#### TELEOPERATION OF MOBILE MANIPULATORS

By

Yunyi Jia

#### A DISSERTATION

Submitted to Michigan State University in partial fulfillment of the requirements for the degree of

Electrical Engineering - Doctor of Philosophy

2014

#### ABSTRACT

#### TELEOPERATION OF MOBILE MANIPULATORS

#### By

#### Yunyi Jia

Mobile manipulators provide larger working spaces and more flexibility than standard manipulators by introducing mobility. Through teleoperation, they can be applied to a variety of areas such as hazardous material handling, outer space exploration, searching and rescue, etc.

Inspired by application requirements, there are four major challenges in the teleoperation of mobile manipulators including the modeling and control of mobile manipulators, teleoperation of multiple mobile manipulators, modeling the human teleoperator in teleoperation system and communications between the human teleoperator and mobile manipulators. Therefore, this study aims to address these challenges.

For the modeling and control of mobile manipulators, the motion accuracy of the endeffector is a problem for the existing methods due to the system performance differences. To address this issue, we introduce a new control method with online motion distribution and coordination to improve the accuracy. In addition, a sensor-based redundancy resolution scheme is proposed to further improve the teleoperation efficiency.

For the teleoperation of multiple mobile manipulators, the system stability under random communication delays and unexpected events is a major problem for the existing methods. To address this issue, we propose a non-time based teleoperation and coordination method. A non-time perceptive reference is designed as the new reference to replace the time in the system modeling and control. Through this design, the system stability under random communication delays and unexpected events could be ensured.

For modeling the human teleoperator in teleoperation system, there are no existing models and the teleoperation efficiency and safety are always subject to the operation status of the teleoperator. To address this issue, we propose a concept named quality of teleoperator (QoT) to represent the teleoperator and incorporate it into the modeling and control of the teleoperation system. Through this design, the teleoperation efficiency and safety could be improved under various operation status of the teleoperator.

For the communications between the human teleoperator and mobile manipulators, the existing methods of using joysticks are neither efficient nor intuitive. Therefore, we introduce the natural language as a new communication manner. However, the existing natural language control methods could not online handle unexpected events in the environment and robotic system. To address this issue, a new systematic natural language modeling and control method is designed to online handle such unexpected events.

Finally, the proposed methods are all implemented on our developed mobile manipulators and the experimental results illustrate their effectiveness and advantages. Copyright by YUNYI JIA 2014 To the ones in my backing.

#### ACKNOWLEDGMENTS

Foremost, I would like to thank and express my appreciation and gratitude to my advisor Dr. Ning Xi. Without his guidance, it would not have been possible for me to finish this dissertation. He has helped to shape this dissertation research with his insight knowledge and numerous efforts. He has also helped to shaped me to be a qualified researcher.

In addition, I would like to thank my dissertation committee members: Dr. Matt Mutka, Dr. Hassan Khalil and Dr. Fathi Salem. I greatly appreciate their valuable feedback and discussions throughout the entire dissertation process.

I would also like to thank our collaborators at Michigan State University: Dr. Joyce Y. Chai and her students Lanbo She, Rui Fang, Changsong Liu and Shaohua Yang and our collaborators at City University of Hong Kong: Dr. Shuang Liu and Dr. Sheng Bi for their plentiful support and efforts on the experiments and system designs.

My many thanks go to my lab members: Jianguo Zhao, Yu Cheng, Dr. Yong Liu, Dr. Bingtuan Gao, Dr. Jing Xu, Dr. Yong Jiang, Dr. King W.C. Lai, Dr. Carmen K.M. Fung, Dr. Chi Zhang, Dr. Hongzhi Chen, Dr. Erick Nieves, Dr. Ruiguo Yang, Bo Song, Liangliang Chen, Hai Wang, Philipp Sturmer, Yunxia Wang, Xin Li, Qin Zhang, Huatao Zhang, Gokhan Erdemir, Liang Sun, Weixian Li, Zhiyong Sun, John Gregory, John Buether, Siyang Liang, Dr. Qizhi Wang, Dr. Chengzhi Su, Dr. Fei Wang, Dr. Biqiang Du, Dr. Zhenxue Chen, Dr. Feng Chen, Dr. Daoxiong Gong, Dr. Jizhan Liu, Dr. Xiaoyong Wang, Dr. Jianyong Lou, Dr. Jianlong Zou, Dr. Zhanxin Zhou and all other professors and students who have worked in our lab. They have provided me a lot of technical and motivational support during all stages of this dissertation.

Finally, I have to thank my beloved family including my parents, my sisters and my

brothers in law and my fiancee Bin Guo. Their selfless love and support have always been my strong backing in my entire study and life.

## TABLE OF CONTENTS

LIST (	IST OF TABLES	
LIST (	OF FIGURES	xii
Chapte	er 1 Introduction	1
1.1	Background and Motivations	1
1.2	Challenges and Difficulties	3
1.3	Literature Review	5
	1.3.1 Review on Modeling and Control of Mobile Manipulators	5
	1.3.2 Review on Teleoperation of Multiple Robotic Systems	8
	1.3.3 Review on Modeling the Human Teleoperator in Teleoperation	10
	1.3.4 Review on Communications between Human Teleoperator and Robots	13
1.4	Objectives of the Study	16
1.5	Organization of the Dissertation	16
Chapte	er 2 Modeling and Control of Nonholonomic Mobile Manipulators	
	for Teleoperation	18
2.1	Introduction	18
2.2	Whole-Body Modeling and Control of Mobile Manipulator in Teleoperation .	19
	2.2.1 Whole-Body Modeling of Mobile Manipulator	19
	2.2.2 Control with Motion Distribution	22
	2.2.3 Control with Motion Coordination	24
2.3	Sensor based Redundancy Resolution for Obstacle Avoidance in Teleoperation	29
	2.3.1 Sensor based Redundancy Resolution Structure	29
	2.3.2 Sensor System Design	30
2.4	2.3.3 Sensor based Online Redundancy Resolution for Obstacle Avoidance	36
2.4	Experimental Results	38
	2.4.1 Results of Whole-Body Control	38
0.5	2.4.2 Results of Sensor-based Redundancy Resolution	42
2.5	Chapter Summary	47
Chapte	er 3 Teleoperation of Multiple Mobile Manipulators	49
3.1	Introduction	49
3.2	Mobile Manipulator Teleoperation System	50
	3.2.1 Perceptive Planning and Control	50
	3.2.2 Mobile Manipulator Teleoperation Design	52
3.3	Multi-Mobile-Manipulator Teleoperation System	55
	3.3.1 Single-Operator-Multi-Robot Teleoperation System	55
	3.3.2 Multi-Robot Teleoperation System in the Perceptive Reference Frame	57

	3.3.3 Design of Single-Operator-Multi-Robot Teleoperation System	60
3.4	Experimental Results	65
	3.4.1 Experimental Setup	65
	3.4.2 Experimental Results	65
3.5	Chapter Summary	69
Chapte	er 4 Teleoperation of Mobile Manipulators Adaptive to Quality of	
onapo	Teleoperator (QoT)	73
4.1	Introduction	73
4.2	Teleoperation System Adaptive to QoT	74
	4.2.1 Quality of Teleoperator (QoT)	74
	4.2.2 Architecture for QoT based Teleoperation	75
4.3	Online Identification of QoT	76
	4.3.1 QoT Indicators	76
	4.3.2 Model for QoT Identification	77
	4.3.3 Training for QoT Identification Model	80
4.4	QoT based Adaptation for Telerobotic Operations	81
	4.4.1 QoT based Adaptation	81
	4.4.2 Performance Analysis and Discussions	85
4.5	Experimental Results	87
	4.5.1 Results of Online OoT Identification	87
	4.5.2 Results of QoT Adaptive Teleoperation	91
4.6	Chapter Summary	103
Chapte	er 5 Teleoperation of Mobile Manipulators by Natural Language	
	Control	105
5.1	Introduction	105
5.2	Natural Language Controlled Robotic System	106
5.3	Natural Language Processing	107
	5.3.1 Semantic Processing	107
	5.3.2 Graph Matching Process	108
	5.3.3 Graph-Matching based Grounding	112
5.4	Planning and Control for Natural Language Controlled Robotic Systems	114
	5.4.1 Perceptive Abstracted Modeling	114
	5.4.2 Planning of Robotic Systems	116
	5.4.3 Perceptive Planning and Control of Robotic Systems	117
5.5	Stability Analysis	120
5.6	Experimental Results and Analysis	122
	5.6.1 Experimental Implementation	122
	5.6.2 Experimental Results on Natural Language Control	124
	5.6.3 Experimental Results on Handling Unexpected Events	126
5.7	Chapter Summary	130

Chapte	er 6 Development of Mobile Manipulator Systems	31	
6.1	Introduction	31	
6.2	Development of Holonomic Mobile Manipulators	32	
	6.2.1 System hardware Development	32	
	6.2.2 System Software Development	32	
6.3	Development of Nonholonomic Mobile Manipulators	34	
	6.3.1 System hardware Development	34	
	6.3.2 System Software Development	36	
6.4	Chapter Summary 1	38	
Chapte	er 7 Conclusions and Future Work	40	
7.1	Conclusions	40	
7.2	Future Research Work	42	
	7.2.1 Human-Robot Shared Control for Teleoperation	42	
	7.2.2 Human-Robot Dialog for Teleoperation	43	
REFE	REFERENCES		

#### LIST OF TABLES

Table 4.1	Statistical analysis of QoT and corresponding teleoperation perfor- mance	90
Table 4.2	Time cost comparison of using different methods $\ldots \ldots \ldots \ldots$	94
Table 4.3	Time cost comparison in navigation task	99
Table 4.4	Collision comparison in navigation task	100
Table 4.5	Time cost comparison in mobile manipulation task	101
Table 4.6	Comparison of collisions and attempt times in mobile manipulation task	102

## LIST OF FIGURES

Figure 2.1	MSU Nonholonomic Mobile Manipulator	20
Figure 2.2	Sensor based online redundancy resolution	30
Figure 2.3	Laser scanning area	31
Figure 2.4	Coordinate frame	33
Figure 2.5	Mobile manipulator and working environment	33
Figure 2.6	Raw data of laser system	34
Figure 2.7	Modified data of laser system	35
Figure 2.8	Contour extraction of bridge structure	35
Figure 2.9	Obstacle and Forearm are projected on Equipotential Surface	36
Figure 2.10	Distance relation between the link projection and the obstacle pro- jection	37
Figure 2.11	Results of the traditional control	40
Figure 2.12	Results of the proposed control with motion distribution and coordi- nation	41
Figure 2.13	Experimental Setup	42
Figure 2.14	Errors of the traditional control	43
Figure 2.15	Errors of the proposed control with motion distribution and coordination	43
Figure 2.16	Obstacle Distance and Elbow Angle	44
Figure 2.17	Avoidance of stationary obstacles	46
Figure 2.18	Avoidance of moving obstacles	47

Figure 3.1	(a) Traditional planning and control (b) Planning and control in perceptive reference frame	51
Figure 3.2	Single mobile manipulator teleoperation system	53
Figure 3.3	Multiple Mobile Manipulator Teleoperation System	56
Figure 3.4	Non-time based single-operator-multi-robot teleoperation system $\ .$ .	58
Figure 3.5	Architecture of single-operator-multi-robot teleoperation system $\ .$ .	66
Figure 3.6	Single-robot teleoperation: (a) Blind guidance (b) Object manipulation	67
Figure 3.7	Multi-mobile-manipulator teleoperation task	68
Figure 3.8	Teleoperator's velocity commands	70
Figure 3.9	Trajectories of the two robots	70
Figure 3.10	Position errors of the two robots	71
Figure 3.11	Coordination error	71
Figure 3.12	Random time delays	72
Figure 4.1	QoT based teleoperation system framework	75
Figure 4.2	Brain headset for EEG signal acquisition	78
Figure 4.3	QoT generation model	78
Figure 4.4	Training of QoT generation model	81
Figure 4.5	QoT (left) and QoT indicators (right) in a day $\ldots \ldots \ldots \ldots$	89
Figure 4.6	QoT of two teleoperators: a female at MSU (upper) and a male in HK (lower)	89
Figure 4.7	Experimental setup for QoT verification	90
Figure 4.8	QoT identification results of the tracking experiment	92
Figure 4.9	Experimental setup for Internet based teleoperation	93

Figure 4.10	Three teleoperation tasks: (a) manipulation; (b) navigation; (c) mo- bile manipulation	93
Figure 4.11	QoT indicators and QoT in manipulation task $\ldots \ldots \ldots \ldots$	96
Figure 4.12	TDI plot in manipulation task	96
Figure 4.13	Robot movement without the adjustment	97
Figure 4.14	Robot motion with the adjustment using QoT $\ldots$	98
Figure 4.15	Robot motion with the adjustment using QoT and TDI $\ . \ . \ .$ .	99
Figure 4.16	Trajectory comparison: (left) traditional method. (right) proposed method	103
Figure 5.1	Schema of natural language controlled robotic system	106
Figure 5.2	Graph matching process	109
Figure 5.3	Graph-matching based grounding process	114
Figure 5.4	Example of perceptive system model	116
Figure 5.5	Perceptive planning and control	119
Figure 5.6	Experimental setup	123
Figure 5.7	Environmental information by the vision system $\ldots \ldots \ldots \ldots$	124
Figure 5.8	Experimental setup for normal natural language control $\ldots$	125
Figure 5.9	Results of natural language control: no uncertain and unexpected events	126
Figure 5.10	Experiment for unexpected environmental change $\ldots \ldots \ldots$	127
Figure 5.11	Experiment for unexpected object removal	127
Figure 5.12	Experiment for unexpected ambiguity	128
Figure 5.13	Results of natural language control: the robot is moved $\ldots \ldots \ldots$	128
Figure 5.14	Results of natural language control: the object is swapped	128

Figure 5.15	Results of natural language control: the robot is blocked	129
Figure 5.16	Results of handling unexpected robot stops $\ldots \ldots \ldots \ldots \ldots$	129
Figure 6.1	Holonomic mobile manipulators	133
Figure 6.2	Implementation of holonomic mobile manipulators $\ . \ . \ . \ . \ .$	134
Figure 6.3	Nonholonomic mobile manipulators	135
Figure 6.4	Implementation of nonholonomic mobile manipulators	137

# Chapter 1

# Introduction

## **1.1** Background and Motivations

Mobile manipulators integrate the functions of manipulability and mobility and can significantly enlarge the working space of the standard manipulators. Through teleoperation, the mobile manipulators can be operated over a distance. Such systems can be applied to a variety of areas [1][2].

Teleoperation technology has been studied for decades and widely used in many applications where common operations cannot be used. The first modern teleoperation system was developed in 1940's in the Argonne National Laboratory when a master-slave manipulator was developed for chemical and nuclear material handling [3]. Afterwards, with the technology evolution of computers, communications, multi-media and advanced sensors such as force/torque sensor and tactile sensor, many teleoperation systems have been developed and applied to a variety of areas.

The first motivated application of the teleoperation system is for material handling in the areas where human beings are not able to reach such as nuclear stations, hazardous chemical plants, etc. [4][5] These demands in these applications promote the development of various teleoperation technologies including robot motion control, video and audio feedback, force feedback and many others.

Outer-space exploration is also an important application of teleoperation systems [6].

Through teleoperation, the human operators could remotely control the robots to perform different tasks in the outer space. It can significantly reduce the cost of outer-space operations like assembly and maintenance like repairs. More importantly, it can reduce the risk of astronauts' safety issues [7].

Teleoperation has also been applied to the medical applications such as telesurgery [8][9]. It makes the medical experts in the world be able to conduct clinical diagnosis and even physical surgery without being present at the patients' sites. It can greatly save the cost and time by transferring the traditional surgery means to a remote surgery means where the patients and medical experts are far away with each other [10].

Except the applications using manipulators, the teleoperation technologies has also been applied to mobile robots [11, 12, 13]. The human operator can remotely control one or several mobile robots to performance different tasks such as formation, co-transportation and multi-robot exploration [14]. Since the working space of the mobile robots is much larger than the manipulators, the requirements on the feedback are much more complicated such as robot locations, environmental maps, surrounding distances, omnidirectional videos, force feedback, etc. In addition, the kinematic similarity between the master and slave does not exist for the teleoperation of mobile robots. Extra teleoperation designs are required to model and analyze such systems [11].

There are also many other applications where teleoperation has been applied. These applications include nano-manipulation [15], entertainment and education [16], forestry [17], excavation [18], etc.

From the theoretical point of view, the studies on teleoperation have been mainly focused on two major issues: stability and telepresence of the teleoperation systems. The study of stability is mainly to design control methods to ensure the stability of the telerobotic systems in presence of various system constraints such as communication delays, packet losses and unstructured environments. The study of telepresence is mainly to develop teleoperation techniques to enable the teleoperator to control the robots and perceive the remote environment in efficient and visual ways as though sympathetic visage. In some cases like communication delays, these two tasks are conflicting with each other. Therefore, for the specific requirements in different applications, appropriate trade-off between the stability and telepresence is required.

Many complications may arise when studying and applying teleoperation systems to different applications. They include the increasing complexity of the robotic systems such as mobile manipulators and multiple robotic systems, communication network constraints such as time delays and packet losses, uncertainties of the human teleoperator such as fatigue and sleepiness and more intuitive control means of teleoperation such as the natural language control. These issues have motivated the research of this dissertation for studying the stable and efficient teleoperation of mobile manipulators.

## **1.2** Challenges and Difficulties

In the teleoperation of mobile manipulators, inspired by different application requirements, there are four major challenges. These challenges are introduced below.

The first challenge is the modeling and control of mobile manipulators. In traditional methods, the mobile platform and manipulator are controlled separately, which is not efficient for some complicated tasks in teleoperation such as the material handling. In order to achieve an efficient teleoperation, it is necessary to develop the whole-body model and control of mobile manipulators. The existing methods model the mobile platform as the new links and joints of the manipulator and apply the traditional control methods on standard manipulators to control the system. However, the accuracy of such system could not be always achieved because of the dynamics differences between the mobile platform and manipulator. The major difficulties of this issue include modeling of dynamic differences, online detection of dynamics differences and control methods to limit the motion errors.

The second challenge is the teleoperation of multiple mobile manipulators. For some tasks such as formation, large-size object transportation and exploration, one mobile manipulator is not enough to perform these tasks. Therefore, teleoperation of multiple mobile manipulators is required. The existing methods model and control the system by taking time as the reference. The system stability under random communication delays and unexpected events is a major problem for these methods. The major difficulties of this issue include the modeling of the random communication delays and unexpected events and multi-robot control design under random communication delays and unexpected events.

The third challenge is to model the human teleoperator in teleoperation system. Since the operation status of human teleoperator is stable at different time and situations, the actual teleoperation performance is always subject to the operation status of the teleoperator. Therefore, it is necessary to model the human teleoperator in the teleoperation system. Unfortunately, most studies on teleoperation have been focused on the stability and telepresence. There are no existing methods for modeling the human teleoperator in teleoperator. The difficulties of this issue include the online evaluation of the human teleoperator and the modeling human teleoperator into the telerobotic operations.

The fourth challenge is about the communications between human teleoperator and mobile manipulator. Conventionally, the joysticks such as spaceballs and phantom joysticks are used as the control ways for the human teleoperator to communicate with the mobile manipulators in teleoperation. Obviously, using such joysticks is not an efficient and intuitive communication way for some general tasks like manipulating objects. The other communication methods such as gestures and minds are usually uninformative and unstable which are also not efficient. Therefore, we choose natural language as the new way. However, the existing methods on natural language control have mainly focused on transferring the natural language commands to the some formal representations which could be understood by the robot. They are not capable of online handling unexpected events in the environment and robotic system. The major difficulties include the modeling and detection of unexpected events and online unexpected event handling design in natural language control.

## **1.3** Literature Review

#### **1.3.1** Review on Modeling and Control of Mobile Manipulators

Due to the integration of the advantages of the standard manipulators and mobile platforms, mobile manipulators have been widely used in many applications including industrial manufacturing, hazardous material operations, outer-space exploration, etc. [19, 20, 21].

The working space of a standard manipulator can be enlarged by introducing a mobile platform, commonly a nonholonomic mobile platform. However, this integration of two different systems also introduces new challenges [20][21]. First, the models for the mobile platform and manipulator are different. The manipulator is usually a holonomic system but the mobile platform may be subject to nonholonomic constraints. A method is required to incorporate the nonholonomic constraints during the system modeling. Second, the integrated system is highly redundant and the redundancy resolution scheme is required. Third, from the view of practical implementation, even if the above two theoretical problems are perfectly solved, the control performance may still be affected by system uncertainties and exceptions, such as different characteristics, unstructured working environment and unexpected events.

Many studies have been conducted on the whole-body modeling and control of nonholonomic mobile manipulators. There are commonly two ways to model the kinematic system with nonholonomic constraints. One way is to directly add the constraints to the velocity kinematic model [22][23]. Another more efficient way is to model the system to explicitly entail the admissible motions with respect to the nonholonomic constraints [24][19], which is also used in this paper. After the model is obtained, the nonholonomic mobile manipulator system is usually redundant for a given end-effector task with usually no more than six dimensions. The redundancy resolution methods for standard manipulators can then be extended to the nonholonomic mobile manipulator including Extended Jacobian method [25], task priority method [26], Reduced Gradient based method [27], Singularity-Robust method [28], Damped Least-Squares Inverse Jacobian method [29], etc. A task-space or end-effector space closed-loop controller can also be incorporated into the redundancy resolution schemes to make the robot track the desired end-effector motions.

