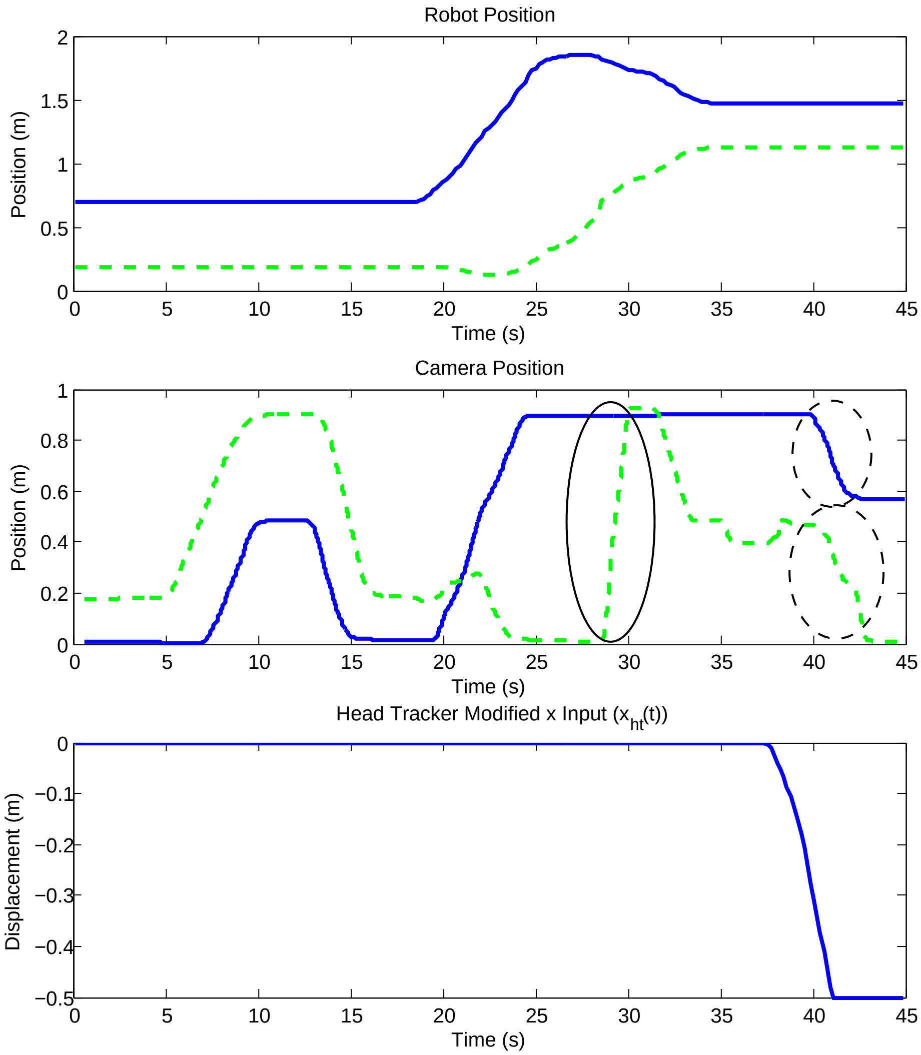


Figure 5.9: Top: x and y coordinates of the robot with respect to time in solid blue and dashed green. Middle: x and y coordinates of the camera with respect to time in solid blue and dashed green. The solid circle demonstrates the fast movement of camera to cross over the obstacle and the dashed circles highlight the effect of head translational movements on the position of the camera. Bottom: The x component of the modified head tracker translational input, $x_{ht}(t)$ saturated at -0.5. The y and z components are zero throughout the experiment, hence they are not shown.