In these existing methods, the fundamental principle is to model the mobile platform as the new joints and links of the manipulator and extend the traditional control methods for the standard manipulators to the mobile manipulators. However, when implementing these control methods to the actual mobile manipulators, two major problems occur to seriously affect the control performance of the mobile manipulators in terms of the motion accuracy.

The first problem is about the non-optimal motion distribution between the mobile platform and manipulator. When using the traditional control methods, there is no preference on the motion of the mobile platform or the manipulator. In other words, the control always tries to move the mobile platform and manipulator at the same time to achieve the task. Because the characteristics and working conditions of the mobile platform and manipulator are different, their tracking performance differs a lot. Usually, the manipulator is more accurate than the mobile platform, which is especially true for a rugged nonholonomic mobile platform designed for outdoor and heavy-load applications, such as the mobile platform used in the later experiments. The indiscriminate motion distribution makes the motion tracking errors of entire mobile manipulator (i.e., errors of the end-effector) large, which can actually be avoided. Intuitively, to achieve a better motion tracking performance of the entire mobile manipulator, we should try to use the manipulator to achieve the task as much as possible instead of equally using the mobile platform and manipulator.

The second problem is about the lack of efficient coordination between the mobile platform and manipulator. Although we can have a preference on the manipulator motion, we still have to use the mobile platform and manipulator at the same time to achieve the task in some cases. For these cases, if using the traditional control methods, the motion tracking performance of the mobile manipulator cannot always be maintained at the best due to the motion differences of the mobile platform and manipulator. For example, sometimes the manipulator may move much faster than the mobile platform and it would increase the tracking error of the entire mobile manipulator when there is no coordination between them. For another extreme example, either the mobile platform or the manipulator may be stopped by unexpected events such as a sudden obstacle or machine halt. The traditional methods would make the tracking error of the entire mobile manipulator larger and larger if this happens because they are not aware of that one system cannot move and always use a single system to achieve the task. Therefore, an efficient coordination between the mobile platform and manipulator is needed to minimize the errors of the entire mobile manipulator and handle the unexpected events.

#### **1.3.2** Review on Teleoperation of Multiple Robotic Systems

The teleoperation system could be considered as two robotic systems a master and a slave which are connected through the communication channels. The problem of the time delays in the communication channels may affect the stability of the system. Ferrel has conducted a lot of experiments to show that within the time delays of 0.3 s a human operator could maintain sensor motor coordination during the manual teleoperation under the visual feedback [30][31].

The teleoperation system can be considered as two robotic subsystems a master and a slave that exchange signals (positions, velocities and/or forces), in which the slave tries to mimic the behavior of the master which in turn takes into account the input torques from the slave.

To handle bigger time delays, the teleoperation system was later on modeled based on the two-port networks. The two-port network was a four-terminal network which has two pairs of terminals connected internally. The input and output signals of the network were defined efforts and flows which were forces and velocities in the teleoperation systems and also corresponding to the voltages and currents in the electric circuits. Then in this two-port network formulation, the teleoperation system was designed based on the passivity theory [32, 33, 34]. Such formulation and teleoperation method have been widely used in different teleoperation systems under communication delays [35, 36, 37].

The advanced control theories have also been applied to teleoperation for handling the communication delays in some special cases including the predictive control [38][39], sliding-mode control [40][41], robust control [42][43], frequency domain control [44][45], etc.

Recently, some researchers have also researched on the multi-robot teleoperation systems

because of the requirement that some teleoperation applications, such as the exploration, patrol and formation, require multiple robots to be teleoperated and simultaneously work together. Multiple mobile manipulators can be teleoperated by a single operator to provide enhanced capacity and efficiency on accomplishing complicated tasks. The design of this kind of systems is challenging because simultaneously tele-controlling multiple robots exceeds the ability of one single operator. The intelligence and autonomy of the robots need to be integrated into the system. Besides, the communication between the operator and the multi-robot system and the communication among the multiple robots are both subject to communication constraints like time delays and packet losses.

In [46][47], multi-robot system is assumed to be highly autonomous. The operator supervises the multi-robot system and sends commands by shared natural language when he/she thinks it is necessary. This approach highly relies on the autonomy of the robots and lack of safety. A semi-autonomous bilateral teleoperation framework for the single-operator-multirobot (SOMR) system is proposed in [48]. The dynamics of the multiple slave robots is decomposed into two decoupled systems while enforcing energetic passivity: the shape system describing the cooperation aspect such as formation keeping, and the locked system abstracting the overall behavior of the multiple slave robots. An approach for balancing a single operator's command and multi-robot's autonomy is proposed to allow a human to better control a team of robots in [49]. A switching control scheme is proposed in [50]. The controller allows implementation of proper switching between different control modes with consideration of different important cases. In [51], information graphs and consensus algorithms are used to approach multilateral teleoperation under dynamically changing networks. In [52][53], leader-follower approach is applied. A leader robot is selected and teleoperated by an operator and the follower robots are autonomously coordinated to make a formation.

In these existing studies on single-robot and multi-robot teleoperation, in order to ensure the stability of the system, the network delays need to be constrained such as being constant or bounded. However, many delays in the communication channels like the Internet are completely random, which cannot satisfy these constraints. In addition, these existing methods are not capable of efficiently handling unexpected events in the system such as unexpected stop of one or several robots in the system.

# 1.3.3 Review on Modeling the Human Teleoperator in Teleoperation

The studies on teleoperation in the past decades have mainly been focused on two major issues: stability and telepresence of the telerobotic systems. The study of stability is mainly to design control methods to ensure the stability of the telerobotic systems in presence of various system constraints such as communication delays, packet losses and unstructured environments [54, 55, 56]. The study of telepresence is mainly to develop teleoperation techniques to enable the teleoperator to control the robots and perceive the remote environment in efficient and visual ways as though sympathetic visage [57, 58, 59].

Most of these studies are based on the assumption that the teleoperator can maintain a good operation status. This assumption, however, cannot be always guaranteed. For instance, the teleoperator may be suffering from sleep deprivation, illness and even being inebriated. Under such conditions, the teleoperation efficiency and safety may be seriously affected due to the deceased operating capacity of the teleoperator. These conditions, however, have been scarcely considered in the previous teleoperation studies.

In order to model the human teleoperator in the teleoperation system to enhance the

performance of telerobotic operations, it is important and necessary to identify the operation status of the teleoperator. Some methods have been proposed for the identification of the operation status of human operators. These methods can be categorized into two groups: off-line methods and online methods.

The off-line methods identify the operation status of human operators based on their performance in accomplishing some designed tests and tasks such as questionnaires, playing computer games, operating virtual or real robots, etc. [60][61]. However, the operation status of humans may vary as the time involves. Unfortunately, the off-line methods cannot monitor such variation.

Online methods have also been studied by some researchers. Most of them are based on the physiological states of the human operators. These methods online collect the physiological information from the operators and use them to online evaluate their operation status. The physiological information include facial expressions, heart rate, blood pressure, heat, sweat, etc. [62][63]. However, these results can only reflect the external status of human operators but cannot reflect their internal status, such as frustration, boredom, etc., which are actually also very importance factors to the evaluation of the operation status.

However, there are still no models for representing the human teleoperators, especially from the internal states of the human teleoperator. Fortunately, there exist some methods to capture the mental states of the teleoperator such as excitement, boredom, frustration, etc. which could provide important information for modeling the human teleoperator. These methods are based on the human teleoperator's brain activities. There are various ways to capture the brain activities such as Magnetic Resonance Imaging (MRI), Magnetoencephalography (MEG) and Electroencephalograph (EEG). Among them, the noninvasive EEG based BCI (Brain Computer Interface) technology is the most convenient one to capture the brain activities and generate the mental states.

The EEG signals, firstly obtained by Hans Berger in 1924, are from the spontaneous and rhythmic electrical activities of brain cells. They can be obtained along the teleoperator's scalp using electrode arrays. Through the EEG based BCI, we can obtain many mental states of human beings such as mental fatigue, interests, frustration, boredom, minds, etc. [64, 65, 66]. The EEG based BCI technology has been applied to many areas such as the clinical diagnosis and therapy. Recently, they have also been applied to teleoperation. Most of these works have been focused on extracting the teleoperator's minds from the EEG signals for constructing the control commands. Such studies have been conducted on various types of robotic systems including mobile robots [67], manipulators [68], remote tour guidance robot [69], wheelchair robots [70], power-assist robots [71], humanoid robots [72], etc.

Unfortunately, as shown in many experimental results, the EEG based commands are not stable signals compared to the commands from the conventional command generators like joysticks. It is very difficult and challenging for the teleoperator to consistently and stably maintain their mind at one point. Therefore, in current circumstances, it is not an appropriate way to use the EEG based BCI to directly control the robotic systems to conduct some tasks requiring high accuracy and dexterity. However, instead of directly controlling the robots, the EEG based BCI could be used for the online identification of the operation status of the teleoperator and the result could be used as the representation of the human teleoperator in the teleoperation system.

In addition, since insufficient operation status of the teleoperator may result in inefficient and even dangerous telerobotic operations, it is necessary and important to model the human teleoperator in the control of the telerobotic operations. However, there are no existing models and studies to solve problem. Therefor, a method for modeling the human teleoperator into the telerobotic systems is required such that the efficiency and safety telerobotic system could be ensured at different operation status of the human teleoperator.

# 1.3.4 Review on Communications between Human Teleoperator and Robots

In the teleoperation systems, joysticks are conventionally used as the control manner for the human teleoperator to communicate with the robotic systems. The examples of the joysticks include the spaceball, PHANToM joystick [73], nine-string joystick [74], FIU 3-DOF force-reflecting joystick [75], etc. These joysticks could only generate low-level control commands like velocities and positions. Obviously, they are neither efficient nor intuitive ways for the communications between the human teleoperator and robotic system for some tasks.

Recently, there are also some other methods which can provide more intuitive ways for the human teleoperator to communicate with the robotic systems. Using body gestures is a popular one of these methods [76, 77, 78]. However, the control commands from gestures are still in the low control level and cannot some informative knowledge for the efficient communication between the human teleoperator and robots.

Using human minds to as the communications between the human and robot is another popular way. These methods collect the brain signals of the human teleoperator and applied different methods to extract what the human teleoperator is thinking to do [65][66]. Then, the mind information could be used as the control commands to control the robots [68] [69]. However, such methods are not appropriate for real-time teleoperation tasks especially when the tasks are complicated because human minds are difficult to control. The minds are always subject to dramatic changes and It is challenging for the human teleoperator to concentrate on a specific mind for a long time [67][72].

Compared to the conventional control devices like joysticks, natural language can be used as a new control manner to provide the teleoperator a more efficient and convenient way to interact with mobile manipulators. Recent years have seen a growing interest in technology on natural language control of robotic operations. The applications of such systems range from service and entertainment robots to industrial robots [79][80].

Natural language carries high-level discrete symbolic information and robot control is a low-level continuous process. Controlling robotic operations by natural language requires a mechanism to connect these two different representations. Despite recent progress, research on natural language control of robotic systems still faces many challenges. For example, most previous work has focused on grounding linguistic terms to the robot's perception and action so that a natural language command can be translated to formal representations accessible by the robot (e.g., sensorimotor skills of the robot). However, an important issue which has not been well addressed is that, even given a perfect translation, unexpected events could happen that will prevent the execution of the natural language command. For example, suppose a human says "pick up the red bottle" and the robot understands perfectly what actions are involved (i.e., coming up with a task schedule given the current situation) and which bottle needs to be picked up. However, the situation may change after the human's command, for example, the robotic arm could be blocked or the object may be moved. The robot would fail in executing the command. Thus it is important to have a scheme that will allow the natural language controlled robotic system to intelligently handle unexpected events in both the robotic system and the environment.

In the human-robot interaction research, many studies have been focused on converting

natural language commands to some types of logic representations and then mapping them to predefined primitives of robot actions [81, 82, 83]. Some studies have specifically looked into temporal logic [84] and spatial references [85] for natural language control. Natural language tasks are interpreted as these structured representations and many approaches have been proposed to generate the robot motions based on these representations and some abstracted plant models [86, 87, 88]. Recently, some studies have also used the probabilitybased methods to directly convert natural language commands to robot motions through off-line training or online learning of the robotic systems [89, 90, 91].

Despite recent progress, the research on natural language controlled robotic system still faces some challenges. Most existing methods designed the connection between the natural language processing and the robot control in an open-loop manner. Once the robot motions were generated, they were directly executed by the robot. This formalism usually needs to assume that the control system could absolutely accomplish the actions and there were no exceptions in both the robotic system and the environment. If the assumption was violated, errors would occur to generate problems for the systems [92][88]. For example, if the environmental setup is changed, the previously generated robot motions may no longer be correct. For another example, if the robot is stopped, some mechanism in the robot planning and control processes needs to be designed to ensure the correctness and safety of the robot execution. Therefore, a systematic design of the natural language controlled robotic system is required to realize the natural language control under system exceptions.

## 1.4 Objectives of the Study

The overall objective of the study is to develop stable and efficient teleoperation of mobile manipulators. Based on the challenges and difficulties, the overall objective can be divided into four objectives.

The first objective is to develop whole-body motion control of mobile manipulators under system dynamics differences so that the teleoperation efficiency for some complicated tasks could be improved.

The second objective is to develop a teleoperation method for controlling multiple mobile manipulators by a single teleoperator under random delays and unexpected events so that some tasks such as the formation and co-transportation could be accomplished stably and efficiently.

The third objective is to model the human teleoperator in the mobile manipulator teleoperation system so that the teleoperation efficiency and safety could be ensured under different operation status of the teleoperator.

The fourth objective is to develop a natural language teleoperated mobile manipulator system under unexpected system events so that the teleoperation could be more intuitive and efficient for the teleoperator.

## **1.5** Organization of the Dissertation

The organization of the dissertation is described as follows. Chapter 2 introduces modeling and control of nonholonomic mobile manipulators including the autonomous whole-body control and online redundancy resolution for teleoperation. Chapter 3 introduces the stable teleoperation of multiple mobile manipulators by a single teleoperator under random communication delays and unexpected events. Chapter 4 introduces the online identification of the quality of teleoperator and the adaptation of the telerobotic operations adaptive to the quality of teleoperator. Chapter 5 introduces the teleoperation of mobile manipulators though the natural language and unexpected event handling in the natural language controlled robotic system. Chapter 6 introduces the hardware and software development of two types of mobile manipulators including the holonomic mobile manipulators and nonholonomic mobile manipulators. Chapter 7 gives conclusions of the dissertation and the future research work of the study.

# Chapter 2

# Modeling and Control of Nonholonomic Mobile Manipulators for Teleoperation

# 2.1 Introduction

This chapter introduces the modeling and control for mobile manipulators for teleoperation. It proposes a new control method for a nonholonomic mobile manipulator to minimize the motion tracking errors in presence of performance differences between the mobile platform and manipulator. With the new method, the problem can be addressed through introducing adaptive motion distribution and coordination between the mobile platform and manipulator [93][94]. First, the system model incorporating the nonholonomic constraint is introduced and the traditional kinematic control methods with redundancy resolution are presented. Second, different from the motion distribution in the traditional methods, a motion preference factor is introduced to distribute as much motion as possible to the more accurate manipulator. The motion preference factor is automatically calculated based on the configuration of the manipulator. Third, a coordination method in a perceptive frame is designed to efficiently coordinate the motions of the mobile platform and manipulator so that the errors of the entire mobile manipulator can be minimized.

In addition, in conventional teleoperation methods, the teleoperator is responsible for taking care of all tasks including the primary end-effector task and other secondary task such as obstacle avoidance. This introduces a lot of burden for the teleoperator and impact the efficiency of the teleoperation especially when the task is complicated. To address this issue, this chapter also introduces a control method to make the teleoperator only concentrate on the primary task related to the end-effector and the obstacle avoidance of the manipulator links is automatically handled by an autonomous controller [95][96]. The method is based on the redundancy of the mobile manipulator. The extra degrees of the mobile manipulator provide and advantage of redundancy that the robot could achieve the primary end-effector task and simultaneously realize some secondary tasks without affecting the primary task.

The designed methods were implemented tested on a nonholonomic mobile manipulator teleoperation system and the experimental results demonstrated the effectiveness and advantages of the methods.

# 2.2 Whole-Body Modeling and Control of Mobile Manipulator in Teleoperation

#### 2.2.1 Whole-Body Modeling of Mobile Manipulator

The nonholonomic mobile manipulator studied in this paper was developed at Michigan State University (MSU). It contains a 4-wheel drive mobile platform and a 7-DOF on-board manipulator as shown in Figure 2.1.

Define r as the end-effector states,  $u_b$  and  $u_a$  as the velocity inputs of the mobile platform



Figure 2.1 MSU Nonholonomic Mobile Manipulator

and the manipulator, the kinematic model is described by

$$\dot{r} = \begin{bmatrix} J_b & J_a \end{bmatrix} \begin{bmatrix} u_b \\ u_a \end{bmatrix}$$

$$= J_m u$$
(2.1)

where  $J_m$  is the Jacobian of the mobile manipulator, and  $J_b$  and  $J_a$  are the Jacobian of the mobile platform and the manipulator.

u contains nine joint motor velocities of the mobile platform and manipulator. The end-effector velocity  $\dot{r}$  is usually a variable of up to six dimensions. Therefore, the mobile manipulator system is highly redundant. For a given desired end-effector velocity  $\dot{r}^d$ , there exist many feasible solutions of u to achieve it. The kinematic control with redundancy resolution methods for standard manipulators can be extended to the kinematic control of the nonholonomic mobile manipulator. The most commonly used task priority method is adopted in this study.

The method separates the control into two spaces: operational space and null space. For a given end-effector velocity  $\dot{r}$ , the solution is given by

$$u = J_m^{\dagger} \dot{r} + (I - J_m^{\dagger} J_m) u_0 \tag{2.2}$$

where  $J_m^{\dagger}$  is the pseudoinverse of  $J_m$ ,  $I - J_m^{\dagger} J_m$  is the orthogonal projection operator which projects the joint motor velocity vector into the null space of  $J_m$ , and  $u_0$  is the designed joint motor velocity for the secondary task.

In the operational space, the task is to achieve the end-effector velocity. In the null space, a secondary task needs to be designed. It could be to achieve a designed joint motor velocity or, more commonly, to optimize a criterion that is related to the robot states. If only the manipulator-dependent criterion is considered, such as manipulability index maximization or joint limit avoidance, the criterion can be described as  $H(q_a)$ . In order to achieve a local optimum of  $H(q_a)$ ,  $u_0$  can be calculated as the gradient of  $H(q_a)$  which is given by

$$u_0 = \pm k_H \begin{bmatrix} 0_{2 \times 1} \\ \nabla_{q_a} H(q_a) \end{bmatrix}$$
(2.3)

where  $k_H > 0$  is a scalar gain.

There are also many other methods which can be applied to the kinematic control of the nonholonomic mobile manipulator including reduced gradient based method, singularity-Robust method, damped Least-Squares inverse Jacobian method, etc. These control methods, however, only consider the tracking of the end-effector's instantaneous velocity. In order to track the desired position at the same time, a closed-loop kinematic controller is necessary. When a desired task is given, the corresponding desired velocity and desired position trajectories of the end-effector can be generated and usually described as a function of time t.
$$\dot{r}^{d}(t) = v_{ee}(t)$$
  $r^{d}(t) = \rho_{ee}(t)$  (2.4)

A task-space closed-loop controller can then be obtained and expressed by

$$\dot{r} = \dot{r}^d(t) + K_r(r^d(t) - r_c)$$
(2.5)

where  $K_r$  is a constant gain matrix and  $r_c$  is the measurement of current end-effector position.

This closed-loop design is adopted in most kinematic controllers of nonholonomic mobile manipulators. However, the closed-loop control in task space cannot always guarantee the performance of the system, in particular for the motion tracking, because they do not consider the tracking performance differences of the mobile platform and manipulator caused by different characteristics and working conditions. Therefore, in order to practically achieve the best possible performance of the nonholonomic mobile manipulator, a motion distribution and coordinated control method is proposed in the following section. It can minimize the tracking error of the mobile manipulator and also handle some unexpected events.

#### 2.2.2 Control with Motion Distribution

If using the task priority method and considering multiple secondary tasks, the kinematic controller of the nonholonomic mobile manipulator can be designed by

$$u = J_m^{\dagger} \left( \dot{r}^d(t) + K_r(r^d(t) - r_c) \right) + \left( I - J_m^{\dagger} J_m \right) \sum_{i=1}^{I} \alpha_i \nabla_q H_i(q)$$
(2.6)

where I is the number of the secondary tasks,  $H_i$  is the measurement of each second task, and  $\alpha_i$  is the weight for each task. It can be seen that this method distributes the motion to the mobile platform and manipulator with no preference. As the characteristics of the mobile platform and manipulator, such as mass and inertia, are different and the working conditions of them are different as well, e.g., the contact-free manipulator working space and ground-contact mobile platform working environment, the motion tracking performance of the two systems are different. Usually the performance of the manipulator is better than the mobile platform. Therefore, a preference on the manipulator is desirable for achieving a better performance of the entire mobile manipulator. To this end, a weighted pseudoinverse is used and an adaptive weight scheduling scheme is designed. The pseudoinverse of  $J_m$  in (6) is replaced by a weighted pseudoinverse [12]

$$J_m^{\dagger} = W J_m^T \left( J_m W J_m^T \right)^{-1} \tag{2.7}$$

where the weight matrix W defined as

$$W = \begin{bmatrix} \lambda I_{2 \times 2} & 0_{2 \times 7} \\ 0_{7 \times 2} & (1 - \lambda) I_{7 \times 7} \end{bmatrix}$$
(2.8)

where  $\lambda \in [0, 1]$  is the weight determining the motion distribution between the mobile platform and manipulator. Larger  $\lambda$  implies a preference of more motion on the mobile platform and less  $\lambda$  implies a preference on the manipulator.  $\lambda = 1$  and  $\lambda = 0$  correspond to the sole motion of the mobile platform and the manipulator respectively.

In this study, maintaining the manipulability of the manipulator is defined as the secondary task and this task is used to determine  $\lambda$ . The measurement of the manipulability index is mapped to  $\sigma \in [0, 1]$ , where 1 means the best possible measurement and 0 means the worst. The adjustment is designed by

$$\lambda = \begin{cases} 1 - \sigma, & if \ \sigma \le L \\ 0, & if \ \sigma > L \end{cases}$$
(2.9)

where L is the minimum endurable limit for the manipulability.

When the manipulator is in its endurable operational range, we have  $\lambda = 0$  and use only the manipulator to accomplish the task. When the manipulator is approaching some dangerous configurations such as singular points or joint limits, we have  $\lambda \in (0, 1)$  to use both the mobile platform and manipulator to accomplish the task. The closer the manipulator is approaching the dangerous configurations, the larger  $\lambda$  is set so as to apply more pushing force and use more motion of the mobile platform. When the manipulator is on the boundary of the dangerous configurations, we have  $\lambda = 1$  and use only the mobile platform to accomplish the task. With this weight scheduling, it can be seen that we are always trying to use the more accurate manipulator as much as possible to achieve the task. Therefore, the errors of the entire mobile manipulator can be reduced through this adaptive motion distribution.

#### 2.2.3 Control with Motion Coordination

When  $\lambda = 0$  or  $\lambda = 1$ , the errors of the end-effector depend on solely the performance of the manipulator or the mobile platform. Once it falls into (0, 1), both mobile platform and manipulator need to move together to achieve the motion tracking of the end-effector. Since the motion tracking performance of the mobile platform and manipulator may be quite different and the traditional control methods are not aware of such difference, the errors of the end-effector may become unnecessarily larger. In order to handle this problem, a coordinated control method is designed based on the traditional control methods so as to minimize the error of the end-effector.

The basic idea of the coordinated control is to model the system planning and control based on a perceptive reference which is determined by the physical system outputs [14]. This idea has been successfully applied to the multi-robot coordination and multi-robot teleoperation in our previous work [15-16]. In this study, it is modified and extended to the accurate motion control of the nonholonomic mobile manipulator. The coordination method mainly contains three steps which are introduced as below.

The first step is to generate new motion plans parameterized by a new perceptive reference s. Define the perceptive reference s as the distance travelled by the end-effector along its desired motion path. The desired trajectory of the end-effector which was traditionally parameterized by the time t can be transferred to a new trajectory parameterized by s,  $\dot{r}^d(s)$ . Using the kinematic control with adaptive motion distribution, the joint motor velocities of the mobile manipulator, u, can be obtained and described by

$$u = U(\dot{r}^d(s)) \tag{2.10}$$

Using the forward kinematics, the desired velocity trajectories for the mobile platform and manipulator,  $\dot{r}_b^d$  and  $\dot{r}_a^d$ , can also be obtained. Furthermore, the desired position trajectories of the mobile platform and manipulator can be easily obtained by integration of the desired velocity trajectories. These desired trajectories are all parameterized by s and described by  $\dot{r}_b^d(s)$ ,  $\dot{r}_b^d(s)$ ,  $\dot{r}_a^d(s)$  and  $r_a^d(s)$ .

Notice that describing the motion plans by s is not a simple replacement of traditional time reference t by s. The perceptive reference s is directly related to the system outputs. Not

only the motion plans for the end-effector but also the motion plans for the mobile platform and manipulators are generated. This provides necessary information for coordinating the mobile platform and manipulator for motion tracking and unexpected event handling.

The second step is to generate the optimal motion reference for the coordinated control. The coordination goal is to minimize the error of the end-effector. This can be achieved by minimizing a coordination error defined by

$$J(s) = \left(r_a^d(s) - r_a\right)^T W_a \left(r_a^d(s) - r_a\right) + \left(r_b^d(s) - r_b\right)^T W_b \left(r_b^d(s) - r_b\right)$$
(2.11)

where  $W_a$  and  $W_b$  are the weighting matrices of the manipulator and mobile platform respectively to weigh the coordination errors based on the specific coordination requirements.

Given outputs of the mobile platform and manipulator,  $r_a$  and  $r_b$ , the instantaneous desired motion of the mobile platform and manipulator should be able to minimize the coordination error. To this end, the perceptive reference for determining the instantaneous desired motion can be calculated by

$$s^* = \arg\min_{s} \left\{ J(s) \right\} \tag{2.12}$$

Furthermore, in order to maintain the coordination error within a limit,  $J_d$ , the optimal perceptive reference can be calculated by

$$\tilde{s}^* = s_l + \frac{J_d - \min\{J(s^*), \ J_d\}}{J_d} \left(s^* - s_l\right)$$
(2.13)

where  $\boldsymbol{s}_l$  is the baseline perceptive reference and calculated by

$$s_{l} = \min \{s_{b}, s_{a}\}$$

$$s_{b} = \arg \min_{s} \left\{ \left(r_{b}^{d}(s) - r_{b}\right)^{T} W_{b} \left(r_{b}^{d}(s) - r_{b}\right) \right\}$$

$$s_{a} = \arg \min_{s} \left\{ \left(r_{a}^{d}(s) - r_{a}\right)^{T} W_{a} \left(r_{a}^{d}(s) - r_{a}\right) \right\}$$
(2.14)

where  $s_b$  represents the motion tracking status of the mobile platform and is calculated based on the mobile platform output, and  $s_a$  represents the motion tracking status of the manipulator and is calculated based on the manipulator output.

The third step is to calculate the coordinated control outputs based on the optimal motion reference. Once the optimal perceptive reference is generated, the instantaneous desired motions of the mobile platform and manipulator can be achieved by simply plugging the perceptive reference into their desired motion plans. The motions include the instantaneous desired velocities:  $\dot{r}_b^d(\tilde{s}^*)$  and  $\dot{r}_a^d(\tilde{s}^*)$ , and the instantaneous desired positions:  $r_b^d(\tilde{s}^*)$  and  $r_a^d(\tilde{s}^*)$ .

To achieve these instantaneous desired motions, the joint motor velocities for the mobile manipulator can be designed by

$$u = \begin{bmatrix} u_b & u_a \end{bmatrix}^T$$

$$u_b = (J_b^u)^{\dagger} \left( \dot{r}_b^d(\tilde{s}^*) + k_b \left( r_b^d(\tilde{s}^*) - r_b \right) \right)$$

$$u_a = (J_a)^{\dagger} \left( \dot{r}_a^d(\tilde{s}^*) + k_a \left( r_a^d(\tilde{s}^*) - r_a \right) \right)$$
(2.15)

where  $k_b$  and  $k_a$  are positive constants. The new velocity inputs are then passed to the dynamic motor controllers of both the mobile platform and manipulator for executions.

#### **Remarks:**

i) With this coordinated control design, when the coordination error J is smaller than

 $J_d$ , both mobile platform and manipulator move in a coordinated way to minimize the error, i.e., slowing down the faster one and speeding up the behindhand one.

- ii) When J is bigger than  $J_d$ , it indicates that one system tracks its desired motion much more slower than the other system. The coordination control then slows down and even stops the motion of the faster one until the behindhand one catches up with it.
- iii) If one of or both the mobile platform and manipulator are stopped by unexpected events, such as an obstacle or machine halt, the optimal perceptive reference will stop evolving as well, which causes the entire robot to stop. Therefore, the errors remain constant and the coordination can be still maintained. Once the unexpected event is removed, the optimal perceptive reference starts to evolve and the motions are automatically recovered. In the traditional control methods, such unexpected event needs to be handled by a pre-designed mechanism and a re-planning is required after the unexpected event is removed. For the proposed method, they are all automatically handled and no re-planning is required at all.

Therefore, using the kinematic control with adaptive motion distribution, the error of the end-effector can be reduced compared to the traditional control methods. Furthermore, using the coordinated control design, the coordination error can always be limited in a desired range in presence of the motion tracking differences between the mobile platform and manipulator. Besides, the coordinated control design can also automatically handle some unexpected events occurred in the robot system.

# 2.3 Sensor based Redundancy Resolution for Obstacle Avoidance in Teleoperation

## 2.3.1 Sensor based Redundancy Resolution Structure

The sensor based online redundancy resolution structure is shown in Figure 2.2 The teleoperator concentrates on the operation of the end effector and the obstacle avoidance of the other parts of the robot is automatically done by the robot itself. There are multiple sensors mounted on the robot including cameras and laser scanners. These sensors detect the operational environment on-line and send these information to an obstacle avoidance analysis scheme. This scheme can extract the obstacles on the motion of the end effector and then generate some proper configurations of the manipulator. During the teleoperation process, the teleoperator only sends task-space velocity commands to conduct the motion of the end effector. Meanwhile, the proper manipulator configuration for the desired motion is also computed on-line. The redundancy resolution scheme is then applied to translate the taskspace velocity to a proper joint-space velocity. It integrates the desired task-space velocity and computed desired manipulator configuration and then applies the inverse kinematics of the mobile manipulator for the command transformation. With this control strategy, the teleoperator can focus on the operation of the end effector regardless of the obstacle avoidance of the other parts of the robot since it is autonomously done by the on-line redundancy resolution scheme.



Figure 2.2 Sensor based online redundancy resolution

## 2.3.2 Sensor System Design

The achievement of redundancy resolution needs the 3D data of the environment, and it is feasible for any kind of sensor or combination of several sensors as long as they possess the ability to obtain 3D information of environment, such as binocular vision, 3D laser measurement sensor, and others. In our design, we use the laser scanner to get the 3D information of environment. The laser scanner is mounted on a pan-tilt system in front of the mobile base, so the scanner can detect different 3D information of the environment with the rotation of the pan-tilt. In our approach, we just use the tilt function, the laser scanning area is shown in Figure 2.3. The control system read the data from laser scanner every tilt-step angle, which is defined as  $\theta_{ts}$ . The entire tilt angle is defined as

$$\theta_{tilt} = N \times \theta_{ts} \tag{2.16}$$

where the N is the number of tilt-step.

The distance from  $O_s$  to the detected object can be defined as  $d_i$ , if  $d_i$  is smaller than a cutoff distance, which is defined as l, the object should be seen as obstacle. Before using the obstacle information, the coordinate frame of laser system needs to be calibrated with



Figure 2.3 Laser scanning area

the coordinate frame of mobile manipulator, which is shown in Fig. ??. The calibration can transfer the point coordinates  $(x_s, y_s, z_s)$  which relate the laser system frame to the coordinates  $(x_m, y_m, z_m)$  in the manipulator frame.

$$(x_m, y_m, z_m)^T = R_s^m (x_s, y_s, z_s)^T + O_s^m$$
(2.17)

where  $R_s^m$  is the transform matrix from laser system coordinates to the manipulator coordinates.  $O_s^m$  is the coordinate of the origin of laser system frame in the manipulator coordinate frame.

Sometimes, the manipulator will block the laser beam and the system will make a mistake to define the manipulator as an obstacle. In this case, we can check the joint position of the manipulator according to the forward kinematic model. If the obstacle point is too close to the joint position, it must be the manipulator itself. So we can delete this kind of points from the obstacle information, and then interpolate some points with linear fit according to the rest obstacle information. As shown in Figure 2.4, a rectangular plane which parallels to the  $Y_m Z_m$  plane of the manipulator coordinate frame is used in order to extract the contour of the obstacle at the front of mobile manipulator. All the points which can be projected to this plane are likely to block the movement of mobile manipulator, otherwise these points are not a threat to the mobile manipulator. This 1m by 2m rectangle is divided into a 20 × 40 grid, every unit of this grid is a 50mm × 50mm square. The projection of obstacle points is also divided into different unit. In each unit, we only select the nearest point, which has the smallest  $x_m$  value in manipulator coordinate frame, as the most threatening obstacle point.

After the process mentioned above, a  $20 \times 40$  matrix P is obtained to describe the contour of the obstacle, every element of this matrix is defined as

$$p_{i,j} = \begin{cases} \min(x_m(i,j)) & \exists x_m(i,j) \\ 0 & else \end{cases}$$
(2.18)

where  $i = 1, 2...20, j = 1, 2...40, x_m(i, j)$  is the  $x_m$  value of the obstacle point in corresponding (i, j) unit.

To illustrate the designed sensor system, the mobile manipulator and the working environment are shown in Figure 2.5. The end-effector will block part of the laser beams when the laser system scanning upper area. Figure 2.6 shows the raw data got by laser system. It is seen that there are a lot of obstacle points of manipulator itself because the manipulator blocks laser beam. Figure 2.7 shows the revised data of the environment. As indicated by arrows, some points are interpolated based on the linear fitting of the remaining environment data. The result shows it reconstruct the original look of the bridge structure. Figure 2.8 shows the extraction of the contour of the bridge structure.

After the 3D environmental information is obtained, the coordinate frame of sensor sys-



Figure 2.4 Coordinate frame



Figure 2.5 Mobile manipulator and working environment



Figure 2.6 Raw data of laser system

tem needs to be calibrated with the base coordinate frame of robot system. And then, next is to estimate the position of the obstacle. Basically, The proposed method in this article can be called *Projection Method*. This method is to project the link of manipulator and the obstacle on a specified planer which is perpendicular to the moving direction of the endeffector, and the relation of projections is used as the criteria. An equipotential which is perpendicular to the moving direction of the end-effector is used to cut off the environment in the front of the manipulator, as shown in Figure 2.9. The effect of the equipotential surface is to locate the obstacle. We assume D is the distance between the equipotential surface and the r-c plane of the sensor frame, and  $d_i$  is the distance from the points on the environment surface to the r-c plane of the sensor frame. If  $d_i < D$ , the points are defined as obstacle points, otherwise the points are non-obstacle one. Hence, we can get the projection



Figure 2.7 Modified data of laser system



Figure 2.8 Contour extraction of bridge structure

of the obstacle on the equipotential surface, which will be used as the obstacle.



Figure 2.9 Obstacle and Forearm are projected on Equipotential Surface

# 2.3.3 Sensor based Online Redundancy Resolution for Obstacle Avoidance

For the teleoperation, the collision avoidance of end effector is controlled by operator' according to his surveillance. When the operator focuses on the end-effector, the forearm part, EW, is in uncared situation, so it is dangerous and the collision may happen. Hence, an idea that utilizes redundancy of redundant manipulator to avoid potential danger is considered.

According to the 3D data of the environment which is discussed above, the projection of obstacle on the equipotential surface can be obtained. If the projection of forearm link EW overlaps or intersects the obstacle projection, the collision may not necessarily happen , otherwise if the projection the projection of forearm link EW does not overlap the obstacle projection, the collision will never happen. Now, the forearm link EW is projected to the equipotential surface and becomes  $E_iW_i$ . Hence, the shortest distance  $\rho$  between the projection  $E_iW_i$  and the edge of obstacle can be used to construct the potential field relation.  $\rho_1$  and  $\rho_2$  represent the limit distance of the potential field influence to avoid undesirable disturbance beyond the obstacle's vicinity, as shown in Figure 2.10. In addition, the point P represents the closest point on the surface of an obstacle to the surface of the link  $E_iW_i$ . The point Q is the point on  $E_iW_i$  that is closest to P.



Figure 2.10 Distance relation between the link projection and the obstacle projection

This is a double layers protection shell.  $\rho_1$  is absolute security distance, which defines the inner layer, and  $\rho_2$  is the relative security distance, which defines the outer layer. Thus, when the projection  $E_iW_i$  enters into the outer layer, namely  $\rho_1 < \rho < \rho_2$ , the forearm link EW will be affected by the virtual velocity. The virtual velocity is designated  $v_f$ 

$$v_f = k(\frac{1}{\rho} - \frac{1}{\rho_2}) \tag{2.19}$$

where k is an user-specified proportional coefficient. Note that the direction of  $v_f$  is from P to Q.

When the forearm link is close to the obstacle, the  $\rho$  is less than  $\rho_1$  possibly, even the  $E_i W_i$  may overlap the obstacle projection. In this case, the arm angle needs to rotate as a fast constant angular velocity, which is denoted as  $V_{\text{max}}$ . This case is dangerous, so we can employ an approach to cease the motion of end-effect and only to rotate the arm angle with predefined velocity until the shortest distance  $\rho$  equals  $\rho_1$ . Generally,  $v_i$  is constrained by the maximum angular velocity  $V_{\text{max}}$  of arm angle. Therefore, the virtual velocity  $v_{\phi}$  is

$$v_{\phi} = \begin{cases} V_{\max} & \rho \le \rho_1 \\ k(\frac{1}{\rho} - \frac{1}{\rho_2}) & \rho_1 < \rho < \rho_2 \\ 0 & \rho \ge \rho_0 \end{cases}$$
(2.20)

where  $V_{\text{max}}$  is an user-specified constant velocity. Note that in order to ensure absolute security,  $V_{\text{max}}$  is applied to the arm angle until projection  $E_i W_i$  leave the inner layer.

# 2.4 Experimental Results

#### 2.4.1 Results of Whole-Body Control

In order to verify the effectiveness and advantages of the proposed control method, both the traditional kinematic control method and the proposed control method were applied to the MSU nonholonomic mobile manipulator. The mobile platform is a heavy and rugged system with slow dynamics and less accurate performance. It moves on unstructured grounds. In contrast, the manipulator is a light-weight and precise system with fast dynamics and more accurate performance. It moves in the contact-free environment. Notice that small errors of the mobile platform, in particular the turning errors, will result in larger errors of the end-effector. Therefore, the motion distribution and coordination between the mobile platform and manipulator becomes necessary for the accurate motion control of the end-effector. The task priority method was taken as the traditional method and the designed kinematic control with both the adaptive motion distribution and coordination was considered as the proposed method. Both simulation and experiments were conducted using the traditional method and the proposed method.

In the simulation, the task was to track a sine curve trajectory. The first one used the traditional method and the second one used the kinematic control with adaptive motion distribution and online coordination. When implementing the methods, random errors were added to the mobile platform and manipulator. The errors for the mobile platform were set as up to three times than the errors for the manipulator.

Figure 2.11 and Figure 2.12 show the results of these two methods. The upper plot is the 3D trajectories of the end-effector and the mobile platform, where the actual and desired trajectories of the end-effector on the top overlaps due to the scale, but the lower plot clearly shows the errors of the end-effector including the x and y position error and yaw error in a proper scale. In Figure 2.12, we can see that when using the traditional method the mobile platform was required to move all the time and the end-effector errors were large. The adaptive motion distribution design was applied to move the mobile platform as less as possible and the end-effector errors were therefore reduced. The coordination design could further reduce the errors when both mobile platform and manipulator were moving at the same time. We can see that the end-effector errors were significantly reduced and always maintained in a small range. Besides, Figure 2.12 also shows the advantages of the proposed method in handling unexpected events. When the mobile platform was stopped at around 105 s, it caused the stop of the optimal perceptive reference and consequently caused the stop of both the mobile platform and manipulator. All errors then remained constant. At around 115 s when the unexpected event was cleared, the optimal perceptive reference started to evolve and the motions were automatically recovered. The entire process was carried out automatically and no re-planning or resetting of the system was required.



Figure 2.11 Results of the traditional control

The traditional kinematic control method and the proposed kinematic control method were also experimentally implemented on the MSU nonholonomic mobile manipulator developed at the Michigan State University. The desired trajectory was generated based on the velocities sampled from a spaceball joystick which was arbitrarily moved by a human operator, as shown in Figure 2.13. The goal was to make the end-effector of the mobile manipulator to track the arbitrarily generated trajectory. Because the yaw angle tracking



Figure 2.12 Results of the proposed control with motion distribution and coordination errors of the mobile platform and manipulator are most different, to better demonstrate the effectiveness of the proposed method without loss of generality, the yaw angle tracking errors of the end-effector using the two methods are presented.

Figure 2.14 shows the results of using the traditional method and Figure 2.15 shows the results of using the proposed method with adaptive motion distribution and coordination. It can be seen that the errors of the former were much larger than the latter. There were some very large error segments in the traditional control. This usually happened when large changes occurred in the motion, e.g., large changes of velocity amplitude or direction for which the responses and motion tracking performance of the mobile platform and manipulator were quite different. Usually the mobile platform responded slowly due to its heavy load and the rough ground condition and therefore had larger motion tracking errors. This would probably cause larger errors of the end-effector when using the traditional method. How-

ever, using the proposed method, the errors could be always kept in a much smaller range for the entire process, which actually benefited from the motion distribution with preference on the manipulator and the coordination between the mobile platform and manipulator for minimizing the coordination error. Figure 2.15 also shows the advantages of the proposed method for handling unexpected events. When the mobile platform and manipulator were unexpectedly stopped at around 20 s and 38 s respectively, the mobile platform and manipulator stopped and did not recover until the obstacles were removed at around 28 s and 42 s. The entire process was carried out automatically and the yaw angle error could always be maintained in a desired small range.