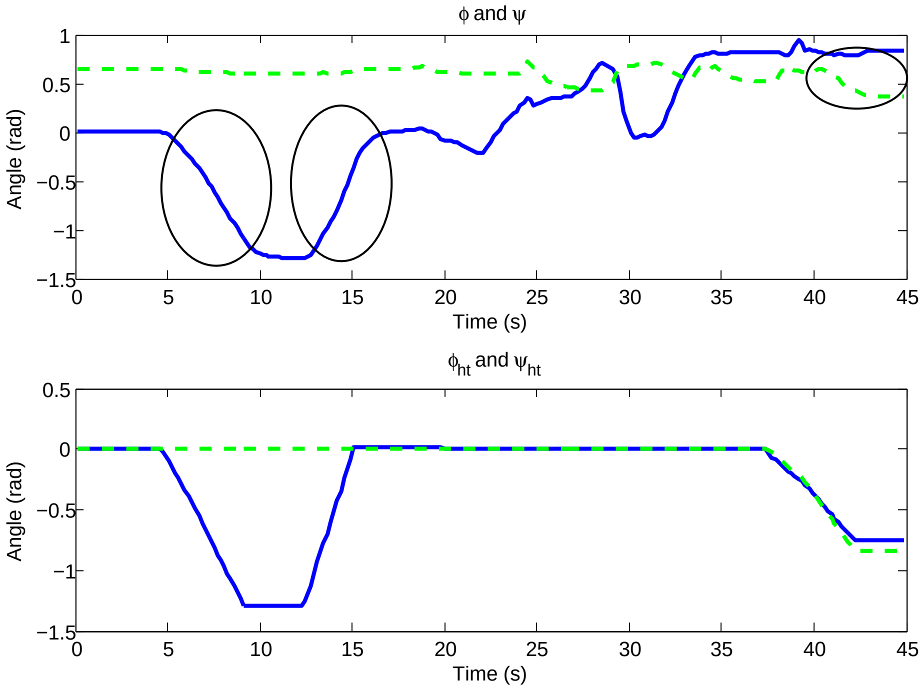


Figure 5.10: Pan and tilt rotations of the camera (top) and head tracker rotational inputs (bottom). Pan and tilt rotations are depicted in solid blue and dashed green, respectively. The circles highlight the effect of head tracker rotational inputs on the orientation of the camera.

workspace of task as well as the workspace of the camera positioning system. First, the participants were asked to maneuver the robot out of a very narrow corridor only 48 cm wide. They were instructed to inspect the environment next to identify the two numbers at the corner of the two passages, after which they were supposed to navigate the robot through the passage (58 cm wide) with the higher number and park the robot inside the green rectangle (60 cm long and 50 cm wide). The numbers were placed randomly and the participants did not have direct view of the field. Finally, each participant was required to report the number associated with the parking space (the one inside the green rectangle). Considering the mobile robot dimensions of 44.5 cm long and 38 cm wide, the task required navigation through confined spaces as well as full inspection of the environment. Note that because of the obstacle in front of the two parking spaces, the numbers inside them were not visible unless the camera was moved to the very corner of its workspace.

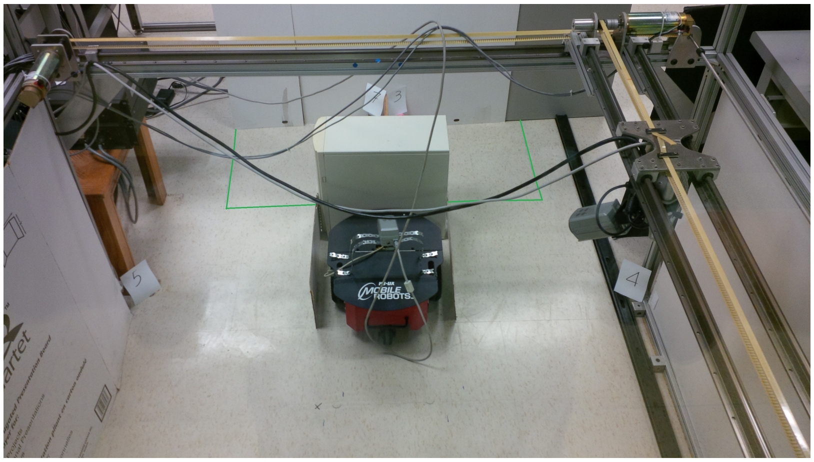


Figure 5.11: A bird's eye view of the task-space in human factors experiment. The computer case in the middle serves as an obstacle to occlude the view of the two numbers inside the parking spaces.

The participants were asked to complete this task using five different viewpoint controllers. The proposed automatic controller with Adjusted Frames (AF) for the camera rotations in the teleoperation controller (see the discussion at the end of Section 5.2.2) and with fixed frames (FF), a conventional manual controller using extra joystick and knobs to position the camera wherever the operator desires, again using adjusted or fixed frames, and finally an egocentric viewpoint using the fixed camera on the robot instead of the bird's eye view camera. Note that these different controllers were presented to the participants in random order to cancel any order bias.

The experiment setup and control parameters were exactly the same as those described in the previous subsection. An extra Quanser 3-DOF planar pantograph was used for the manual viewpoint controller; the 2 translational DOF were mapped to the movements of the gantry system and the rotational DOF was used to control the pan rotation of the camera. Furthermore, an additional knob (potentiometer) was mounted on the haptic device to let the operator control the tilt angle. Twelve (12) participants volunteered for the experiment. The participants had little to no background in teleoperation and ranged between 23 to 27 years of age. The performance measure used was TCT. The subjects were penalized for each collision with 30 seconds added to their recorded time. Given their lack of experience in teleoperation, participants were briefly trained individually prior to performing the task.

The results of the experiment are presented in Table 5.2 where the ± values represent standard error of the mean. Fig. 5.12 also shows a diagram of the composite scores achieved with each camera control method. Statistical analysis was performed by the Fisher's LSD test using a Repeated Measures ANOVA with a significance level of $\alpha = 0.05$ for all cases, which indicated significant difference between different viewpoint controllers (probability of $P = 6 \times 10^{-7}$ for the null-hypothesis). The analysis was followed by protected t tests. The first test revealed that adjusting the mapping between master/slave coordinate frames to preserve eye-hand coordination is significantly advantageous to using fixed frames (probability of $P = 4 \times 10^{-5}$ for the null-hypothesis). The next tests showed that the proposed view planner yields a significantly better composite score compared to both manually controlled (probability of $P = 0.0036$ for the null-hypothesis) and egocentric (probability of $P = 0.002$ for the null-hypothesis) viewpoints.

Table 5.2: Human factor experiment results.

Camera Control Method	TCT (s)	Number of collisions	Composite score
Automatic-AF	56.9 ± 5.9	0.08 ± 0.08	59.4 ± 6.6
Automatic-FF	80.2 ± 9.2	0.17 ± 0.1	85.2 ± 9.9
Manual-AF	88.7 ± 8.7	0.08 ± 0.08	91.2 ± 8
Manual-FF	124.3 ± 11.5	0.25 ± 0.18	131.8 ± 13.8
Egocentric	70.75 ± 5.9	0.75 ± 0.3	93.25 ± 12.9

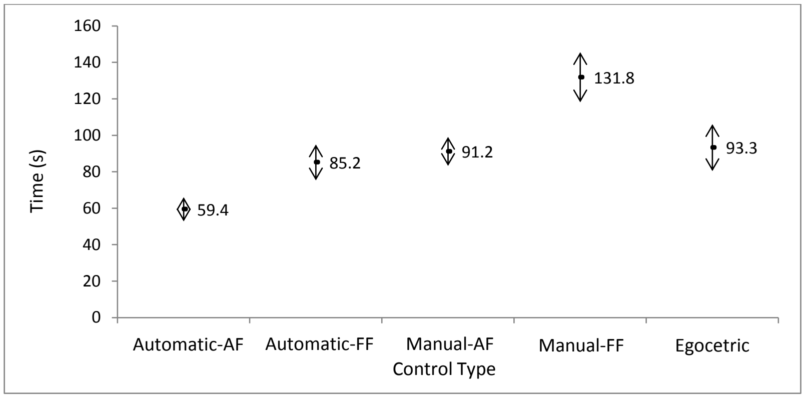


Figure 5.12: Test results of the volunteers performing the task using the proposed automatic viewpoint controller, manual controller, and the egocentric camera. The arrows are the standard error of the mean (SEM).

All participants expressed dissatisfaction with the FF control methods and mentioned that they required a great deal of mental activity to carry out the task. Another notable observation was that when using the manual control methods, most participants tended to completely stop the teleoperation task in order to reposition the camera. While such behaviour is expected in teleoperation tasks requiring both hands, it is interesting to note that the teleoperation task of these experiments only needed one hand. This observation suggests that the cognitive load of camera repositioning is sufficiently high to require operators full attention. Finally, the results in Table 5.2 indicate that the number of collisions using the egocentric viewpoint was drastically higher compared to other methods. The reason for this is that the operator cannot see the full body of the robot which makes navigation in confined environments extremely hard.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This thesis presented novel control approaches for autonomous control of subtasks in a teleoperation system involving mobile robotic manipulators. The focus has been on defining subtasks which are critical to the success of the teleoperation mission, but they could also be carried out autonomously to abate the complexities of the overall task from the operator's perspective. Two common difficult problems in teleoperation were investigated in which the subtasks were defined such that they reduce the cognitive workload of the operator.