Figure 2.13 Experimental Setup

## 2.4.2 Results of Sensor-based Redundancy Resolution

The designed sensor based online redundancy resolution method was implemented on the nonholonomic mobile manipulator. The 7-DOF manipulator was controlled by a configuration controller which ensures that end-effector tracked user-specified trajectories. The configuration controller made the arm execute the commanded motion for the end-effector and the arm angle. During the manipulation, the teleoperator only focused on the motion of end-effector, while the detection system continuously computed the closest distance between the forearm link projection and the obstacle projection. The real-time controller implements



Figure 2.14 Errors of the traditional control



Figure 2.15 Errors of the proposed control with motion distribution and coordination

the redundancy resolution strategy so that the forearm link was always in safe situation. There were three experiments to illustrate the effectiveness and real-time ability of the proposed method respectively. The value of k was found empirically after a few trial-and-error runs, we took k = 2. Additionally, we took  $\rho_1 = 50mm$ ,  $\rho_2 = 150mm$  and  $V_{\text{max}} = 0.04rad/s$ . The first experiment aimed to give a general motion relationship between the closest distance and the arm angle. The process was that the teleoperator controlled the manipulator to catch a target in front, while a box was put between them as a movable obstacle. When the manipulator was moved forward and the obstacle was moved to block the motion, we could find the relationship between the closest distance and the arm angle. Figure 2.16 shows that the distance between the obstacle and the manipulator was detected in real time by a laser scanner. we can also see that when the obstacle was detected far away from the intermediate links of manipulator, the arm Angle kept still. However when the obstacle was detected near to the intermediate links of manipulator, the arm Angle would adjust autonomously to make the links be far away from the obstacles.



Figure 2.16 Obstacle Distance and Elbow Angle

The second experiment demonstrated the autonomous avoidance of stationary obstacles. In the experiment, with a obstacle in the front of forearm link EW, the end-effector was commanded to move along x-axis direction of base frame. If the arm angle did not change, the collision would happen along with the motion of the end-effector. However, with the proposed method, the arm angle can change according to the detected information, and the collision avoidance based on sensor system can be realized. In Figure 2.17, the velocity command of x-axis positive direction is shown in (a), which indicates end-effector keeps moving forward in x-axis direction. In (b), the change of the closest distance between the forearm link projection  $E_iW_i$  and the obstacle projection, i.e.  $\rho$  is indicated with the blue curve. At 10.6s, the figure illustrates that the link  $E_iW_i$  leaves the inner layer. So, the Initial part shows the  $E_iW_i$  overlaps the obstacle projection and the inner layer before 10.6s; the following part shows the  $E_iW_i$  is moving in outer layer after 10.6s. (c) shows the angular velocity of arm angle. Correspondingly, we can find the angular velocity keeps to be 0.04rad/s before 10.6s and  $v_{\phi}$  is decreasing as the  $\rho$  is increasing. The virtual velocity can 'push' the forearm link away from the obstacle.

The third experiment demonstrated the autonomous avoidance of moving obstacles. In this experiment, the real-time ability of the proposed method was demonstrated. The obstacle was kept moving in the front of forearm link randomly, namely the  $\rho$  was changed randomly, and the response of the angular velocity of arm angle was checked. Similarly, the task space command of the end-effector in x-axis direction of the base frame was sent by the teleoperator, then the angular velocity of the arm angle was autonomously calculated according to the sensor information. In Figure 2.18, the velocity command of x-axis direction in this experiment is shown in (a). The value of the blue curve indicates end-effector keeps moving forward in x-axis direction. The change of  $\rho$  is indicated in (b), and the corresponding change of  $v_{\phi}$  is shown in (c). Obviously, once the  $\rho$  is less than 50mm, e.g. from 3.7s to 5.1s, the  $v_{\phi}$  becomes 0.04rad/s. It is clearly that the velocity of the arm angle changes



Figure 2.17 Avoidance of stationary obstacles

accordingly and correspondingly to the change of the closest distance between the forearm link and the obstacle.

Therefore, through these experiments, the effectiveness of real-time autonomous obstacle avoidance was demonstrated. More importantly, the obstacle avoidance did not affect the motion of the end-effector. This made the teleoperator be able to focus on the end-effector task while other tasks like obstacle avoidance could be handled by the mobile manipulator itself based on the sensor based redundancy resolution.



Figure 2.18 Avoidance of moving obstacles

# 2.5 Chapter Summary

This chapter introduced a whole-body control method for a nonholonomic mobile manipulator. It can distribute the motion to the mobile platform and manipulator adaptively to the configuration of the manipulator and coordinate the motion between the mobile platform and manipulator to minimize the errors of the entire mobile manipulator. Compared to the traditional methods, it can significantly reduce the errors of the end-effector and automatically handle some unexpected events. Both simulation and experimental results demonstrated the effectiveness and advantages of the proposed method.

In addition, a sensor based real-time redundancy resolution is proposed for the teleop-

eration of the redundant mobile manipulator. The real-time obstacle avoidance is achieved using this scheme. A projection method is proposed to extract the proximity between the link of manipulator and the obstacle from the real-time sensor information. A sensor based online redundancy resolution scheme is employed properly control the mobile manipulator to realize the motion tracking of the end effector and obstacle avoidance of the arm links at the same time. The experimental results demonstrated the effectiveness of the proposed method.

# Chapter 3

# Teleoperation of Multiple Mobile Manipulators

# 3.1 Introduction

This chapter introduces the teleoperation of multiple mobile manipulators under random communication delays and unexpected events. A non-time perceptive reference based method is designed to model and control the mobile manipulator teleoperation system and the stability of the system can be ensured under random communication delays [97]. The method is then extended to the teleoperation of multiple mobile manipulators to handle the random delays. Besides, a online coordination method is also designed based on the perceptive reference to coordinate the multiple mobile manipulators for accomplishing complicated multi-robot tasks like the formation and co-transportation under unexpected events of the robotic systems. Through these designs, a single teleoperator is able to stably control multiple mobile manipulators under random communication delays and unexpected events. The designed methods were experimentally implemented on the mobile manipulator teleoperation systems and the experimental results demonstrated the effectiveness and advantages.

# 3.2 Mobile Manipulator Teleoperation System

#### 3.2.1 Perceptive Planning and Control

Traditional planning and control is described in Figure 3.1 (a). As shown, the planning is usually an open-loop process. The time variable is used as the motion reference and the motion plan is parameterized by time based on the defined tasks. Then, the planner generates the desired instant output to the control system by plugging the current referenced time into the time-based motion plan. All uncertain and unexpected events are left to the control system to handle. If these events are not considered in the action plan, then the controller alone is not able to handle them. This is especially true when it is working in an unstructured environment. Therefore, it is very important to incorporate the planning and control processes based on system output measurement to handle these events and achieve the best possible performance.

The planning and control in the perceptive reference frame aims to handle uncertain and unexpected events in both planning and control levels and is described in Figure 3.1 (b). It was named event-based planning and control originally [98] and then renamed as planning and control in the perceptive reference frame later since "perceptive" can better describe the essence of the method. The basic idea of the method is to introduce a new motion reference variable named perceptive reference, denoted by s. The perceptive reference is independent of time and related to the system output measurement relative to the task. The motion plan is parameterized by the perceptive reference rather than the time variable. The desired instantaneous output of the planner to the system is no longer generated based on current time but instead based on the perceptive reference, which is determined by current system output measurement. As a result, the desired system input is a function of the current system outputs for any given instant of time. This creates a mechanism to adjust the plan based on system outputs and, more importantly, makes the planning a closed-loop process to handle uncertain and unexpected events. In addition, the perceptive reference is calculated at the same rate as the feedback. In other words, the planning process is running at the same rate as the feedback control process, which enables the system to handle both discrete uncertain and unexpected events, such as sudden obstacles, machine halt, etc., and continuous uncertain and unexpected events, such as error accumulation, parameter drifting, etc.



Figure 3.1 (a) Traditional planning and control (b) Planning and control in perceptive reference frame

The planning and control in perceptive frame has been used in several robotic applications including mobile robots, manipulators, satellites [98][99][100]. It has also been proven to be efficient in teleoperation to enhance its stability and safety [54][2].

## 3.2.2 Mobile Manipulator Teleoperation Design

The single mobile manipulator teleoperation consists of three individual modules which are teleoperation station, communication media and robot system. The teleoperation station includes the operator and the master, where the joystick is used as master. And also Internet is used as the communication media. The robotic system consists of a mobile manipulator and the sensor systems. The mobile manipulator is the developed nonholonomic mobile manipulator which contains a 4-wheel mobile base and a dexterous 7-DOF arm robot. The sensor system is used to detect the information of environment. In the control process, the teleoperator sends the six dimensional velocity commands of the gripper, through the Internet to the remote robot system. Then, the robot responds to implement the task. At the same time, the sensor systems detect the environmental information, such as image, audio, and force information, and send them back to the teleoperator. Usually, the motion of the robot is supervised by teleoperator to ensure the safety of the teleoperation.

For the teleoperation of a mobile manipulator, the robot and the teleoperator are usually connected through some physical networks such as the Internet, cellphone network and satellite network. However, such networks are usually subject to some communication constraints such as time delays. In order to address this issue, a teleoperation design in the perceptive frame is proposed to ensure the stability of the teleoperation system. The system is shown in Figure 3.2. The teleoperator operates the remote robot through a network by sending velocity commands and observing the environmental feedback. The teleoperation velocity is first processed by the Motion Planner to generate the desired trajectories. Then, the Kinematic Controller computes the input for the robot to achieve the desired trajectories. There are three loops in the framework. The most external loop is for the task level. This loop is mainly dominated by the teleoperator. The most internal loop is for the lower control level. This loop is controlled by the kinematic controller with redundancy resolution. It will be responsible for tracking control of the desired trajectories also may also be responsible for some additional objectives such as manipulability keeping, joint limit avoidance, obstacle avoidance, etc. Between these two loops, there is another loop for the motion planning which is different from the open-loop motion planning in the traditional teleoperation. The planning and control is designed in the perceptive reference frame. Benefiting from it, the stability and safety of the robotic system can be ensured under random network constraints and unexpected events. The design is introduced below in detail.



Figure 3.2 Single mobile manipulator teleoperation system

The velocity for the end-effector is generated by the teleoperator and sent to the robot. Once the robot receives it, it uses this velocity to generate its desired trajectories parameterized by a perceptive reference in the next teleoperation round as  $\dot{y}_p^d(s)$  and  $\dot{y}_p(s)$ ,  $s \in (s_c, s_c + \Delta s)$ , where s is the perceptive reference,  $s_c$  is the reference at the time when the robot receives the velocity, and  $\Delta s$  is reference extension related to the teleoperation velocity. There are many choices for the perceptive reference. Among them, the distance travelled by the end-effector is the most popular one. If this reference is chosen, then the reference extension can be computed as  $\Delta s = ||v_{tele}T_{tele}||$ , where  $v_{tele}$  is the teleoperation velocity and  $T_{tele}$  is the sampling time for the teleoperator. After these are all set, the planning and control contains three steps.

The first step is to compute the optimal perceptive reference from the robot feedback to represent the robots current status. This can be computed by a minimization expressed by

$$s^* = \arg\min_{s} \left\{ \left\| y_p^d(s) - y_p \right\| \right\}$$
(3.1)

In this design, the optimal perceptive reference is computed by minimizing the tracking error and can be calculated by an orthogonal projection from the current end-effector position to the desired trajectory parameterized by the perceptive reference.

The second step is to plan the desired instantaneous inputs for the controller based on the optimal perceptive reference. This is a simple process of plugging the optimal perceptive reference into the desired trajectory to obtain the desired instantaneous inputs  $\dot{y}_p^d(s^*)$  and  $y_p^d(s^*)$ .

The third step is to construct the controller based on the instant input, which is a simple replacement process. The structure of the controller is the same as the traditional time-based controller. The only change is to replace time-driven desired instantaneous inputs  $\dot{y}_p^d(s)$  and  $y_p^d(s)$  by the new desired instantaneous inputs driven by the optimal perceptive reference  $\dot{y}_p^d(s^*)$  and  $y_p^d(s^*)$ .

These three steps form not only a closed-loop control but also a closed-loop planning which could guarantee the control accuracy, stability and safety under random network constraints such as delays and packet losses and unexpected events such as sudden obstacles. This is due to the fact that the controller input is always determined by the robot output through the closed-loop planning. Generating the optimal perceptive reference by minimizing the tracking error makes the planner always provide the best controller input with the best tracking accuracy. In addition, the random network constraints such as delays and packet losses have no effect on the stability of the controller. If large network delays or long-term continuous packet losses occur, the optimal perceptive reference will end evolving at the latest once it reaches it, which will suspend the robot motion until new teleoperation commands arrives or the network restores normal conditions. If an unexpected event such as a sudden obstacle occurs to block the end-effector, at the same time, the optimal perceptive reference is stopped as well, which will also suspend the end-effector motion until the unexpected event is cleared. It is worth noting that the recover of the robot motion is completely automatic and no reset or re-planning of the robot system is required.

# 3.3 Multi-Mobile-Manipulator Teleoperation System

## 3.3.1 Single-Operator-Multi-Robot Teleoperation System

The multiple mobile manipulator teleoperation System is shown in Figure 3.3. There are mainly two parts in the system: local operator controller and remote multi-robot system. The two parts are connected though any existing communication networks. The local operator controller is consisted of a human operator, some joystick controllers and some softwares and devices for telepresence. The remote multi-robot system contains multiple heterogeneous robots, which can be any kinds of robots with any kinds of system models. The multiple robots are also connected though any existing communication networks. The operator sends operation commands to the remote multi-robot system by operating the joystick controllers, and meanwhile, the remote environmental information is fed back to the operator and reproduced by the telepresence softwares and devices. The objective of the system design is to make the single operator be able to stably control the multiple robots to perform a specific coordinated task, which is to make the multi-robot group move along the trajectory sent by the operator and meanwhile satisfy some coordination requirements, like keeping a given formation.



Figure 3.3 Multiple Mobile Manipulator Teleoperation System

Since there is only one operator, in order to simultaneously tele-control multiple robots, the intelligence and autonomy of the robots need to be integrated into the teleoperation system. Besides, since the components in the system are connected though communication networks, the limitations of the networks may affect the stability and performance of the teleoperation system. For real communication networks, in particular for the Internet, the time delay is completely random, which is time-varying and can be neither bounded nor predicted. This limitation makes it much more difficult to realize the single-operator-multirobot teleoperation system.

# 3.3.2 Multi-Robot Teleoperation System in the Perceptive Reference Frame

The fact that time delay can affect the stability of the system is a result of using time as the reference for different system components. In other words, the control and feedback signals are being sampled with respect to time. Therefore, intuitively, if a non-time based reference is used, these effects of delays would be eliminated or reduced significantly. In traditional control systems, the systems are modeled by differential equations in which the reference is the time variable. In addition, the trajectory is usually a function of time. The general idea of the planning and control in the perceptive reference frame is to model the system and the trajectory as functions of a non-time based variable, which is usually denoted by s and called the non-time based reference [101]. The non-time reference is a map from the output or state of the robotic system to a scalar variable. This reference is usually taken to be a physical output of the system such as distance or position. However, the non-time based reference can also be a virtual one that does not correspond to any physical quantity. For example, it can be chosen to be the distance travelled by the robot or the number of control cycles the system has performed. Non-time based planning and control has been used in many studies and several robotic applications including mobile robots, manipulators, teleoperation and others [102] [2]. It has been proven to be very efficient in dealing with uncertainties and delays. In this paper, it is extended to the application of single-operator-multi-robot teleoperation with random communication delay.

The non-time based single-operator-multi-robot system is shown in Figure 3.4. The system can be divided into two parts. One is the teleoperation part where the operator observes the environmental information and sends operation commands to the multi-robot
system through communication network, and the other is the multi-robot coordination part where the multiple robots share information with each other and then coordinate together to move along the desired trajectory sent by the operator and meanwhile satisfy the coordination requirement such as a specific formation. All signals in the system are referenced to a non-time based reference s. Thus the random delay in communication networks do not have any effects on the signals because they are no longer referenced to the time t.



Figure 3.4 Non-time based single-operator-multi-robot teleoperation system

The teleoperation part is as same as the single-operator-single-robot teleoperation. The non-time based teleoperation method is applied for this part. This method has successfully been applied to the Internet-based teleoperation of many kinds of robotic systems including mobile robots and manipulators and been proved to be reliable and effective. The multi-robot coordination part is realized by the perceptive coordination method. The basic idea of perceptive coordination is to introduce a perceptive coordination reference r(s) that is direct-

ly computed based on the robot sensing measurements according to the desired trajectory and coordination requirement sent by the operator. The control input is parameterized by the perceptive coordination reference. Since the reference is a function of the measurements, the values of the desired robot states are functions of the measured data. As a result, for any given instant of non-time reference s, the desired input is a function of system output. This creates a mechanism to adjust or modify the robot plans based on the measurements. Thus, the planning becomes a closed loop process. Each robot's planner generates its desired values to the system, according to the on-line computed perceptive coordination reference parameter r(s). The commands from the operator can be executed by just affecting the perceptive coordination reference r(s) and the closed-loop planners of the robots.

In Fig. 2, the Perceptive Coordination Reference block maps the measurements of the robots to the perceptive coordination reference r(s). r(s) has to be chosen such that all the information of the robots can be properly represented in sense of achieving the coordination requirement. The mapping not only considers the system output of one robot, but also the mission of the coordination described by the system outputs of all other robots in the system. The motion of the robot in this system is coordinated by the common reference, r(s), which is related to the system outputs of all robots. Since each robot's planner is driven by r(s), the behavior of one robot in the formation will affect the mission of the coordination by affecting the motion reference r(s). For example, if the motion of one robot is stopped by an obstacle, this event will affect the computation of r(s) according to the specification of the coordination will affect the computation of the specification of the coordination by an obstacle, this event will affect the computation of r(s) according to the specification of the coordination scheme.

# 3.3.3 Design of Single-Operator-Multi-Robot Teleoperation System

In traditional control systems, the desired trajectory is usually modeled as a function of time t, which is generally denoted by  $x^d(t)$ . The variable x(t) represents the state of the system. Controllers are designed so that the trajectory x(t) of the system asymptotically approaches the desired trajectory  $x^d(t)$ . This is a typical tracking control problem. There exist a lot of control design methods in the literature of tracking control. Since the design is based on the model driven by t, the controller is likely to be time dependent. In non-time based control systems, the controller is independent of time. Furthermore, the desired path is not necessarily defined as a function of time. It is defined by parametric equations. The parameter is called non-time reference. It is denoted by s. A desired path can be defined by a parametric equation  $x^d(s)$ , where s is fusion of the sensory output of the robots in the formation.

The goal of the non-time based multi-robot teleoperation contains two parts. One is to make the multiple robots move according to the trajectory sent by the operator and the other is to make the multiple robots satisfy the coordination requirement such as keeping a given formation. Both two have to be fulfilled simultaneously and the communication is subject to random time delay.

The heterogeneous multi-robot system can be described by

$$\begin{cases} \dot{x}_i(t) = f_i(x_i(t), u_i(t)) \\ y_i(t) = h_i(x_i(t)) \end{cases} \quad i = 1, 2, ..., n$$
(3.2)

where i denotes the *i*th robot.  $x_i(t) \in \mathbb{R}^{n_i}$  denotes the state of the *i*th robot.  $u_i(t) \in \mathbb{R}^{m_i}$ 

denotes the input of the *ith* robot. And  $y_i(t) \in \mathbb{R}^p$  denotes the output of the *ith* robot.

The first step is to generate the desired trajectories according to the operator's commands. The operator's commands contain the moving velocity of the multi-robot group and a specific coordination requirement like a formation. Once the command is received, the desired trajectory of the multi-robot group with respect to can be computed by

$$\mathbf{y}^{\mathbf{d}}(s) = y_0 + \int\limits_0^s V(\tau) d\tau$$

where,  $y_0$  is the initial position of the multi-robot group, and V(s) is the operator's velocity with respect to s. Here s is the non-time reference which is a increasing function of time, s = v(t). In this design, it is chosen as the distance traveled by the multi-robot group. Afterwards, based on the trajectory, the desired motion of each robot can be computed according to the coordination requirement such as a specific formation.

The second step is to find a feedback law  $u_i = \mu_i(x_i, t)$  to track the desired path  $y_i^d(v(t))$ . There exist many well known tracking design algorithms in the literature of control theory. In this paper, the feedback law for the mobile robots used in the later experiment is based on feedback linearization and PD control [12]. The feedback needs to satisfy

$$\lim_{t \to \infty} (h_i(x_i(t)) - y_i^d(v(t))) = 0$$
(3.3)

The third step is to find a suitable projection to map the outputs of the robots to a perceptive coordination reference. The projection is described as a transformation  $r(s) = \Gamma_i(x)$  satisfying

$$\Gamma_i(x_d(s)) = s \tag{3.4}$$

where  $x^{d}(s) = \begin{bmatrix} x_{1d}^{T}(s) & x_{2d}^{T}(s) & \cdots & x_{nd}^{T}(s) \end{bmatrix}^{T}$ . Condition (3) implies that, if the system state is on the desired path,  $\Gamma_{i}(x(s))$  should give the corresponding value of s on the desired trajectory  $x_{d}(s)$ . If X is not on the desired path, then  $\Gamma_{i}(x(s))$  serves as the perceptive value of s for the system. For example, given any state x, let  $x_{di}(s)$  be the orthogonal projection from x to  $x_{di}$ . If we define  $\Gamma(x(s)) = s$ , then it satisfies (3).