A semi-autonomous teleoperation controller was designed for a mobile manipulator in which the arm end-effector(s) and mobile base need to be both controlled. The proposed controller employed autonomous subtasks with shared priority with teleoperation task in order to allow the operator mainly focus on the teleoperation aspect of the task. A constrained optimization problem was formulated and solved that utilizes the redundant DOM to position the arms and mobile base in a desired configuration. The results of the optimization guide the autonomous subtasks in charge of internal configuration control. Experimental results showed that the proposed controller performs as expected and significantly reduces task completion time compared to the case when the operator has to control both the base and the arms, under an ego-centric viewpoint.

In the second problem investigated, a solution was provided for autonomous control of camera viewpoint during teleoperation of mobile robots based on an optimal model-predictive control philosophy. The proposed controller ensures that all important points for the teleoperation are included in the video feed while the operator can seamlessly rotate and/or move the camera with head motions, if needed. Motion constraints of the camera positioning system and smoothness of camera trajectory were taken into consideration for camera view planning. A constrained optimization problem was formulated and solved which determines the best configuration for the camera in a fixed time frame ahead and produces a trajectory for the camera to smoothly reach that configuration. The proposed controller was implemented and tested in teleoperation experiments with a mobile robot; the results indicated that it performs as expected. The results also showed some of the key features of the proposed controller such as its predictive nature, the mechanism by which it avoids obstacles obscuring the video, how the head tracker commands are used as a natural interface for the operator to control the camera, and how the controller deals with infeasible scenarios. Human factor studies were also conducted, which showed that the proposed viewpoint controller is more user friendly compared to manually controlling the camera position, since it imposes less cognitive load on the operator.

6.2 Future Work

This thesis presented a novel approach to reduce the complexities of operating in a teleoperation environment. While the results from the methods proposed in this thesis are very encouraging, there are still unexplored possibilities that could lead to further improvements. Some possible directions for future research are discussed below.

- The two problems addressed in this thesis could be combined into a single optimization problem. Unifying the two optimization problems into one would give rise to interesting questions regarding the interaction between the robot and the camera's movement and its implications for problem formulation and system performance.
- In the mobile manipulator control problem, the optimization problem could be modified to include other objectives such as obstacle avoidance.
- The current work in the autonomous viewpoint planning relies on wheel odometry to estimate the current position and velocity of the robot. It also assumes that the positions of the obstacles in the workspace are known, or estimated through a separate process. In future, image processing algorithms could be integrated into this control framework so that the obstacles could be automatically detected. Furthermore, using video processing and visual tracking could increase the robustness of the controller to errors in wheel odometry, e.g. due to the robot slipping on a surface.
- View planning controller could be further improved by considering moving obstacles in the formulation. The current formulation involves only stationary

objects.

- As the results of the human factor study in Chapter 4 revealed, ignoring the zoom capability of a bird's eye view camera could result in poor performance of an operator using this viewpoint. In the viewpoint planning problem, the zooming feature of the camera system could also be integrated in the optimization formulation. This could be especially helpful in situations where the workspace of the camera positioning system is limited.

- One other interesting subject for future work is to consider robust optimization/control in each case under uncertainty in the system models, e.g., in robot/operator movements and also the location of the obstacles.

Appendix A

Parabolic Spline Interpolation

The optimization problem formulation in (5.6) considers the x, y, z, ϕ, and ψ coordinates of the internal camera positions ($P_{c_{Ij}}$) as optimization variables. Using the values of these variables during the optimization, the workspace trajectory planning routine designs a trajectory for the camera at each iteration.

Using the parabolic spline interpolation method, the desired workspace trajectory will consist of $m + 2$ parabolic functions described below:

$$\Gamma(t) = \begin{cases} a_0 t^2 + b_0 t + c_0 \text{ for } 0 < t \leq \xi_1 \\ a_j t^2 + b_j t + c_j \text{ for } \xi_j < t \leq \xi_{j+1}, j \in [1, m] \\ a_{m+1} t^2 + b_{m+1} t + c_{m+1} \text{ for } \xi_{m+1} < t \leq T_m \end{cases} \tag{A.1}$$

where

$$
\begin{aligned}
\xi_1 &= \frac{T_{I_1}}{2} \\
\xi_j &= \frac{T_{I_j} + T_{I_{j-1}}}{2} \text{ for } j \in [2, m] \\
\xi_{m+1} &= \frac{T_{I_m} + T_m}{2}
\end{aligned}
\tag{A.2}
$$

Γ denotes any of the variables x, y, z, ϕ, or ψ. The above trajectory is designed for each DOF of the camera separately. One can easily show that by writing down the position and velocity continuity equations at ξ_1 to ξ_{m+1}, the position equations at times $0, T_{I_1}, \cdots, T_{I_m}, T_m$, setting the initial velocity to current velocity read through feedback, and setting the final velocity equal to 0, the $3 \times (m + 2)$ coefficients of the parabolic splines are uniquely specified. Arranging the above mentioned equations in matrix format, the following matrix equation is obtained:

$$\underbrace{\left[\begin{array}{cccccccccccc}
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
\hdashline
\xi_1^2 & \xi_1 & 1 & -\xi_1^2 & -\xi_1 & -1 & 0 & 0 & 0 & 0 & \cdots & 0 \\
2\xi_1 & 1 & 0 & -2\xi_1 & -1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & T_{I_1}^2 & T_{I_1} & 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\
\hdashline
0 & 0 & 0 & \xi_2^2 & \xi_2 & 1 & -\xi_2^2 & -\xi_2 & -1 & 0 & \cdots & 0 \\
0 & 0 & 0 & 2\xi_2 & 1 & 0 & -2\xi_2 & -1 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & T_{I_2}^2 & T_{I_2} & 1 & 0 & \cdots & 0 \\
\hdashline
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & & & & \vdots \\
\hdashline
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & T_m^2 & T_m & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 2T_m & 1 & 0
\end{array}\right]}_{3(m+2)\times 3(m+2)}
\underbrace{\left[\begin{array}{c}
a_0 \\ b_0 \\ \hdashline c_0 \\ a_1 \\ b_1 \\ \hdashline c_1 \\ a_2 \\ b_2 \\ \hdashline \vdots \\ \hdashline b_{m+1} \\ c_{m+1}
\end{array}\right]}_{3(m+2)\times 1}
=
\underbrace{\left[\begin{array}{c}
q_{c_0} \\ \dot{q}_{c_0} \\ \hdashline 0 \\ 0 \\ q_{c_{I1}} \\ \hdashline 0 \\ 0 \\ q_{c_{I2}} \\ \hdashline \vdots \\ \hdashline q_c \\ 0
\end{array}\right]}_{3(m+2)\times 1}
\tag{A.3}$$

where q is replaced by x, y, z, ϕ, or ψ for calculating the trajectory of each degree of freedom of the camera positioning system. The first two lines are derived from initial position and velocity constraints, the next three-line sets are the position and velocity continuity constraints at ξ_j and position constraint at T_{I_j}, the final two constraints indicate the final position and velocity. Note that as long as the time intervals are constant, the square matrix on the left hand side is fixed and can be inverted offline in order to calculate the parabolic splines coefficients based on the optimization variables.

Bibliography

Abbott, J. J., Marayong, P., and Okamura, A. M. (2007). Haptic virtual fixtures for robot-assisted manipulation. In *Robotics research*, pages 49–64. Springer.

Arimoto, S., Hashiguchi, H., Sekimoto, M., and Ozawa, R. (2005). Generation of natural motions for redundant multi-joint systems: A differential-geometric approach based upon the principle of least actions. *Journal of Robotic Systems*, **22**(11), 583–605.

Boggs, P. T. and Tolle, J. W. (1995). Sequential quadratic programming. *Acta numerica*, **4**(1), 1–51.

Brooks, B. G. and McKee, G. T. (2001). The visual acts model for automated camera placement during teleoperation. In *IEEE International Conference on Robotics and Automation, 2001.*, volume 1, pages 1019–1024. IEEE.

Buss, M., Lee, K. K., Nitzsche, N., Peer, A., Stanczyk, B., and Unterhinninghofen, U. (2007). Advanced telerobotics: Dual-handed and mobile remote manipulation. *Advances in Telerobotics*, pages 471–497.

Cândido, C., Santana, P., Correia, L., and Barata, J. (2008). Shared control of a pan-tilt camera on an all-terrain mobile robot. In *IEEE International Conference* *on Emerging Technologies and Factory Automation. ETFA 2008.*, pages 177–183. IEEE.

Chen, J. Y., Haas, E. C., and Barnes, M. J. (2007). Human performance issues and user interface design for teleoperated robots. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews.*, **37**(6), 1231–1245.