The coordination can be expressed by the requirement that for any point along the desired path, the states of the robots should be in their corresponding desired positions. For given desired motion plans of the robots  $x_{di}(s)$ , i = 1, 2, ..., n, where s is the non-time variable and lies in the range of the non-time reference introduced in the first step, and the measurement states of the robots  $x_i$ , i = 1, 2, ..., n, the coordination criterion is defined as

$$J = \sum_{i=1}^{n} (x_{di}(s) - x_i)^T W_i(x_{di}(s) - x_i)$$
(3.5)

where  $W_i$ , i = 1, 2, ..., n are weight matrices. They can weigh the coordination errors in the different directions to insure efficient coordinated control in some specific directions, which are determined by the task. And then the value of the perceptive coordination reference r(s)can be computed by solving the following optimization problem min<sub>s</sub>  $\{J\}$ . The closed form solution can be easily obtained for most motion plans such as straight lines, arcs, etc., by solving

$$\frac{\partial J}{\partial s} = 0 \tag{3.6}$$

It can be seen that the planners driven by the perceptive coordination reference always give the optimal plan to minimize the coordination error.

The fourth step is to construct the control based on the perceptive coordination reference

for each robot. This step is simply a substitution. On the desired path, we have  $s = \Gamma_i(x_d(s)) = v(t)$ . Similarly, we define the perceptive time by

$$T_i(x) = v^{-1}(\Gamma_i(x))$$
 (3.7)

And then, the control law is defined by  $u_i(x) = u_i(x_i, T_i(x))$ . With this control law, the closed-loop system with non-time based feedback is described by

$$\dot{x}_i = f_i(x_i, u_i(x_i, T_i(x)))$$
(3.8)

The stability of this closed-loop system may be affected by the perceptive reference projection. The necessary condition for the system stability is discussed in [103].

The only issue left for the system design is to determine the non-time based reference according to the robot outputs. To ensure the stability of the system, the non-time reference must be a non-decreasing function of time. In our previous work, lots of variables have been used as the non-time reference according to different applications. Traveling distance was used as non-time reference to design a tracking controller for mobile robots and manipulators in [101]. Control cycle number was also used for Internet-based teleoperation system in [2]. These previous work involved only one robot and the non-time reference only depended on the feedback of the single robot. In the case of multi-robot system with communication constraints, choosing non-time reference is much more complicated. Besides that it contains multiple robots, the communication between the robots is also constrained by random time delays. So the previous non-time based references can not be used in this problem.

In multi-robot teleoperation systems, usually, making the multiple robots satisfy the co-

ordination requirements is more important than making them move along the trajectory given by the operator. For example, when the multiple robots are teleoperated to transport an object together, keeping the specific formations is the primary condition of tracking the desired trajectory. As we known, the perceptive coordination reference determines the desired motion plans of the multi-robot group. From the system design, it is seen that the non-time reference determines the measurement states of the robots, and furthermore determines the computation of the perceptive coordination reference. It is seen that the perceptive coordination reference is computed according to the choosing of the non-time reference. In order to ensure that the multiple robots have satisfied the coordination requirement before moving forward along the desired trajectory, the non-time reference can be chosen as the minimal distance traveled along the desired trajectory by all robots, which is described as

$$s = \min\{s_1, s_2, \dots, s_n\}$$
(3.9)

where  $s_i$  denotes the distance traveled along the desired trajectory by robot *i*. It can be described as the orthogonal projection from  $y_i$  to  $y_i^d$  and defined by

$$\mathbf{s}_{\mathbf{i}} = \arg\min_{s} \left\{ \left\| y_{i}^{d}(s) - y_{i} \right\| \right\}$$
(3.10)

Since every robot's desired motion is determined by the perceptive coordination reference which is computed according to the non-time reference, the design will ensure that every robot's desire motion is computed according to the robot which is the last one along the desired trajectory. Thus, if some robots move too fast, they will not move forward until the laggard robots catch up with them. This insures that all the robots can always keep their desired formation before moving forward along the desired trajectory.

## **3.4** Experimental Results

#### 3.4.1 Experimental Setup

Both the single-robot teleoperation and multi-robot teleoperation methods were implemented on the mobile manipulator teleoperation systems. For the single-robot teleoperation, both the holonomic mobile manipulator and nonholonomic mobile manipulator were used. The teleoperator and the mobile manipulators were connected through the Internet.

For the multi-robot teleoperation, two holonomic mobile manipulators were used. The teleoperation system architecture is shown in Figure 3.5. The teleoperator and the remote mobile manipulators communicated via the Internet and the tow mobile manipulators communicated with each other via the Wi-Fi. There was no assumption or knowledge regarding the delay characteristics in the networks. For better demonstrating the experiment, some random delays were randomly introduced to the two networks by the communication software. The delay was then shown to be completely random. This allowed for an evaluation of the effectiveness of the non-time based multi-robot teleoperation method with random communication delay.

#### 3.4.2 Experimental Results

The experiment task for the single holonomic mobile manipulator was to teleoperate the robot to remotely guide a blind person to stand up from a chair, move to another chair, and sit on the chair, as shown in Figure 3.6 (a). In this experiment, the teleoperator controlled



Figure 3.5 Architecture of single-operator-multi-robot teleoperation system

the mobile manipulator through a joystick while the blind person held the end-effector of the mobile manipulator. The holding forces were detected by the force/torque sensor on the end-effector. The forces were then transferred through the Internet and delivered to the teleoperator through the force feedback on the joystick. Therefore, it was like that the teleoperator guided the blind person by holding hands with him. Through the proposed teleoperation method, the teleoperator could successfully accomplished the assistance task and the system was always stable under random internet delays.

The experimental task for the single nonholonomic mobile manipulator was to teleoperate the robot to uncover a box, pick up an object, transport the object to another box and place it into the box, as shown in Figure 3.6 (b). The the teleoperator and the mobile



Figure 3.6 Single-robot teleoperation: (a) Blind guidance (b) Object manipulation

manipulator were connected through the Internet and the delays were completely random. The experimental results demonstrated that the proposed single-robot teleoperation method could enable the teleoperator stably accomplish the teleoperation task.

The experiment task for the multi-robot teleoperation was to tele-control two mobile manipulators to carry a long bar together and move from the initial position to a destination position and meanwhile avoid a still and known obstacle during the motion, as shown in Fiture 3.7 The multi-robot system was controlled by a single operator and the two robots were required to keep a rigid line formation during the motion. Besides, one unexpected obstacle was placed to block robot 1's motion during the experiment. The effectiveness of the proposed single-operator-multi-robot teleoperation system design was demonstrated.

The experimental results are shown in the figures blow. Figure 3.8 shows the x and y velocities sent by the single operator by observing the two robots' motion. Figure 3.9 shows the actual x and y trajectories of the two robots respectively. Figure 3.10 shows the position tracking errors of the two robots. Figure 3.11 shows the coordination error (line-formation



Figure 3.7 Multi-mobile-manipulator teleoperation task

keeping error) which is defined by

$$E_c = \sqrt{\sum_{i=1}^{2} \left( \left( x_i^d(s) - x_i \right)^2 + \left( y_i^d(s) - y_i \right)^2 \right)}$$
(3.11)

The communication delay between the teleoperator and the remote multi-robot system and the communication delay between the two robots are shown in Figure 3.12 respectively.

From the results, it is seen that the two-robot group were successfully tele-controlled by a single operator to move from the initial position to the destination and meanwhile avoid a still obstacle located at the motion path. Due to using the non-time reference for system modeling and control, the random communication delays had no influence on the stability of the system. Because of the coordination method using the perceptive coordination reference, the two robots could always keep the rigid line formation during the motion. The synchronized motion of the two robots is clearly shown in Figure 3.9. The average position tracking errors of the two robots were kept under 2 mm and the average coordination error was kept under 4 mm, as shown in Figure 3.10. The maximal coordination error was only about 8 mm, which was acceptable for the formation keeping of the two mobile manipulators. At about 140 s, robot 1 was stopped by an unexpected obstacle. It is shown in Figure 3.9 and Figure 3.10 that robot 1 stopped and therefore robot 2 also stopped because of the coordination scheme. The position errors and coordination error all stopped increasing. And the operator's commands were not executed at this time. At about 160 s, the obstacle was removed, the two robots recovered their motion autonomously according to the coordination scheme and the operator's commands autonomously took effect again from this time. There was no replanning or reset of the controller and the coordination scheme.

The experiments demonstrated the effectiveness of the designed single-operator-multirobot teleoperation system with random communication delay. Through the design, the single operator successfully tele-controlled the multiple robots to move to the destination, avoid a still obstacle, handle an expected obstacle and meanwhile keep the rigid line formation while the communication was subject to random time delays.

## 3.5 Chapter Summary

This chapter introduces the teleoperation of mobile manipulators under random communication delays and unexpected events. A non-time perceptive planning and control approach is applied. A non-time reference is used to replace the traditional reference of time. The stability problem caused by the time delays is eliminated when using the non-time reference for system modeling and control design. A coordination method based on the perceptive coordination reference are designed to make the single operator be able to tele-control the multiple robots under random delays and unexpected events. Experiments on real mobile manipulator teleoperation systems demonstrated the effectiveness of the proposed methods.



Figure 3.8 Teleoperator's velocity commands



Figure 3.9 Trajectories of the two robots



Figure 3.10 Position errors of the two robots



Figure 3.11 Coordination error



Figure 3.12 Random time delays

## Chapter 4

# Teleoperation of Mobile Manipulators Adaptive to Quality of Teleoperator (QoT)

## 4.1 Introduction

Extensive studies have been conducted on telerobotic operations for decades due to their widespread applications in a variety of areas. Most studies have been focused on two major issues: stability and telepresence. Few have studied the influence of the operation status of the teleoperator on the performance of telerobotic operations. As subnormal operation status of the teleoperator may result in insufficient and even incorrect operations, the quality of teleoperator (QoT) is an important impact on the performance of the teleoperator of the teleoperator of the teleoperator is an important impact on the performance of the teleoperator are guaranteed.

Therefore, this chapter investigates the online identification of the QoT and its application to telerobotic operations. The QoT is identified based on five QoT indicators which are generated based on the teleoperator's brain EEG signals. A QoT adaptive control method is designed to adapt the velocity and responsivity of the robotic system to the operation status of the teleoperator such that the teleoperation efficiency and safety can be enhanced [104][105]. The online QoT identification method was tried on various teleoperators and the QoT adaptive control method was implemented on a mobile manipulator teleoperation system. The experimental results demonstrated the effectiveness and advantages of the proposed methods.

## 4.2 Teleoperation System Adaptive to QoT

## 4.2.1 Quality of Teleoperator (QoT)

The quality of teleoperator (QoT) reflects the level of confidence in the teleoperator's commands. It can be applied for improving the telerobotic operations from various aspects. For example, the QoT can be used to reinforce the efficiency and safety of the telerobotic systems by adapting the robot velocity and responsivity to the teleoperator's operation status by changing the planner and controller parameters of the robotic systems. In addition, QoT can be used for the shared control between the autonomous robot controller and the human teleoperator through the QoT based command fusion. Besides, QoT can be used for the telerobotic systems. Since the requirements of resources for the teleoperator may differ as the QoT changes, the resources such as the sensory information, multi-media, network bandwidth and many others can be online allocated based on the QoT.

Therefore, there are a number of potentials to enhance the teleoperation performance through integrating the QoT into the telerobotic systems.

### 4.2.2 Architecture for QoT based Teleoperation

The general framework of the QoT based teleoperation system is shown in Figure 4.1. The perceptive planning and control method and a QoT based adjustment mechanism are designed to construct the teleoperation system.



Figure 4.1 QoT based teleoperation system framework

The perceptive planning and control method is applied in order to guarantee the stability of robot system under network delays [54]. However, the efficiency and safety of telerobotic operations are still affected by the quality of teleoperator. Especially, the effects of the low QoT become more evident when the teleoperator is performing tasks which require high dexterity.

In order to address this issue, a QoT based adjustment mechanism is proposed. The QoT is obtained by an online QoT generator and then sent to the robotic systems associated with the teleoperator's commands. Based on the QoT, the parameters of the robot planner and controller can be adaptively adjusted to assign appropriate velocity and responsivity for the robotic system. Intuitively, with a high QoT, the robotic system needs to perform quickly and responsively, so that the teleoperator can efficiently control the robot and in turn achieve the teleoperation goal as fast as possible. On the other hand, if the QoT is low, the robotic system needs to perform relatively slowly with a decreased responsivity in order

to make the teleoperator be able to commendably perform the operations so as to reduce or avoid the potential influence of the insufficient and incorrect commands.

Therefore, with this QoT based adjustment mechanism, the telerobotic system can be adaptively adjusted to enhance the efficiency and safety.

## 4.3 Online Identification of QoT

#### 4.3.1 QoT Indicators

The QoT indicator is defined to represent a specific state of the teleoperator. The QoT indicators could come from the mental states of the teleoperator such as excitement and boredom and from the physiological states of the teleoperator such as heartbeat and blood pressure. In this paper, we identify the QoT using only mental states of the teleoperator. Five QoT indicators generated from the Emotiv SDK are used including the long-term excitement, short-term excitement, boredom, meditation and frustration which are introduced separately as below.

*Long-term excitement:* a mental state representing being emotionally aroused, agitated or worked up. It could describe the cerebral indefatigability of the teleoperator.

Short-term excitement: a momentary mental state representing being emotionally aroused, agitated or worked up. It could describe the momentary excitement and vitality of the teleoperator.

*Boredom:* an unpleasant and transient affective state of feeling a pervasive lack of interest and having difficulties to concentrate on current activities. It could describe the mental distraction of the teleoperator.

Meditation: an internal personal thinking behavior without external involvement. It

could describe the concentration of the teleoperator.

*Frustration:* a common emotional response to the opposition or failure. It could describe the teleoperator's perceived deficiencies and lack of confidence in accomplishing the tasks.

## 4.3.2 Model for QoT Identification

The EEG brain headset used in the study is an Emotiv EPOC Neuroheadset as shown in Figure 4.2. The overall process for the QoT identification is shown in Figure 4.3. There are three steps: EEG signal acquisition, QoT indicator generation and QoT generation.

The EEG signal acquisition is realized by the Emotiv EPOC Neuroheadset. It contains 14 saline sensors and can collect EEG signals from 14 channels including AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8 and AF4 by touching them to the scalp of the teleoperator.

The QoT indicator generation is realized by the Emotiv Software Development Kit (S-DK). It can recognize the five QoT indicator values of the teleoperators by analyzing their 14 EEG signals [106]. The recognition function of this SDK has been proved to be effective and successfully implemented in various applications [107][108].

According to the developers, the recognition model was built based on the empirical data of a wide diversity of human subjects. The subjects were required to do various activities such as playing games and watching movies while their EEG data and physiological information were recorded including the heart rate, blood pressure, respiration, skin conductance, etc. They were then required to complete questionnaires about their experiences and feelings. Psychologists evaluated the affective states and labeled the EEG data by different affective states. Specific features to the affective states were designed by analyzing all data and then used to build the recognition model. The model was also designed to perform self-scaling in real-time running to range the affective state values in [0, 1] for the current user.



Figure 4.2 Brain headset for EEG signal acquisition



Figure 4.3 QoT generation model

The last step is the QoT generation based on the five QoT indicators. As the relationship between the QoT and five QoT indicators involves complicated knowledge in various fields such as psychology, physiology, etc., it creates a complex web of data and makes it very difficult and even impossible to use an analytic model to represent this relationship. Fortunately, the neural network is an alternative way to approximate the relationship. The B-spline neural network [109], used in this research, is a type of basis function neural network which has been proved to be an effective and efficient way to model the unknown and complicated relationship. One of its main advantages is the feasibility of controlling the basis function curves. As the curves only change in the vicinity of the modified control points, it provides convenient means of control for the users to design the neural network.

The designed B-spline neural network takes the five QoT indicators as the input to output the QoT. It consists of three layers: a normalized input space layer, a basis functions layer and a linear weight layer. The value of QoT is computed based on p multivariate basis functions defined on the five QoT indicators, x. The *jth* multivariate basis functions of order k is denoted by  $N_k^j(x), j = 1, ..., p$ . These multivariate basis functions are formed by the tensor product of a set of univariate basis functions defined on each independent QoT indicator. The *jth* univariate basis function of order k for  $x_i$  is defined by

$$N_{k,i}^{j}(x_{i}) = \left(\frac{x_{i} - \lambda_{j-k}}{\lambda_{j-1} - \lambda_{j-k}}\right) N_{k-1,i}^{j-1}(x_{i}) + \left(\frac{\lambda_{j} - x_{i}}{\lambda_{j} - \lambda_{j-k+1}}\right) N_{k-1,i}^{j}(x_{i})$$

$$N_{1,i}^{j}(x_{i}) = \begin{cases} 1 & , if \ x \in I_{j} \\ 0 & , otherwise \end{cases}$$

$$(4.1)$$

where  $x_i$  is the input variable denoting a specific QoT indicator,  $\lambda_j$  is the *jth* knot and  $I_j$  is the *jth* interval of  $x_i$  defined in the normalized input space. The output of the B-spline neural network is the value of the QoT which is computed by a weight sum of these multivariate basis functions and described by

$$QoT = \phi(x) = \sum_{j=1}^{p} w_j N_k^j(x)$$
(4.2)

where  $w_i$  is the weights needing to be determined by the training.

## 4.3.3 Training for QoT Identification Model

The major challenge for the QoT generation using the proposed model is the training of the neural network. The training process is shown in Figure 4.4.

Two important issues of the training are the acquisition of the neural network input and output and the parameter identification algorithm. The input of the neural network contains 5 QoT indicators which can be captured using the Emotiv SDK based on teleoperator's EEG signals. These measurements are represented by normalized values in the range of [0, 1]. The larger value represents the stronger corresponding QoT indicator and the smaller value represents the inverse.

The output of the neural network is the value of the QoT. It is also represented by a normalized variable in the range of [0, 1]. Different from the input, it is impossible to be directly captured. Thus, an indirect way has been designed to obtain this value.

The QoT was obtained based on the performance of accomplishing some designed teleoperation tasks. A real mobile manipulator teleoperation environment and a virtual teleoperation environment were built. A number of teleoperators were selected to accomplish the designed teleoperation tasks at various times in the two environments. They performed these tasks, such as the path tracking, manipulation, obstacle avoidance and unexpected event handling, while observing the visual environmental feedback. The teleoperators had been trained before so they were familiar with these teleoperation tasks. The task accomplishment performance mainly depended on the operation status of the teleoperators.

The observations of the task accomplishment performance were recorded while the tasks were performed including the time cost, average tracking errors, maximal tracking errors, collision times with obstacles, reaction time to unexpected events, etc. By statistical analysis of all these observations, their ranges could be obtained. Based on these ranges, all observations were normalized into the range of [0, 1]. Then, a QoT evaluation scheme, which was a weighted average of all normalized observations, was used to fuse these normalized observations into the QoT. The weights were predefined according to the importance of the observations to the task accomplishment. Finally, a QoT value in [0, 1] was generated to represent the operation status of the teleoperator.



Figure 4.4 Training of QoT generation model

After the input and output data of the neural network are obtained, many algorithms can be applied to train the B-spline neural network. In our study, we choose the Levenberg-Marquardt Algorithm (LMA) to perform the training.

## 4.4 QoT based Adaptation for Telerobotic Operations

#### 4.4.1 QoT based Adaptation

High QoT indicates high alertness of the teleoperator and low QoT indicates the inverse. With a high QoT, the teleoperator can sensitively observe environmental feedback and be alert to the environmental changes. In contrast, low QoT reflects subnormal operation status of the teleoperator. Therefore, adjusting the velocity and responsivity of the robotic system according to the QoT can help to improve the teleoperation performance. However, the adjustment according to only QoT cannot always make the system efficient. For instance, when the teleoperator is completing some simple tasks, such as driving a robot along a straight line at a low speed, a low QoT may be sufficient for the teleoperator to accomplish the task.

To address this issue, in the QoT adaptive control design, a concept named task dexterity index (TDI) [110] is introduced to represent the dexterity and complexity of the task. A collection of task indicators are generated online based on the teleoperator's control commands and the sensory information from the robot. By using a fuzzy inference scheme, these task indicators are mapped into a scalar TDI in the range of [0, 1]. The fuzzy mapping function is described by

$$TDI = f(u) = \sum_{j=1}^{M} a_j g_j(u_j) / \sum_{j=1}^{M} a_j$$
(4.3)

where  $u_j$  is the task indicator, M is the number of task indicators,  $g_j$  is the mapping function of  $u_j$ , and  $a_j$  is the relative weight of  $u_j$  in the task which is determined base on the importance of a particular task indicator.

In order to enhance the teleoperation performance, the robot performance can be adjusted according to the QoT and TDI together to make the teleoperator be able to efficiently and safely operate the robot to accomplish different tasks at different operation status. The adjustment contains two parts: velocity adjustment and responsivity adjustment of the robot.

For the robot velocity adjustment, the velocity gain  $k_{is}$  is used, which maps the teleop-

erator's commanded velocity to the robot velocity in the ith direction. A linear mapping function is adopted in the design which is described by

$$v_{i,r} = k_{is} v_{i,tele} \tag{4.4}$$

where  $v_{i,r}$  and  $v_{i,tele}$  are the robot velocity and the teleoperator's commanded velocity in the *i*th direction respectively. A larger velocity gain of  $k_{is}$  leads to a relatively larger robot velocity and a smaller velocity gain leads to the inverse.

For robot responsivity adjustment, the controller gains of the robotic system are used. For a robotic system where nonlinearities exist, by applying some nonlinear and decoupling techniques, such as the nonlinear feedback control (Tarn et al., 1984), the system models can be linearized and decoupled into several independent linear subsystems. Then, a PD controller can be introduced to control the subsystems. For each controller, there are two gains  $[k_{i1}, k_{i2}]^T$ . These gains are proportional to the robot responsivity, which makes them applicable for the responsivity adjustment.