Corde Lane, J., Carignan, C. R., Sullivan, B. R., Akin, D. L., Hunt, T., and Cohen, R. (2002). Effects of time delay on telerobotic control of neutral buoyancy vehicles. In *IEEE International Conference on Robotics and Automation, 2002. Proceedings. ICRA'02.*, volume 3, pages 2874–2879. IEEE.

Dai, L. and Li, Z. (2010). Design and implementation of motion control for teleoperated mobile manipulators. In *8th World Congress on Intelligent Control and Automation (WCICA).*, pages 6691–6696. IEEE.

Das, H., Sheridan, T. B., and Slotine, J. J. (1989). Kinematic control and visual display of redundant teleoperators. In *IEEE International Conference on Systems, Man and Cybernetics, 1989. Conference Proceedings.*, pages 1072–1077. IEEE.

De Boor, C. (1978). *Practical Guide to Splines.* Springer-Verlag.

Diolaiti, N. and Melchiorri, C. (2002). Teleoperation of a mobile robot through haptic feedback. In *Haptic Virtual Environments and Their Applications, IEEE International Workshop 2002 HAVE*, pages 67–72. IEEE.

Ferre, M., Buss, M., Aracil, R., Melchiorri, C., and Balaguer, C. (2007). *Advances in telerobotics.* Springer Publishing Company, Incorporated.

Fournier, J., Mokhtari, M., and Ricard, B. (2011). Immersive virtual environment for mobile platform remote operation and exploration. In *IEEE International Symposium on Robotic and Sensors Environments (ROSE), 2011.*, pages 37–42. IEEE.

Freedman, L., Crooks, W., and Coan, P. (1977). TV requirements for manipulation in space. *Mechanism and machine theory*, **12**(5), 425–438.

Frejek, M. and Nokleby, S. (2013). A methodology for tele-operating mobile manipulators with an emphasis on operator ease of use. *Robotica*, **1**(1).

Funda, J., Taylor, R., Eldridge, B., Gomory, S., and Gruben, K. (1996). Constrained cartesian motion control for teleoperated surgical robots. *IEEE Transactions on Robotics and Automation*, **12**(3), 453–465.

Goel, M., Maciejewski, A., Balakrishnan, V., and Proctor, R. W. (2003). Failure tolerant teleoperation of a kinematically redundant manipulator: an experimental study. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, **33**(6), 758–765.

Hayashi, K., Yokokohji, Y., and Yoshikawa, T. (2005). Tele-existence vision system with image stabilization for rescue robots. In *IEEE International Conference on Robotics and Automation. ICRA 2005.*, pages 50–55. IEEE.

Hayhoe, M. and Ballard, D. (2005). Eye movements in natural behavior. *Trends in cognitive sciences*, **9**(4), 188–194.

Held, M. (1997). ERIT a collection of efficient and reliable intersection tests. *Journal of Graphics Tools*, **2**(4), 25–44.

Hishinuma, T. and Nenchev, D. (2006). Singularity-consistent vibration suppression control with a redundant manipulator mounted on a flexible base. In *IEEE/RSJ International Conference on Intelligent Robots and Systems.*, pages 3237–3242.

Hokayem, P. F. and Spong, M. W. (2006). Bilateral teleoperation: An historical survey. *Automatica*, **42**(12), 2035–2057.

Hollands, J. and Lamb, M. (2011). Viewpoint tethering for remotely operated vehicles effects on complex terrain navigation and spatial awareness. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, **53**(2), 154–167.

Hollands, J. and Wickens, C. (1999). *Engineering psychology and human performance.* Prentice Hall New Jersey.

Howell, D. C. (2004). *Fundamental Statistics for the Behavioral Sciences.* Brooks/Cole, 5th edition.

Hsu, P., Mauser, J., and Sastry, S. (1989). Dynamic control of redundant manipulators. *Journal of Robotic Systems*, **6**(2), 133–148.

Hsu, S. C., Liang, S. F., Fan, K. W., and Lin, C. T. (2007). A robust in-car digital image stabilization technique. *IEEE Transactions on Cybernetics*, **37**(2), 234–247.

Hughes, S. B. and Lewis, M. (2005). Task-driven camera operations for robotic exploration. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans.*, **35**(4), 513–522.

Hwang, D. Y. and Hannaford, B. (1998). Teleoperation performance with a kinematically redundant slave robot. *The International Journal of Robotics Research*, **17**(6), 579–597.

Jin, J., Zhu, Z., and Xu, G. (2000). A stable vision system for moving vehicles. *IEEE Transactions on Intelligent Transportation Systems.*, **1**(1), 32–39.

Johansson, R., Westling, G., Bäckström, A., and Flanagan, J. (2001). Eye–hand coordination in object manipulation. *the Journal of Neuroscience*, **21**(17), 6917–6932.

Kaufman, A. E., Bandopadhay, A., and Shaviv, B. D. (1993). An eye tracking computer user interface. In *IEEE 1993 Symposium on Research Frontiers in Virtual Reality.*, pages 120–121. IEEE.

Keyes, B., Casey, R., Yanco, H. A., Maxwell, B. A., and Georgiev, Y. (2006). Camera placement and multi-camera fusion for remote robot operation. In *Proceedings of the IEEE International Workshop on Safety, Security and Rescue Robotics*, pages 22–24.

Khatib, O. (1999). Mobile manipulation: The robotic assistant. *Robotics and Autonomous Systems*, **26**(2-3), 175 – 183.

Laviola, J. J. (2008). Bringing VR and spatial 3D interaction to the masses through video games. *IEEE Computer Graphics and Applications*, **28**(5), 10–15.

Lee, S., Sukhatme, G., Kim, G. J., and Park, C. M. (2005). Haptic teleoperation of a mobile robot: a user study. *Presence: Teleoperators and Virtual Environments*, **14**(3), 345–365.

Malysz, P. (2011). *A Kinematic Control Framework for Asymmetric Semi-autonomous Teleoperation Systems*. Ph.D. thesis, McMaster University.

Malysz, P. and Sirouspour, S. (2011a). A kinematic control framework for single-slave asymmetric teleoperation systems. *IEEE Transactions on Robotics*, **27**(5), 901–917.

Malysz, P. and Sirouspour, S. (2011b). A task-space weighting matrix approach to semi-autonomous teleoperation control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2011*, pages 645–652. IEEE.

Malysz, P. and Sirouspour, S. (2013). Mixed autonomous/teleoperation control of asymmetric robotic systems. *submitted for publication to IEEE Transactions on Automatic Control*.

Marín, R., Sanz, P. J., and Del Pobil, A. P. (2003). The UJI online robot: an education and training experience. *Autonomous Robots*, **15**(3), 283–297.

Martins, H. and Ventura, R. (2009). Immersive 3-D teleoperation of a search and rescue robot using a head-mounted display. In *IEEE Conference on Emerging Technologies & Factory Automation, 2009. ETFA 2009.*, pages 1–8. IEEE.

Mitsugami, I., Ukita, N., and Kidode, M. (2005). Robot navigation by eye pointing. In *Entertainment Computing-ICEC 2005*, pages 256–267. Springer.

Mukhopadhyay, S. and Smith, B. (1999). Passive capture and structuring of lectures. In *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, pages 477–487. ACM.

Murphy, R. R. and Burke, J. L. (2005). Up from the rubble: Lessons learned about HRI from search and rescue. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 49, pages 437–441. SAGE Publications.

Murphy, R. R., Casper, J., Micire, M., and Hyams, J. (2000). Mixed-initiative control of multiple heterogeneous robots for urban search and rescue. Tech. Rep., Univ. South Florida.

Nagatani, K., Hirayama, T., Gofuku, A., and Tanaka, Y. (2002). Motion planning for mobile manipulator with keeping manipulability. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002.*, volume 2, pages 1663 – 1668.

Nakanishi, J., Cory, R., Mistry, M., Peters, J., and Schaal, S. (2005). Comparative experiments on task space control with redundancy resolution. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005. (IROS 2005)*, pages 3901–3908.

Nanayakkara, D., Kiguchi, K., Murakami, T., Watanabe, K., and Izumi, K. (2001). Skillful adaptation of a 7-DOF manipulator to avoid moving obstacles in a teleoperated force control task. In *IEEE International Symposium on Industrial Electronics, Proceedings. ISIE 2001.*, volume 3, pages 1982 –1987.

Nath, N., Tatlicioglu, E., and Dawson, D. (2008). Teleoperation with kinematically redundant robot manipulators with sub-task objectives. In *47th IEEE Conference on Decision and Control, 2008. CDC 2008.*, pages 4320–4325.