At the beginning, a set of nominal gains  $\tilde{K}_i$  are predefined for both the velocity gains and controller gains. The nominal velocity gains are chosen as 1 and the nominal controller gains are chosen to achieve the best possible responsivity of the robotic system based on the experimental tests. Then, based on the QoT and TDI, the desired gains can be computed by

$$\mathbf{K}_{i}^{d} = \begin{cases} \tilde{\mathbf{K}}_{i}, & if \ QoT > \gamma \ or \ TDI < D_{L} \\\\ \lambda \tilde{\mathbf{K}}_{i} + (1 - \lambda) \tilde{\mathbf{K}}_{i} * QoT * (1 + D_{H} - TDI) , \\\\ & if \ QoT \le \gamma \ \& \ D_{L} \le TDI \le D_{H} \\\\ \lambda \tilde{\mathbf{K}}_{i} + (1 - \lambda) \tilde{\mathbf{K}}_{i} * QoT , \\\\ & if \ QoT \le \gamma \ \& \ TDI > D_{H} \end{cases}$$
(4.5)

where  $K_i^d$  and  $\tilde{K}_i$  are the desired and predefined nominal gains separately,  $\gamma$ ,  $D_H$  and  $D_L$  are threshold constants, and  $\lambda$  is a weight used to adjust the influence of the QoT on the robot controller gains. These parameters are all defined based on the teleoperation requirements and the teleoperator's experience.

The first part of the formula means that the robot should perform as fast as possible with high responsivity when the QoT is high enough or the task is simple enough. The second part means that the robot should perform suitably fast or slow with suitable responsivity according to both the QoT and TDI when the QoT is low and the task is intermediately complicated. The third part means that the robot should perform with a appropriately low velocity and low responsivity adaptively to only the QoT when the QoT is low and the task is very complicated. During the switchings, the gains drop quickly when QoT becomes low or the task becomes very complicated to immediately decline the robot velocity and responsivity. It may cause some instantaneous jerkings of the robot velocity and responsivity during the switchings. In some applications where the smooth velocity and responsivity are required, the following adaptive gain adjustment can be used

$$\mathbf{K}_{i}^{d} = \begin{cases} \tilde{\mathbf{K}}_{i}, & if \ QoT > \gamma \ or \ TDI < D_{L} \\ \lambda \tilde{\mathbf{K}}_{i} + (1 - \lambda) \tilde{\mathbf{K}}_{i} * \max\left\{\frac{QoT}{\gamma}, \ \frac{D_{H} - TDI}{D_{H} - D_{L}}\right\}, \\ & if \ QoT \le \gamma \ \& \ D_{L} \le TDI \le D_{H} \\ \lambda \tilde{\mathbf{K}}_{i} + (1 - \lambda) \tilde{\mathbf{K}}_{i} * \frac{QoT}{\gamma}, \\ & if \ QoT \le \gamma \ \& \ TDI > D_{H} \end{cases}$$
(4.6)

In this adjustment, the switchings of the gains are always continuous so that the changing of the robot velocity and responsivity is smooth.

#### 4.4.2 Performance Analysis and Discussions

Autonomous interventions on teleoperation systems have been studied recently. Many studies showed that introducing the robot autonomy to teleoperation to take over partial or full control of the robot could help to make the telerobotic operations more efficient. In these studies, the robot autonomy was introduced in various ways such as full-autonomy [111], semi-autonomy [112] and shared-autonomy [113]. However, some other studies showed that the teleoperators preferred a manual mode rather than the autonomous mode although the autonomous mode could help to reduce their efforts required for the operations [114]. In the manual mode, they had a higher level of satisfaction when they had a full control of the robot and gained more learning effects during the teleoperation. Dragan et al. argued that the robot autonomy should be introduced in a user-customized way [115].

Our QoT adaptive control design is different from the exiting autonomous intervention approaches for teleoperation. The autonomy that we introduce to teleoperation is not aimed at taking over partial or full control of the robot. Instead, the teleoperator always has a full control of the robot. The autonomy is just designed to adjust the robotic system performance such as the velocity and responsivity adaptively to the operation status of the teleoperator. The telerobotic operations can be benefited from this adaptation, and at the same time, the satisfaction and learning effects of the teleoperator will not be attenuated much.

After the desired gains  $K_i^d$  are generated, we can directly apply the velocity gains because they will not affect the underlying control system. However, for the controller gains, we can not directly apply them because they may affect the system performance. The efficient region of the controller gains was investigated in [116] and can be described by

$$\Omega: \begin{cases} 0 < k_{i2} < \frac{k_{i1}\tau_k(T-\tau_k)+2}{T-2\tau_k} \\ \alpha(k_{i1},k_{i2})\beta(k_{i1},k_{i2}) > 0 \\ \frac{1}{2}\tau_k k_{i1} - \frac{1}{\tau_k} < k_{i2} < \frac{1}{2}\tau_k k_{i1} + \frac{1}{\tau_k} \end{cases}$$
(4.7)

where T is the sampling period,  $\tau_k$  is the time delay, and

$$\alpha(k_{i1}, k_{i2}) = -\tau_k^2 (T - 2\tau_k) k_{i1}^2 + 4\tau_k k_{i2}^2 + 2\tau_k (T - 3\tau_k)$$

$$k_{i1}k_{i2} + 2(T + 2\tau_k)k_{i1} - 4k_{i2}$$

$$\beta(k_{i1}, k_{i2}) = \tau_k^2 (T^2 - 2T\tau_k + 2\tau_k^2)k_{i1}^2 + 4\tau_k (2\tau_k - T)k_{i2}^2$$

$$+ 2\tau_k (3T\tau_k - 4\tau_k^2 - T^2)k_{i1}k_{i2} - 2T(T + 2\tau_k)k_{i1} + 4Tk_{i2} - 8$$

Then, a complete controller gain adjustment process is designed as follows. At the beginning, the desired controller gains  $K_i^d = \begin{bmatrix} k_{i1}^d, k_{i2}^d \end{bmatrix}^T$  are computed based on the QoT and TDI by equation (5) or (6). They are then examined whether they lie in the region defined by  $\Omega$ . If they lie in the region, the desired gains are then set as the controller gains. Otherwise, the gains which lie in the region  $\Omega$  and are the nearest to the desired gains  $K_i^d$  in the Euclidean space of  $k_{i1} - k_{i2}$  plane are selected.

This gain selection process can be considered as a quadratic programming problem. In

this problem, the constraint set is constructed from the conditions  $\Omega$  which contains two linear constraints and a hyperbolic constraint:  $\alpha(k_{i1}, k_{i2})\beta(k_{i1}, k_{i2}) > 0$ . This hyperbolic constraint can be approximated by three linear boundaries in which two of them comes from the asymptotes of the hyperbola and the left one comes from a line that passes through the focus and is perpendicular to the major axis of the hyperbola. Therefore, using this approximation, all conditions described in  $\Omega$  are transformed to be linear. Then, the selection of the gains can be expressed by a quadratic programming problem described by

$$\min_{K_{i}} \quad \frac{1}{2} \mathbf{K}_{i}^{T} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \mathbf{K}_{i} - 2 \begin{bmatrix} k_{i1}^{d} & k_{i2}^{d} \end{bmatrix} \mathbf{K}_{i}$$

$$st. \quad \mathbf{A}_{C} \mathbf{K}_{i} \leq \mathbf{B}_{C}$$

$$(4.8)$$

where  $A_C$  and  $B_C$  are coefficient matrices of the linear constraints derived from conditions  $\Omega$ . The controller gains can be obtained by solving this well-known quadratic optimization problem using many existing algorithms [117].

## 4.5 Experimental Results

#### 4.5.1 Results of Online QoT Identification

In order to illustrate the effectiveness of the online QoT identification method, three teleoperators were selected as the subjects. They were required to wear the brain headset and their QoT data were recorded at different time periods. The average QoT results for the three teleoperators are presented as below.

The average QoT and QoT indicators of the teleoperator 1 (Sex: male; Age: 29; Place:

East Lansing, Michigan, USA) for a day are shown in Figure 4.5. He woke up at 7 AM and stayed up until 6 AM of the next day. It is seen that the teleoperator had a low QoT when he woke up. The value increased quickly following breakfast. After 11 AM, the QoT began decreasing due to hunger. Following lunch and a brief time later, the value kept increasing and reached the highest around 2-3 PM. Afterwards, the QoT varied slightly and remained in a normal range till 7 PM. After dinner at 8 PM, the QoT value increased slightly and then remained in a relatively low range. After midnight, because the teleoperator still remained awake, it is seen that the QoT kept decreasing as time increasing. The QoT values after midnight were in a very low-value range.

The teleoperator 2 (Sex: female; Age: 32; Place: East Lansing, Michigan, USA) and teleoperator 3 (Sex: male; Age: 31; Place: Hong Kong, China) were also selected to conduct the test. Their QoT data were recorded hourly from 9 AM to 11 PM and the average QoT data are shown in Figure 4.6. The upper one shows the QoT of the teleoperator 2 and the lower one shows the QoT of the teleoperator 3. We can see that the changes of the QoT of the two teleoperators followed roughly similar trends as the teleoperator 1. For example, they all had higher QoT when being energetic in the morning and afternoon and lower QoT when they felt tired at night. These QoT data were also coincident with the self assessment of these three teleoperators.

In order to illustrate the relationship between the QoT and the teleoperator's operation status, a couple of experiments were performed. A virtual environment was constructed as shown in Figure 4.7. The teleoperator 1 was required to use a joystick to control a virtual manipulator to track the edge of a disk. While he was conducting the task, his QoT was online recorded. At the same time, the performance measurements of accomplishing the task were also recorded including the time cost and the average and maximal tracking errors. The



Figure 4.5 QoT (left) and QoT indicators (right) in a day



Figure 4.6 QoT of two teleoperators: a female at MSU (upper) and a male in HK (lower)

test was carried out at three different time periods when the teleoperator had different QoT values. For each time period, the test was completed 5 times. The statistic results are shown in Table 4.1. The data in the parentheses are the ranges of the QoT and the performance measurements. It can be seen that when the QoT was high, the teleoperator could finish

Time	Avg. QoT	Avg. error	Max. error	Time cost
		(mm)	(mm)	(s)
9-10 AM	0.6412	18.42	24.42	140.6
		(16.15 - 21.63)	(18.29-26.85)	(131-149)
2-3 PM	0.7605	12.32	18.04	122.2
		(10.13-14.68)	(15.76-22.01)	(121-124)
1-2 AM	0.3477	26.94	40.33	168.2
		(21.86-31.99)	(29.50-54.54)	(162-185)

Table 4.1 Statistical analysis of QoT and corresponding teleoperation performance

the task in relatively shorter time with smaller tracking errors. Otherwise, the teleoperator needed a longer time and generated bigger tracking errors.

The Pearson product-moment correlation coefficients were calculated to assess the relationship between the QoT and the performance measurements including the average error, maximal error and time cost. They were reported as r(13) = -0.81, p < .005, r(13) = -0.75, p < .005 and r(13) = -0.89, p < .005 respectively. It is seen that there was a strong correlation between the QoT and the three performance measurements.



Figure 4.7 Experimental setup for QoT verification

The relationship between the QoT and operation status was also illustrated through the

experiments on a mobile manipulator teleoperation system. The mobile manipulator consisted of a four-wheel Segway RMP400 mobile base and an onboard 7-DOF Schunk manipulator. The experiment was to control the mobile manipulator to track a rope. It was performed by the teleoperator 3 at ten time periods in the afternoon. The average QoT and average performance evaluation were plotted in Figure 4.8. The performance evaluation for each test was a value in [0, 1] and computed by a weighted average of the normalized performance observations including the time cost, average tracking error and maximal tracking error. From the result, we can see that the QoT was basically consistent with the performance evaluation of the teleoperator although there were some small variances.

The Pearson product-moment correlation coefficient between the two sets of data was reported as r(8) = .80, p < .01, which illustrated the correlation between the QoT and performance evaluation.

After statistically analyzing the data of these subjects, it is seen that the average variance of their QoTs is about 0.08 for different subjects with similar performances. Although different subjects may have different QoT indicators and QoTs, their QoTs are all in the similar range when they have similar performances.

### 4.5.2 Results of QoT Adaptive Teleoperation

The QoT adaptive control design was implemented on a mobile manipulator teleoperation system. The system is shown in Figure 4.9. The robotic system was located at the Michigan State University in East Lansing, USA. It was similar to the mobile manipulator at the City University of Hong Kong. The teleoperators were located at both the Michigan State University in East Lansing and the City University of Hong Kong in Hong Kong. The teleoperator controlled the manipulator using a 6-DOF space-ball joystick while observing



Figure 4.8 QoT identification results of the tracking experiment

the environmental information through the video feedback. The communication between the teleoperation station and the robot station was realized by the Internet. The average round trip time delays were about 3 ms and 300 ms for teleoperating the robot in East Lansing and Hong Kong respectively.

Three teleoperation tasks were designed to demonstrate the effectiveness of the designed method including the manipulation task, navigation task and mobile manipulation task. The manipulation task was performed by the teleoperator 1 in East Lansing and the other two tasks were performed by the teleoperator 3 in Hong Kong.

For each task, the traditional teleoperation method and the proposed QoT based teleoperation method were used separately. In the traditional teleoperation method, the perceptive planning and control method was applied to ensure the stability of the robotic system under communication delays (Xi and Tarn, 2000). The velocity gains were set as 1 and the controller gains were chosen to be practically best for the responsivity of the robotic system based on experimental tests. In these tests, we applied different controller gains and chose the ones which made the system have the shortest response time to move from zero velocity to a given velocity without visible vibration. The only difference between the proposed method and the traditional method was to introduce the QoT adaptive control or not. All other conditions between these two methods remained the same. The experimental results are presented as below.



Figure 4.9 Experimental setup for Internet based teleoperation



Figure 4.10 Three teleoperation tasks: (a) manipulation; (b) navigation; (c) mobile manipulation

The manipulation task was to operate the manipulator to pick up a bottle at location A, transport it to location B and then place it in a small cylinder at location B, as shown in Figure 4.10 (a). The experiments were performed at midnight when the teleoperator 1
Method	Pick (s)	Transport (s)	Place (s)	Total (s)
Traditional	130	50	120	300
QoT	85	60	65	210
TDI+QoT	78	40	62	180

Table 4.2 Time cost comparison of using different methods

was sleepy and physically exhausted. Three kinds of experiments were performed to demonstrate the effectiveness of the QoT adaptive control. The first experiment used the traditional method without any adjustment. The second experiment used only the QoT for the adjustment and set the TDI as a constant of 1. The third experiment used both the QoT and TDI for the adjustment. All experiments were conducted 5 times. The comparison of the average time cost when using these three methods is shown in Table 4.2.

It can be seen that, compared to the traditional method without any adjustment, the control adjustment based on only the QoT in the second experiment could reduce the total time cost of the entire task and the concrete time cost of the subtasks including the picking, transportation and placing. Furthermore, after using the adjustment based on both the QoT and TDI in the third experiment, the time cost was further reduced, especially for the transportation subtask which was relatively simple and did not require a high QoT.

The QoT and QoT indicators of the teleoperator are plotted in Figure 4.11. It is seen that the QoT was generally low at midnight due to the sleepiness and tiredness. The QoT values at the beginning and in the end were larger than the middle. This was because the teleoperator paid more attention to picking and placing the bottle than transporting it. Figure 4.12 shows the TDI plot. It can be seen that the transportation was much simpler compared to the picking and placing. To compare these three kinds of experiments in detail, three representative results of the experiments are presented respectively. Figures 4.13, 4.14 and 4.15 show the position trajectories and the *y*-axis velocity trajectories of the manipulator in these three experiments.

From the plots in Figure 4.13, we can see that without the adjustment the teleoperator accomplished the task circuitously with frequent target-missing behaviors, which led the teleoperator to take a longer time and sometimes caused collisions between the manipulator and the objects.

Figure 4.14 shows the result of using only the QoT for the control adjustment. The robot behaviors changed according to only the value of QoT. We can see that the teleoperator accomplished the task smoothly and efficiently with a significantly shorter time. The teleoperator could accomplish the picking and placing tasks much smoother with this adjustment. For the transportation portion in the middle of Figure 4.14, the task was simple but the robot still moved slowly because of the low QoT. Therefore, this portion of task took a long time, which was actually not necessary because the QoT was high enough for the teleoperator to complete this simple transportation task.

Figure 4.15 shows the result of using both the QoT and TDI. Besides the smooth and efficient operations, it can be seen that the robot moved faster while accomplishing the simple transportation task even though the QoT was low. This further improved the efficiency of the teleoperation. The time cost to finish the entire task with this method was the shortest among these three experiments.

In addition, the ANOVA (Analysis of variance) was conducted on the time cost data of the traditional method and the method using both QoT and TDI. The result was reported as F(1,8) = 35.67, p = .0003. It indicated that the reduction of the time cost for the manipulation task was significant when using the proposed QoT adaptive control method.

The navigation task was to operate the mobile base to move from location A to location



Figure 4.11 QoT indicators and QoT in manipulation task



Figure 4.12 TDI plot in manipulation task

B, as shown in Figure 4.10 (b). There was an obstacle between the two locations so that the teleoperator 3 would need to operate the mobile base to avoid the obstacle while controlling it via the Internet in Hong Kong. The experiment was performed at three different time



Figure 4.13 Robot movement without the adjustment

periods: 9-10 AM, 2-3 PM and 10-11 PM. During each time period, the task was completed 5 times and the QoT was recorded at each test. The average QoT was then computed in each time period. The time cost and collision times between the mobile base and the obstacle were recorded as well for each test.

The comparison of the results of using the traditional teleoperation method and proposed teleoperation method are shown in Table 4.3 and Table 4.4. We can see that the QoT was highest during 2-3 PM, intermediate during 9-10 AM, and lowest during 10-11 PM. From the time cost and collision times of using the traditional method at different time periods, we can see that the performance of the teleoperator was consistent with the value of QoT. Basically higher QoT could result in better performance in terms of shorter time cost and less collision times. When comparing the results of the two methods, we can see that both



Figure 4.14 Robot motion with the adjustment using QoT

time cost and collision times were significantly reduced for all time periods when using the proposed method. The average time cost was reduced by about 30 percent and the average total collision times were reduced by about 90 percent.

The ANOVA was conducted on the data of the time cost and collision times for the traditional method and proposed method. The results were reported as F(1, 28) = 14.99, p =.0006 for the time cost and F(1, 28) = 6.16, p = .02 for the collision times. The results suggested that the reduction of the time cost and collision times was significant for the navigation task by using the proposed method.

The mobile manipulation task was to operate the mobile manipulator to move from location A to location B, pick up an oblique bottle at location B, transport the bottle to location C, and then place the bottle into a very small cylinder at location C, as shown



Figure 4.15 Robot motion with the adjustment using QoT and TDI

Time	Avg. QoT	Method	Test 1	Test 2	Test 3	Test 4	Test 5	Avg. cost (s)
0 10 AM 0 5836	0 5836	Trad.	122	112	93	130	115	114.4
<i>J</i> -10 AM	0.0000	Prop.	105	100	92	82	91	94.0
2.2 DM 0.6527	0.6527	Trad.	93	103	133	119	108	111.2
2-5 1 M	0.0521	Prop.	82	81	68	53	81	73.0
10 11 PM	0 4475	Trad.	206	200	147	158	149	172.0
10-11 1 101	0.4410	Prop.	137	122	96	92	104	110.2

Table 4.3 Time cost comparison in navigation task

in Figure 4.10 (c). In addition to following these operation procedures, the teleoperator 3 would also need to avoid three types of collisions while controlling the mobile manipulator via the Internet in Hong Kong, including the collision of the mobile base with the obstacle in the navigation portion, the collision of the gripper with the bottle in the pick-up portion, and the collision of the bottle with the cylinder in the placing portion. Notice that the

Time	Method	Test 1	Test 2	Test 3	Test 4	Test 5	Avg. times
9-10 AM	Trad.	1	1	0	0	1	0.6
<i>5</i> -10 AM	Prop.	0	0	0	0	0	0.0
2 3 PM	Trad.	0	1	0	0	0	0.2
2-3 1 M	Prop.	0	0	0	0	0	0.0
10 11 PM	Trad.	0	3	1	0	2	1.2
10-11 I MI	Prop.	0	1	0	0	0	0.2

Table 4.4 Collision comparison in navigation task

mobile manipulation task was not a simple combination of the manipulation task and the navigation task. It required the teleoperator to use both the manipulator and mobile base at the same time to accomplish the mobile manipulation task. The mobile platform needed to be placed in an appropriate position for finishing the following subtasks and both the motions of the manipulator and mobile base were necessary for accomplishing the task. Thus, the requirements for the teleoperator were higher than the previous two tasks.

The experiment was performed five times as well during each time period of 9-10 AM, 2-3 PM and 10-11 PM. For each test, the QoT, time cost and collision times were all recorded. The average QoT was computed in each time period. As the cylinder was very small, the teleoperator usually required more than one time to finish the placing task. So this attempt times were also recorded for each test. The comparison of the results of using the traditional and proposed teleoperation methods are shown in Table 4.5 and Table 4.6.

It can be seen that with the proposed method the teleoperation performance was significantly improved in all three time periods and each test. The time cost was reduced in all tests compared to the traditional teleoperation method and the collision times in navigation, pick-up and placing were all reduced as well. Besides, the attempt times to accomplish the placing task were also reduced when using the proposed method. The average time cost was

Time	Avg. QoT	Method	Test 1	Test 2	Test 3	Test 4	Test 5	Avg. cost (s)
9-10 AM 0 5988	Trad.	627	609	514	523	714	597.4	
<i>J</i> -10 AW	0.0900	Prop.	434	409	380	354	331	381.6
9.2 DM	0.6036	Trad.	625	569	449	584	750	595.4
2-3 I WI	0.0050	Prop.	383	302	255	375	336	330.2
10 11 PM	0.4840	Trad.	621	781	802	782	840	765.2
10-11 1 1/1	0.4040	Prop.	385	436	434	309	317	376.2

Table 4.5 Time cost comparison in mobile manipulation task

reduced by about 45 percent. The average total collision times were reduced by about 80 percent and the average attempt times of placing were reduced by about 40 percent. The videos of the experiments of using the traditional method and the proposed method are shown in the multimedia extension 1 and extension 2 respectively.