Olmos, O., Wickens, C. D., and Chudy, A. (2000). Tactical displays for combat awareness: An examination of dimensionality and frame of reference concepts and the application of cognitive engineering. *The International Journal of Aviation Psychology*, **10**(3), 247–271.

Park, J. and Khatib, O. (2006). A haptic teleoperation approach based on contact force control. *The International Journal of Robotics Research*, **25**(5-6), 575–591.

Ricks, B., Nielsen, C. W., and Goodrich, M. A. (2004). Ecological displays for robot interaction: A new perspective. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004.(IROS 2004).*, volume 3, pages 2855–2860. IEEE.

Romano, J. M., Webster, R., and Okamura, A. M. (2007). Teleoperation of steerable needles. In *IEEE International Conference on Robotics and Automation.*, pages 934–939. IEEE.

Rubi, J., Rubio, A., and Avello, A. (2002). Involving the operator in a singularity avoidance strategy for a redundant slave manipulator in a teleoperated application. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002.*, volume 3, pages 2973 – 2978.

Rui, Y., Gupta, A., Grudin, J., and He, L. (2004). Automating lecture capture and broadcast: technology and videography. *Multimedia Systems*, **10**(1), 3–15.

Sanchez, E., Rubio, A., and Avello, A. (2002). An intuitive force feed-back to avoid singularity proximity and workspace boundaries in bilateral controlled systems based on virtual springs. In *IEEE/RSJ International Conference on Intelligent Robots and Systems.*, volume 2, pages 1302–1307. IEEE.

Scholtz, J., Young, J., Drury, J., and Yanco, H. (2004). Evaluation of human-robot interaction awareness in search and rescue. In *IEEE International Conference on Robotics and Automation.*, volume 3, pages 2327 – 2332 Vol.3.

Sciavicco, L. and Siciliano, B. (2000). *Modelling and control of robot manipulators*. Springer, second edition.

Seraji, H. (1998). A unified approach to motion control of mobile manipulators. *The International Journal of Robotics Research*, **17**(2), 107–118.

Shahdi, A. and Sirouspour, S. (2009). Adaptive/robust control for time-delay teleoperation. *IEEE Transactions on Robotics*, **25**(1), 196–205.

Sheridan, T. B. (1989). Telerobotics. *Automatica*, **25**(4), 487–507.

Shimizu, M., Kakuya, H., Yoon, W. K., Kitagaki, K., and Kosuge, K. (2008). Analytical inverse kinematic computation for 7-DOF redundant manipulators with joint limits and its application to redundancy resolution. *IEEE Transactions on Robotics*, **24**(5), 1131–1142.

Shin, D. H., Hamner, B. S., Singh, S., and Hwangbo, M. (2003). Motion planning for a mobile manipulator with imprecise locomotion. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003.(IROS 2003)*, volume 1, pages 847–853. IEEE.

Shiroma, N., Kobayashi, J., and Oyama, E. (2009). Compact image stabilization system for small-sized humanoid. In *IEEE International Conference on Robotics and Biomimetics, 2008. ROBIO 2008.*, pages 149–154.

Siciliano, B. (1990). Kinematic control of redundant robot manipulators: A tutorial. *Journal of Intelligent and Robotic Systems*, **3**(3), 201–212.

Sirouspour, S. (2005). Modeling and control of cooperative teleoperation systems. *IEEE Transactions on Robotics*, **21**(6), 1220–1225.

Sitti, M., Aruk, B., Shintani, H., and Hashimoto, H. (2003). Scaled teleoperation system for nano-scale interaction and manipulation. *Advanced Robotics*, **17**(3), 275–291.

Talamini, M., Chapman, S., Horgan, S., and Melvin, W. (2003). A prospective analysis of 211 robotic-assisted surgical procedures. *Surgical Endoscopy and Other Interventional Techniques*, **17**(10), 1521–1524.

Wang, S., Xiong, X., Xu, Y., Wang, C., Zhang, W., Dai, X., and Zhang, D. (2006). Face-tracking as an augmented input in video games: enhancing presence, role-playing and control. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1097–1106. ACM.

Yanco, H. A. and Drury, J. (2004). Where am I? Acquiring situation awareness using a remote robot platform. In *IEEE International Conference on Systems, Man and Cybernetics.*, volume 3, pages 2835–2840. IEEE.

Zhu, D., Gedeon, T., and Taylor, K. (2011). Exploring camera viewpoint control models for a multi-tasking setting in teleoperation. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, pages 53–62. ACM.