The ANOVA was conducted on the date of the time cost, collision times and attempt t times for the traditional method and proposed method. The results were reported as F(1,28) = 52.74, p < .0001 for the time cost, F(1,28) = 45.69, p < .0001 for the collision times and F(1,28) = 5.11, p = .03 for the attempt times. The results suggested that the reduction of the time cost, collision times and attempt times was significant for the mobile manipulation task through using the proposed teleoperation method.

From these results, it is seen that, with the QoT adaptive teleoperation method, the time cost, collision times and attempt times were all reduced for the three different tasks. This was credited to the online QoT based adaption in teleoperation. When the QoT was high, which indicated that the teleoperator could handle most tasks well, the velocity and responsivity of the robot were then changed to a high mode by higher gains. Therefore, the teleoperator could complete the tasks in a shorter time with less collision and attempt times. When the QoT was low, which indicated that the teleoperator might not be able to handle some tasks

Time	Method		Tes	st 1		Test 2				Test 3			
		А	В	С	D	А	В	С	D	А	В	С	D
0 10 AM	Trad.	1	2	3	2	0	0	1	1	0	1	3	2
<i>J</i> -10 AM	Prop.	0	1	1	2	0	0	0	1	0	0	1	1
2 3 PM	Trad.	1	1	3	1	0	1	1	1	0	1	0	1
2-3 1 1/1	Prop.	0	0	1	1	0	1	1	2	0	0	0	1
10 11 PM	Trad.	0	1	2	1	0	1	3	2	1	2	3	3
10-11 1 11	Prop.	0	0	0	1	0	0	1	2	0	0	0	1
		Test 4							1				
Time	Method		Tes	st 4	1		Tes	st 5			Avg.	times	3
Time	Method	A	Tes B	st 4	D	A	Tes B	st 5 C	D	A	Avg. B	times C	, D
Time	Method Trad.	A 1	Tes B	st 4 C 2	D 1	A 0	Tes B 1	st 5 C 2	D 1	A 0.4	Avg. B 1.0	times C 2.2	D 1.4
Time 9-10 AM	Method Trad. Prop.	A 1 0	Tes B 1 0	st 4 C 2 0	D 1 1	A 0 0	Tes B 1 0	st 5 C 2 0	D 1 1	A 0.4 0.2	Avg. B 1.0 0.2	times C 2.2 0.4	D 1.4 1.2
Time 9-10 AM	Method Trad. Prop. Trad.	A 1 0 1	Tes B 1 0 2	st 4 C 2 0 3	D 1 1 2	A 0 0 0	Tes B 1 0 1	st 5 C 2 0 3	D 1 1 3	A 0.4 0.2 0.4	Avg. B 1.0 0.2 1.2	times C 2.2 0.4 2.0	D 1.4 1.2 1.6
Time           9-10 AM           2-3 PM	Method Trad. Prop. Trad. Prop.	A 1 0 1 0	Tes B 1 0 2 0	st 4 C 2 0 3 0	D 1 1 2 1	A 0 0 0 0	Tes B 1 0 1 0	st 5 C 2 0 3 1	D 1 1 3 1	A 0.4 0.2 0.4 0.0	Avg. B 1.0 0.2 1.2 0.2	times C 2.2 0.4 2.0 0.6	D 1.4 1.2 1.6 1.2
Time 9-10 AM 2-3 PM	Method Trad. Prop. Trad. Prop. Trad.	A 1 0 1 0 0	Tes B 1 0 2 0 2	st 4 C 2 0 3 0 2	D 1 1 2 1 4	A 0 0 0 0 1	Tes B 1 0 1 0 1	st 5 C 2 0 3 1 3	D 1 1 3 1 2	A 0.4 0.2 0.4 0.0 0.4	Avg. B 1.0 0.2 1.2 0.2 1.4	times C 2.2 0.4 2.0 0.6 2.6	D 1.4 1.2 1.6 1.2 3.0

Table 4.6 Comparison of collisions and attempt times in mobile manipulation task

A: collision times of mobile base with the obstacle in navigation portion;

B: collision times of gripper with the bottle in pick-up portion;

C: collision times of the bottle with the cylinder in placing portion;

D: attempt times of placing the bottle into the cylinder.

well, especially for some complicated tasks, the velocity and responsivity of the robot were then adaptively changed to a properly low mode by appropriate gains determined by the QoT and TDI. Therefore, the teleoperator could still complete the tasks with a relatively shorter time and less collision and attempt times compared to the traditional method. With this adaption design, the teleoperators could feel that the operations of the robot were much easier and smoother.

Figure 4.16 shows two robot trajectories of using the traditional method and proposed method to accomplish the bottle manipulation portion in the mobile manipulation task. It can be seen that when using the traditional method there were many back-and-forth motions in the operations. However, the same task became much easier for the teleoperator to accomplish when the QoT based method was used. The trajectory was much shorter and smoother.

The experimental results of manipulation, navigation, and mobile manipulation demonstrated that the efficiency and safety of the telerobotic operations could be significantly improved by using the QoT adaptive control in the teleoperation. This proposed teleoperation method could make the telerobotic operations easier, safer and more efficient.



Figure 4.16 Trajectory comparison: (left) traditional method. (right) proposed method

#### 4.6 Chapter Summary

This chapter introduce the teleoperation of mobile manipulators adaptive to quality of teleoperator (QoT). A new concept named Quality of teleoperator (QoT) is proposed to represent operation status of the teleoperator. The QoT is obtained based on five QoT indicators generated based on the teleoperator's brain EEG signals. The experimental results

illustrated that the QoT could represent the operation status of the teleoperator in various situations. A QoT adaptive control method is designed to adjust the velocity and responsivity of the robotic system such that the efficiency and safety of the telerobotic operations can be enhanced. The proposed method was experimentally implemented on a mobile manipulator teleoperation system. Compared to the traditional teleoperation method, the QoT based method could facilitate the teleoperator to accomplish teleoperation tasks with easier, safer and more efficient operations.

### Chapter 5

# Teleoperation of Mobile Manipulators by Natural Language Control

#### 5.1 Introduction

This chapter introduces the design of a mobile manipulator teleoperation system by natural language control with the capability of handling such unexpected events in the environment and robotic system. Controlling robots by natural language is more intuitive and efficient than using the conventional joysticks in many applications such as the teleoperation. Most existing methods on natural language control have been focused on translating the natural language commands to the control commands which are executable by the robots. However, the exceptions in the system such as unexpected events in the environment and robotic system are also very important issues. Therefore, this chapter proposes a new natural language control framework for handling these unexpected events. The proposed method integrates the environmental and robot feedback into the natural language processing and the robot planning and control such that these unexpected events could be automatically handled [118]. The designed method was implemented and tested on a natural language teleoperated mobile manipulator system and the experimental results demonstrated the effectiveness and advantages of the method.

#### 5.2 Natural Language Controlled Robotic System

The typical schema of a natural language controlled robotic system is shown in Figure 5.1. The human user gives the robot tasks by natural language. Through the natural language processing, it is converted to a formal language which could be understood by the robot. Then, a planner is applied to generate the low-level continuous robot trajectories for achieving the given task. In the planning process, a sequence of actions are usually first generated for accomplishing the task and the trajectories are then generated for achieving these actions. Finally, the robot feedback controller is applied to control the robot to track the generated trajectories.



Figure 5.1 Schema of natural language controlled robotic system

In the design, the natural language processing and the robot planner are executed in an open-loop manner. Once they have generated the outputs, they sent the outputs to the lower processes directly and will not check if these outputs will be correct or not in future. Such a design will encounter problems when some unexpected events happen in the environment and robotic system because these events may cause the previously generated outputs no longer correct for achieving the given task.

To address this issue, similar to feedback control process, the feedback information could be sent to the natural language processing and the robot planner for handling these unexpected events in the environment and robotic system. In general, the feedback includes the static and dynamic information of the robotic system such as robot position, moving direction, etc., and the environment such as object position, color, shape, etc. It is like the human's perception and thus named *perceptive feedback*. It can be obtained from the sensory information including both robot sensors and environmental sensors. The feedback information can be used to for the grounding process in natural language processing for handling unexpected events in the environment. The feedback information can also be used for the perceptive planning and control of the robotic system for handling unexpected events in the robotic system.

#### 5.3 Natural Language Processing

#### 5.3.1 Semantic Processing

For the semantic processing in this paper, a partial parser based on combinatory categorial grammar (CCG) [119] is applied to extract semantic information from human natural language utterances. We have defined a set of basic CCG lexicon rules, which covers key lexicons in our domain, such as actions, object colors, shapes, spatial relations and so forth. Given a human utterance, our CCG parser repeatedly searches for the longest sequence covered by the grammar until the end of the utterance. For example, given the utterance "Pick up the small brown block", our parser will generate a semantic representation as follows:"Action:Pickup; Argument:x(size(x,small),color(x,brown), object-type(x, block))".

#### 5.3.2 Graph Matching Process

The output of semantic the processing describes the task from the perspective of the human, which has different representations and references from the robot. For example, the robot can only recognize some sensory features such as the RGB values and coordinates which are totally different from the symbolic information in the semantic processing output. To make this output understandable by the robot, the common grounding between the human and robot is required. Extensive studies have been conducted on the human-robot common grounding [120][121][122]. However, due to their complicated grounding processes and design, it is difficult and costly to apply them for online handling of unexpected events. Thus, we use a simple graph-matching based grounding method. Previously, we have used the similar idea to design an inexact graph-matching based grounding method for mediating the shared perceptual basis in the human-robot dialogue [123]. In this paper, we redesign the graph-matching method for online handling the unexpected events in the natural language control.

The goal of the grounding in this paper is to convert the objects described by the human in natural language to the physical objects detected by the robot in the environment. The basic idea of the graph-based common grounding is to build the human natural language task and the robot perception as two graphs and consider the common grounding as a graphmatching process, as shown in Figure 5.2. The former graph is named as task graph and the latter is named as perceptive graph.

The task graph from the perspective of the human is represented by

$$G_H(V, E) \tag{5.1}$$



Figure 5.2 Graph matching process

where  $V = \{v_i, i \in \{1, 2, ..., |V|\}\}$  is the set of vertices and  $E = \{e_{ij}, i \neq j \ i, j \in \{1, 2, ..., |V|\}\}$  is the set of edges.

The vertices represent the robot and objects. If the vertex is an object, it also includes some symbolic attributes such as the name (e.g.: block, ball) and color (e.g.: red, blue). These attributes are denoted by  $a_{v,k}$ , k = 1, 2, ... The edges represent the relationship between two vertices such as the spatial relationship (e.g.: left, right) and size comparison relationship (e.g.: bigger, smaller). Similar to the vertices, for the edges, they may also include multiple symbolic attributes which are denoted by  $a_{e,k}$ , k = 1, 2, ...

The task graph is constructed based on the output of the semantic processing. The output of the semantic processing is usually in the form of some formal semantic representations, e.g., first-order logic representations. The vertices of the graph come from the entities in the formal semantics. The one-arity predicates in the semantics correspond to the vertex attributes and the two-arity predicates correspond to the edge attributes. This paper assumes that the natural language commands have already incorporated enough information for constructing the task graph. Sometimes, to obtain an informative task graph, the dialog between the human and robot can be used.

The perceptive graph from the perspective of the robot is represented by  $G_R(V, E)$ . The vertices V represent the robot and objects in the environment detected by the robot sensors. Similar to the task graph, an object vertex may include multiple attributes such as the name, color and position. The names are represented by some symbolic values such as O1, O2, etc., the colors are represented by RGB values and the positions are represented by coordinates. The edges E represent the relationship between two vertices. Each edge may also include multiple edge attributes which are defined based on the vertex attributes.

To ground the objects between the human and robot, the goal is to find objects in the robot perception to match the objects mentioned by the human. Based on the graph representations, this goal could be transferred to finding vertices in the perceptive graph to match the vertices in the task graph, as shown in Fig. 2. It could be further transferred to finding a subgraph  $G'_R(V, E)$  in the perceptive graph which has the maximal matching value with the task graph  $G_H(V, E)$ . It could be expressed by:

$$\max_{G'_R \subset G_R} \Phi\left(G'_R, \, G_H\right) \tag{5.2}$$

where  $\Phi(G'_R, G_H)$  is the matching value between the two graphs  $G'_R$  and  $G_H$ .

The difficulty in computing the matching value is the different representations of the two graphs. The attributes in the task graph are represented by symbolic values but the attributes in the perceptive graph are represented by numeric values. To address this difficulty, some symbol grounding functions are defined to bridge these two representations. The output of the function is a value in [0, 1]. The larger the output is, the closer the numeric value matches with the symbolic value.

For the vertex attribute matching, a set of symbol matching functions are defined to compute the matching value between a symbolic representation and a numeric representation. They are denoted by  $\phi_{a_{v,k}}(m_{v,k})$ , where  $m_{v,k}$  is the numeric value corresponding to the symbolic value  $a_{v,k}$ .

Taking the red color for an example, the symbolic representation is red and the numeric representation is RGB values. The symbol matching function is defined by

$$\phi_{red}(m_{rgb}) = \begin{cases} 1 - \|m_{rgb} - m_{red}\|/l_{red}, \\ & if \ \|m_{rgb} - m_{red}\| < l_{red} \\ & 0, \quad otherwise \end{cases}$$
(5.3)

where  $m_{red}$  is the RGB values which best describe the red color and  $l_{red}$  is a limit to determine the RGB boundary for the red color.

For the edge attribute matching, a set of symbol matching functions are also defined. They are denoted by  $\phi_{a_{e,k}}(m_{e,k}, n_{e,k})$ , where  $m_{e,k}$  and  $n_{e,k}$  are the numeric values on the two vertices of the edge which are related to the symbolic value  $a_{e,k}$ .

Taking the spatial relationship for an example, the symbolic representation is "left" and the numeric representation contains two coordinates of the related vertices:  $m = [x_m, y_m, z_m]^T$  and  $n = [x_n, y_n, z_n]^T$  where the positive y axis is the left direction. The symbol matching function is defined by

$$\phi_{left}(m,n) = \begin{cases} |y_m - y_n| / |y_{left} - y_n|, & if \ y_m > y_n \\ 0, & otherwise \end{cases}$$
(5.4)

where  $y_{left}$  is the y axis coordinate of the leftmost object in the environment. Notice that

if  $y_{left} - y_n = 0$  we must have  $y_m \le y_n$  and then  $\phi_{left}(m, n) = 0$ .

Based the definition of the symbol grounding functions, the matching value between the two graphs  $G'_R$  and  $G_H$  can be computed by

$$\Phi\left(G'_{R}, G_{H}\right) = \frac{1}{N} \left(\sum_{a_{v,k} \subset A} \phi_{a_{v,k}}\left(m_{v,k}\right) + \sum_{a_{e,k} \subset E} \phi_{a_{e,k}}\left(m_{e,k}, n_{e,k}\right) - \sum_{a_{v,k} \subset \bar{A}} p_{a_{v,k}} - \sum_{a_{e,k} \subset \bar{E}} p_{a_{v,k}}\right)$$

$$(5.5)$$

where N is the total number of the vertex and edge attributes in  $G_H$ . A is the set of matched vertex attributes between  $G'_R$  and  $G_H$ , E is the set of matched edge attributes between  $G'_R$ and  $G_H$ ,  $\bar{A}$  and  $\bar{E}$  are the sets of unmatched vertex attributes and unmatched edge attributes in  $G_H$ , and  $p_{a_{v,k}}$  and  $p_{a_{e,k}}$  are the penalties for the unmatched vertex and edge attributes. In this paper, the penalties are all assigned as a small constant 1/N.

After solving the matching value computation problem, the next problem is how to find the optimal subgraph in the perceptive graph which can match the best with the task graph. The problem can be commonly solved by the  $A^*$  search algorithm [124]. In this paper, to save the memory cost, we employ a simple beam search algorithm to retain the tractability [125]. The beam size is set as  $hJ^2$ , where h is the current level of the searching tree and Jis the size of the bigger graph.

#### 5.3.3 Graph-Matching based Grounding

In the graph-matching based grounding process, the optimal matched graph is adopted only when the matching value is bigger than a threshold  $\Phi_{Th}$ . Otherwise, it will be considered that there is no appropriate matching. If the appropriate matching could be found, the grounding of the objects is then achieved. The objects in the semantic processing output can be replaced by the object names described in the perceptive graph to form the formal language which could be understood by the robot.

In this formulation of natural language processing, as shown in Fig. 3, the task graph can be considered as the control objective, the perceptive graph can be considered as the control feedback, the graph matching value can be considered as the control error, and the graph searching algorithm can be considered as the controller. Therefore, the entire process can be considered as a closed-loop control process and is capable of handling some system unexpected events.

Notice that in order to save the system resources the grounding process is unnecessary to be always run in real time. This paper uses this process to handle unexpected events in the environment, e.g., objects are added to or removed from the environment. Thus, only when the environment changes are detected, the perceptive graph will be updated and the graph searching algorithm will be rerun. If the maximal matching value is larger than  $\Phi_{Th}$ , a new formal language output of the natural language processing will be generated. Otherwise, the robot will be stopped and wait for new formal language commands. If the maximal value is smaller than  $\Phi_{Th}$ , a Not found notification will be generated. If there are multiple optimal matching which have competitively maximal matching values, i.e., the difference among these maximal matching values is smaller than a very small threshold, an Ambiguity notification will be generated.



Figure 5.3 Graph-matching based grounding process

## 5.4 Planning and Control for Natural Language Controlled Robotic Systems

#### 5.4.1 Perceptive Abstracted Modeling

Based on the perceptive feedback, the entire system can be abstracted by a finite-state automaton using some predefined rules. In order to obtain a simple model for possible online implementations, only the states and actions (transitions) necessary for the tasks are considered to construct the automaton. The primitive actions are modelled as automata and then connected through the mutual states to construct the final automaton. Each state contains not only a state symbol representing the discrete state but also some variables representing the continuous states such as the positions. Since the model can be updated based on the perceptive feedback, it is named perceptive system model and described by a 5-tuple automaton

$$\mathbf{A} = \left(Q(P), \ \Sigma, \ \delta, \ q_0(p_{0,\delta}), \ Q_m(p_{m,\delta})\right) \tag{5.6}$$

where

- Q(P) is the set of states determined by the perceptive feedback (Q is the set of state symbols and P is the set of variables). A single state i is represented by q<sub>i</sub>(p<sub>i,δ</sub>) and p<sub>i,δ</sub> contains the continuous variables such as positions to represent the initial or final states of all actions related to the state i.
- $\sum$  is the set of actions.
- $\delta: Q(P) \times \sum \rightarrow Q(P)$  is the state transition of actions.
- $q_0(p_{0,\delta}) \in Q(p)$  is the initial state determined by the perceptive feedback.
- Q<sub>m</sub>(p<sub>m,δ</sub>) ∈ Q(P) is the set of marked states representing the completion of a task, which is determined by the task specifications and perceptive feedback.

**Example 1:** Consider that there are three objects and the task is to control a robot to pick up one object. Figure 5.4 shows the model to describe the states and transitions of interest for the task. The states are represented by  $d_1d_2(p_{d_1d_2,\delta})$ . The first digit  $d_1 \in \{0, 1, 2, 3\}$  means that the end-effector is at the position of object  $d_1$  if  $d_1 \in \{1, 2, 3\}$  and at the initial position if  $d_1 = 0$ . The second digit  $d_2 \in \{0, 1\}$  means that the gripper is open if  $d_2 = 0$  and closed if  $d_2 = 1$ . The actions  $\delta$  are represented by symbols, where  $m_{ij}$  means to move the end-effector from position *i* to position *j* and *o* and *c* means to open and close the gripper respectively.  $p_{d_1d_2,\delta}$  represents the positions of the end-effector or the gripper fingers for actions  $m_{ij}$ , *o* or *c* related to the state  $d_1d_2$ .



Figure 5.4 Example of perceptive system model

#### 5.4.2 Planning of Robotic Systems

To generate the trajectories for achieving the task described by the formal language, it needs two sequential processes. The first process is the action scheduling which is to generate a sequence of actions for achieving the given task and the second process is the motion planning which is to generate trajectories for achieving the action sequence.

The action scheduling process could commonly be solved by some synthesis algorithms in the discrete event systems such as the supervisory control synthesis algorithm [126][127] and linear temporal logic synthesis algorithm [92]. In this paper, for online action scheduling, a simple automata-based method is adopted. The perceptive automata model is first built to abstract the robot and the environment based on the perceptive feedback. The given task described by the formal language is then converted to a marked state in the model based on the template rules. Then, the problem becomes to find the shortest path from the current state to the marked state in the graph-like automata model. The searching techniques in graph theory, such as Dijkstra's algorithm, could be applied to find the solution. The action sequence is then obtained from the actions along the shortest path.

The motion planning process is to generate the trajectories to achieve the actions along the action sequence, i.e., generate a trajectory for moving from the start position to the finish position for each action. Such a problem has been well studied and a number of algorithms can be adopted to solve it. In an obstacle-free environment, the motion planning problem can be easily solved by using the minimum-time based method or minimum-energy based method [128].

In traditional planning methods, the two steps above have finished the planning process and the generated trajectories are parameterized by the time t. The desired trajectories for achieving the action sequence are expressed by

$$Y_{\delta 1}^{d}(t): t \in (t_{0}, t_{1}), \quad Y_{\delta 2}^{d}(t): t \in (t_{1}, t_{2}),$$
  
$$\cdots, \quad Y_{\delta n}^{d}(t): t \in (t_{n-1}, t_{n})$$
(5.7)

where  $\delta i$ , i = 1, 2, ..., n are the actions along the action sequence. These trajectories are then sent to the low-level controller for tracking.

#### 5.4.3 Perceptive Planning and Control of Robotic Systems

In the traditional time-based planning and control, the instantaneous desired input for the control system is computed by plugging time t into the desired trajectories which are parameterized by t. The control system is then driven by time and the motion planning is a completely open-loop process. This may cause problems when unexpected events happen on the robotic system. For example, if the robot is stopped by unexpected system halt or obstacles, the time-based desired trajectories will still keep evolving since the time t never stops. Then, the system may become unstable and unsafe. Conventionally, to cope with such unexpected events, some safety mechanism has to be designed in advance to completely stop the system. After the unexpected events are resolved, the motion planning process needs to be performed again for regenerating the new trajectories.

To solve this problem more efficiently, a perceptive motion planning and control method is designed. The methods based on the similar idea have been successfully applied to continuous motion tracking [99] and multi-robot formation tracking [97] in our previous work. It is redesigned and extended to the tracking of a sequence of actions in this paper.

The basic idea of the method is to make the control system driven by the physical output of the robotic system instead of the time. At the beginning, the desired trajectories for the actions are parameterized by a non-time reference defined as perceptive reference s. sis independent of time and directly relates to the physical output of the robotic system. For example, for moving the robot from one position to another position, s could be the distance traveled by the robot along the desired trajectory. For another example, for closing or opening the gripper, s could be the distance traveled by the gripper fingers. The new desired trajectories for achieving the action sequence are then expressed by

$$Y_{\delta 1}^{d}(s): s \in (s_{\delta 1}^{0}, s_{\delta 1}^{f}), \quad Y_{\delta 2}^{d}(s): s \in (s_{\delta 2}^{0}, s_{\delta 2}^{f}),$$
  
$$\cdots, \quad Y_{\delta n}^{d}(s): s \in (s_{\delta n}^{0}, s_{\delta n}^{f})$$
(5.8)

where  $s_{\delta i}^0$  and  $s_{\delta i}^f$  are the perceptive reference variables corresponding to the start and finish positions of the action  $\delta i$ .



Figure 5.5 Perceptive planning and control

The perceptive planning and control process can be seen in Figure 5.5. The instantaneous desired input for the control system is computed by plugging the optimal perceptive reference  $s^*$  into the *s*-parameterized trajectories, where  $s^*$  is determined by the physical output of the robotic system. The computation of  $s^*$  is expressed by

$$s^{*} = s', \quad if \ s' \in \left[s^{0}_{\delta i}, s^{f}_{\delta i}\right)$$

$$\begin{cases} s^{*} = s^{0}_{\delta(i+1)} & if \ s' = s^{f}_{\delta i} \ and \ i < n \\ i = i+1 & (5.9) \end{cases}$$
where
$$s' = \underset{s \in \left[s^{0}_{\delta i}, s^{f}_{\delta i}\right]}{arg \min} \left\{ \left\| Y^{d}_{\delta i}(s) - Y_{\delta i} \right\| \right\}$$

where s' is computed by minimizing the tracking error while the robot is executing an action  $\delta i$  and  $Y_{\delta i}$  is the current physical output of the robot corresponding to the action  $\delta i$ . The computation of s' can be also considered an orthogonal projection from  $Y_{\delta i}$  to the desired trajectory  $Y_{\delta i}^d(s)$ . If the robot has accomplished an action  $\delta i$ , which could be indicated by  $s' = s_{\delta i}^f$ , we have  $s^* = s_{\delta (i+1)}^0$  and i = i + 1 to switch the perceptive reference for driving the next action  $\delta(i+1)$ .

The perceptive planning and control design changes the motion planning process to be an online and closed-loop process. The control system is driven by the physical output of the robotic system instead of time t. The desired system input is then online determined by the system output such that the previously mentioned unexpected events on the robotics system can be automatically handled. If the robot is stopped by unexpected events, the computed perceptive reference  $s^*$  stops evolving as well. The following trajectories and actions are then suspended until the unexpected events are resolved. Once the events are resolved, the robot starts to move and  $s^*$  continues to evolve to drive the system. The unexpected event handling process is completely automatic and no replanning process is required.

#### Remarks:

- The high-level planning are no longer open-loop processes. Based on the perceptive feedback, they become closed-loop processes and the system is therefore able to handle some unexpected events.
- 2. The formalism can be extended to natural language control of multiple concurrent robotic systems. The only concern is to design a perceptive reference which can carry perceptive feedback from all concurrent robotic systems. Some ideas of designing the perceptive reference for such systems can be found in [97][129].

#### 5.5 Stability Analysis

**Theorem 1:** If the perceptive reference s is a monotone increasing function of time t and the original time-driven robot control system is asymptotically stable, then the closed-loop perceptive planning and control system is (asymptotically) stable under (finite) unexpected stops of the robot.

*Proof:* If the original time-driven robot control system is asymptotically stable, according to the converse theorem [130], we can find a Liapunov function V(x(t)) such that V(x(t)) > 0

and V(x(t)) < 0. After introducing the perceptive reference s, we have

$$\frac{dV(x(t))}{dt} = \frac{dV(x(s))}{ds}\frac{ds}{dt} < 0$$
(5.10)

If the perceptive reference s is a monotone increasing function of time t, then we have ds/dt > 0. Notice that this is always true no matter the robot is stopped or not.

During the perceptive planning and control process, if the robot is not stopped, then s keeps evolving, which makes ds/dt > 0. Therefore, we always have  $\dot{V}(x(t)) < 0$ . So the system is asymptotically stable.

If the robot is unexpected stopped, this stop also results in the stop of the perceptive reference s. So we have  $\dot{V}(x(t)) = (dV(x(s))/ds) (ds/dt) = 0$  and the system is still stable. Furthermore, if the number of stops is finite, then we have  $\dot{V}(x(t)) \leq 0$  and the time for  $\dot{V}(x(t)) = 0$  is finite. So the system is asymptotically stable.

**Definition:** The action scheduling is stable if the formal language task could be always achieved through the actions of the robot.

Theorem 2: If the graph corresponding to the system automata model is always connected, the closed-loop action scheduling is stable under unexpected moves of the objects.

*Proof:* If the graph is connected, then given two nodes in the graph there exists at least one path to move from one node to another. Correspondingly, we can always find a sequence of actions in the system automata model to move from the current state to the formal language state under any unexpected moves of objects. This implies that we can always find a sequence of robot actions to achieve the formal language task. So the action scheduling is stable.

We can properly design the primitive robot actions to make sure that the states in the

system automata model could be always connected through these actions. Through this, the stability of the action scheduling could be always ensured.

**Definition:** The natural language processing is stable if the output formal language could correctly represent the natural language task.

We can prove that the designed closed-loop natural language processing is stable under unexpected events in the environment. The correctness of the output formal language could be represented by the following two conditions

(1) The subgraph corresponding to the output formal language,  $G_R^{FL} \subset G_R$ , is the unique subgarph in  $G_R$  to have the largest matching value  $\Phi(G_H, G_R^{FL})$  with the task graph  $G_H$ 

(2) Define the correctness threshold as  $M, \ 0 < M \leq 1$  and the matching value satisfies  $\Phi(G_H, G_R^{FL}) \geq M.$ 

In the designed natural language processing, if unexpected events occur, the perceptive graph will be updated and the natural language processing will generate output formal language only if we can find a subgraph in  $G_R$  which can satisfy the conditions (1) and (2). Otherwise, the notifications to the human will be generated and no formal language will be outputted. This ensures that the output formal language could always correctly represent the natural language task. Therefore, the designed natural language processing is stable.

#### 5.6 Experimental Results and Analysis

#### 5.6.1 Experimental Implementaiton

The designed system was implemented on a natural language controlled mobile manipulator system. The mobile manipulator consisted of a 4-wheel Segway RMP 400 mobile platform and a 7-DOF Schunk LWA3 manipulator, as shown in Figure 5.6.



Figure 5.6 Experimental setup

A wireless microphone was used to capture the human's speech and then sent it to a laptop computer via the Bluetooth. The speech was recognized as natural language text by the Dragon Speech Recognition software and further processed by the natural language processing to transfer the natural language representation to the formal language representation. The formal-language-described task was then sent to the mobile manipulator's onboard computer and the robot planning and control were performed to execute the task. The natural language processing computer and the robot computer communicated with each other via the Internet.

At the same time, the robot and environmental information were sensed by the robots onboard sensors. The robot information including the end-effector position and the griper fingers position were calculated based on the encoder readings and the environmental information including the coordinates and RGB values of the objects in the environment were obtained through a calibrated 3D vision system as shown in Figure 5.7. These information were then shared to the natural language processing and the robot planning and control

#### processes.



Figure 5.7 Environmental information by the vision system

#### 5.6.2 Experimental Results on Natural Language Control

The experiments were to control the mobile manipulator to manipulate blocks by natural language. The tasks for the first set of experiments were to pick up objects in the environment. The experimental results showed that all these tasks could be successfully accomplished without exceptions. For the setup shown in Figure 5.8, two natural language tasks were tested.

The first task was "Pick up the left green block". Through the natural language processing, the formal language "Pickup (O1)" for picking up the leftmost blue object was generated with the maximal matching value. The matching values for picking up O1, O2, O3 and O4 were 0.85, 0.17, 0.00 and 0.35 respectively. Notice that the "left" was considered to be relative to the robot's end-effector position in the natural language processing. Through the perceptive planning and control, the action sequence of  $m_{O1}$ , c, l and their corresponding trajectories were correctly generated and executed by the mobile manipulator controller, where the action sequence stood for "Move to O1", "Close gripper" and "Lift".

The second task was "Pick up the green block on right of red block". Through the natural language processing, the correct formal language "Pickup (O4)" with the maximal matching value of 0.84 was generated. Through the perceptive planning and control, the trajectories were correctly generated and executed to accomplish the given task.



Figure 5.8 Experimental setup for normal natural language control

In Figure 5.9, the speech commands from the human user included "Close gripper", "Open the gripper", "Move to the red block", and "Pick up the small brown block". The corresponding robot motions are presented. It can be seen that using the designed natural language control scheme, the robot could successfully accomplish all tasks described by natural language when there are no uncertain and unexpected events. These experiments demonstrated the effectiveness of the designed natural language control method under no exceptions.



Figure 5.9 Results of natural language control: no uncertain and unexpected events

#### 5.6.3 Experimental Results on Handling Unexpected Events

The experiments of the designed natural language method were conducted under unexpected events in the environment and also the robotic system. The experimental results are presented below.

The tasks for the first three experiments were all "Pick up the left green block". In the first experiment, at the beginning, the human could see all objects but the leftmost block was sheltered by a board in the robots vision system, as shown in Figure 5.10, so the generated formal language was "Pickup (O1)". During the execution, the board was removed and robot could see the object. The grounding process was rerun and a new form language of "Pickup (O5)" was generated because "Pickup (05)" had a larger matching value of 0.85 than "Pickup (04)" whose matching value was 0.67. Then, the robot actions and corresponding trajectories were generated and executed for accomplish the new formal language task for achieving the original task of the human.

In the second experiment, the unexpected event was to remove the left green block during the execution as shown in Figure 5.11. The perceptive graph was then updated and the grounding process was rerun. The matching values for the remained three objects were



Figure 5.10 Experiment for unexpected environmental change

0.45 0.30 and 0.00, which were all smaller than the threshold  $\Phi_{Th} = 0.5$ . So the robot was stopped and a notification "Object not found!" was generated.



Figure 5.11 Experiment for unexpected object removal

In the third experiment, a new green block was put at the similar location as the previous leftmost block as shown in Figure 5.12. The grounding process was rerun and generated two formal languages of "Pickup (O6)" and "Pickup (O1)" which had competitively large matching values of 0.84 and 0.82. Thus, the robot was stopped and an ambiguity notification "Two objects were found!" was generated.

Figure 5.13 and Figure 5.14 show the results of handling other unexpected events in the environment. The human user asked the robot to pick up the big brown block and big red block by saying "Take that big brown block" and "Grasp that red block" respectively. While the robot was moving to them, the big brown block was moved to other positions twice at



Figure 5.12 Experiment for unexpected ambiguity

11 s and 22 s in Figure 5.13 and the red block was swapped with the big brown block at 4 s in Figure 5.14. The perceptive feedback was then sent back to the update the perceptive system model and new plans were also generated to achieve the original tasks.

Notice that although the designed tasks are simple, these results can clearly demonstrate the advantages of the designed method for handling unexpected events in the robotic system and the environment.



Figure 5.13 Results of natural language control: the robot is moved



Figure 5.14 Results of natural language control: the object is swapped

In the final experiment, the robot was unexpectedly stopped during executing the trajectories for achieving the task of picking up a red block. Figure 5.15 shows the process of handling these unexpected events in the robotic system. Figure 5.16 shows a portion of the trajectories of the end-effector and gripper. When the robot was moving to the red block, it was blocked by the human by hand at about 7 s and 11s respectively. As a result, the calculated perceptive reference  $s^*$  stopped evolving as well, which caused the automatic stop of the involvement of robot trajectories. The following actions and trajectories were then suspended. When the hand was removed at about 9 s and 13 s respectively, the robot could move again and  $s^*$  started to evolve. The suspended actions and trajectories were then recovered and driven by  $s^*$  again. The entire handling process was completely automatic and no replanning of the system was required.



Figure 5.15 Results of natural language control: the robot is blocked



Figure 5.16 Results of handling unexpected robot stops
Notice that it does not mean that the natural language method in the current design could best handle all unexpected events. Taking the unexpected event in Figure 5.10 for example, we assumed that the human could always see all objects, but it was also possible that some objects like O5 was sheltered to the human too at the beginning. So an ambiguity notification should be the better response in this situation. Thus, to better handle these unexpected events with uncertainties, a dialog between the human and the robot will be necessary.

In the dialog, the proposed natural language control method could actually provide important hints to formulate the questions from the robot to the human, e.g., "Which left green block, leftmost or the other?". So the proposed natural language control method could be an important step towards well handling unexpected events and also uncertainties in the natural language control system.

#### 5.7 Chapter Summary

This chapter introduced a method for teleoperation of mobile manipulators through natural language commands. The natural language control method for robotic operations using the perceptive feedback is proposed to make the system be able to handle some unexpected events in both the environment and robotic system. A graph-matching based grounding method is designed in the natural language processing for handling unexpected events in the environment and a perceptive robot planning and control method is designed for handling unexpected events in the robotic system. The experimental results on a natural language controlled mobile manipulator system demonstrated the effectiveness and advantages of the designed methods.

## Chapter 6

# Development of Mobile Manipulator Systems

#### 6.1 Introduction

The working space of a standard manipulator can be enlarged by introducing a mobile platform such as a holonomic mobile platform or a nonholonomic mobile platform. Through the integration of the advantages of the standard manipulators and mobile platforms, the application areas of mobile manipulators could be significantly expanded.

This chapter introduces the development of two types of mobile manipulators. The first type is to add a holonomic mobile platform to a 6-DOF PUMA manipulator. The second type is to add a nonholonomic mobile platform to a 7-DOF Schunk manipulator. The development of the system hardware and software for the two types of mobile manipulators are introduced in detail. Both mobile manipulators can significantly enlarge the working spaces of the onboard manipulators.

#### 6.2 Development of Holonomic Mobile Manipulators

#### 6.2.1 System hardware Development

The holonomic mobile manipulators are show in Figure 6.1. Each of them consists of a PUMA 560 robotic manipulator and a mobile platform. The mobile platform is a holonomic mobile robot which contains four powered caster wheels. Each powered caster wheel can drive and turn and controlled by two DC motors separately. It has a full three degrees of freedom for plane motion and therefore can move in any directions at any configurations. The mobile platform was completely rebuilt based on the mechanical hardware of the original NOMAD XR4000 robot. The onboard manipulator is a popular PUMA 560 robot. It has six degrees of freedom and has a full motion in the 3D cartesian coordinate system including three position motion and three orientation motion. Multiple onboard sensors are integrated into the mobile manipulator system, including a SICK laser range finder, a SONY camera with a pan-tilt, an array of sonar and infrared sensors, a force/torque sensor and an acceleration sensor. The laser range finder and camera are mounted on the mobile platform. They are mainly used for detecting the environmental information of the robot for navigation. The array of sonar and infrared sensors are integrated into the body of the mobile platform. They are mainly used for detecting the near surrounding information of the robot. The force/torque and acceleration sensors are mounted on the end-effector of the manipulator. They are mainly used for detecting the contact and dynamic information of the gripper.

#### 6.2.2 System Software Development

The holonomic mobile manipulator is controlled by three controllers which run three different operational systems, as shown in Figure 6.2. An industrial PC running the real-time



Figure 6.1 Holonomic mobile manipulators

QNX system is used to control the PUMA 560 manipulator and connect the force/torque/acceleration sensors. A PMAC controller running the real-time embedded PMAC system is used to control the mobile platform. Another industrial PC running the Linux system is used as a host system to coordinate the manipulator and mobile platform. It is also used for connection to the Internet. Besides this, it is used as a controller for all other sensors, including the laser scanner, cameras, sonar and infrared distance sensors. A PHANTOM haptic device and a MS force feedback joystick are used for the operation of the system. Using different controllers, the mobile manipulator can work either autonomously or by teleoperation via the Internet.



Figure 6.2 Implementation of holonomic mobile manipulators

# 6.3 Development of Nonholonomic Mobile Manipulators

#### 6.3.1 System hardware Development

The nonholonomic mobile manipulators are shown in Figure 6.3. They are developed for both indoor and outdoor applications and consist of a 4-wheel mobile platform and a 7-DOF manipulator. The Segway Robotic Mobility Platforms (RMP) are integrated into the two MSU mobile manipulators. One mobile manipulator is equipped with the RMP 400 and another mobile manipulator is equipped with the RMP 400X. The RMP is customized transportation platform that is suitable for moving heavy payloads in tight spaces. It rolls on four rugged ATV tires and can work over a variety of terrain including the concrete road, grassland, snow field, etc. It is powered by lithium-ion battery packs and can work up to 12 hours. It is controlled by an onboard industrial computer through the USB and CAN bus interfaces. It is capable of carrying up to 400 lbs while maintaining impressive tractive ability over the most challenging terrain. The platform provides the mobility function of the mobile manipulator and carries the manipulator and all onboard computers and sensing devices.



Figure 6.3 Nonholonomic mobile manipulators

The 7-DOF Schunk Light Weighted Arms (LWA3) are integrated into the two MSU mobile manipulators. The manipulator is a lightweight and powerful arm which contains seven individual modules and a gripper connected through CAN bus. The gripper contains two fingers equipped with two pieces of tactile sensors and an ATI force/torque sensor is mounted at the end of the gripper. Its payload is up to 5 kg and its repeatability accuracy is up to 0.2 mm. It is powered by the lead-acid battery packs and can work up to 4 hours. It is controlled by an onboard industrial computer through the CAN bus interface. The

application possibilities of the lightweight arm LWA3 are versatile and cover various areas in robotics including manipulation, sensor (such as camera and force) based control, service robots, human-robot interaction, etc.

They are also equipped with two onboard computers and multiple advanced sensors including binocular and omnidirectional vision systems, high-definition cameras, laser range finders, force/torque sensors, tactile sensors, global position systems, inertial measurement units, and pan-tilt units for controlling the movement of the sensors.

#### 6.3.2 System Software Development

The implementation of the nonholonomic mobile manipulator is shown in Figure 6.4. In order to efficiently implement the mobile manipulator system and evenly distribute the computation tasks, two industrial computers are adopted. One computer is used to control the mobile platform and manipulator and acquire real-time data from the force/torque sensor and tactile sensor. Another computer is used to acquire sensory information from other onboard sensors and also control the pan-tilt unit for providing movement of some sensors such as the laser range finder and cameras.

The industrial control computer runs the RTX real-time operation system. It connects to the mobile platform and manipulator though the CAN bus interface. For the control mobile platform, in each control cycle, it reads the four wheel encoders, computes the desired velocities for each wheel based on the control algorithm and the wheel encoder readings, and then sends the velocity commands to the motor controller for real-time execution. For the control of the manipulator, in each control cycle, it reads the angle of each joint from the joint encoders, calculate the joint velocity for each joint using the kinematic controller based on the joint readings, and then send the velocities to the joints for execution. All these



Figure 6.4 Implementation of nonholonomic mobile manipulators

processes are implemented in real time. The control computer can also read the position of the gripper fingers and control the motion of the gripper fingers through the CAN bus interface. For the force/torque sensors, one mobile manipulator acquires the data through the PCI interface using an NI DAQ card and the other mobile manipulator acquires the data through the CAN bus interface. For the tactile sensors, the control computer reads its data through the serial port interface.

The industrial sensor computer runs the Windows XP operation system. It connects to the pan-tilt unit and the onboard sensors through various interfaces. It connects to the pan-tilt unit though the serial port interface. It can read the angles for the pan and tilt and control the motions of the planning and tilting for the mounted sensors such as the laser range finder and the camera. It connects to two high-definition cameras through the USB interface. One is mounted on the gripper for manipulation and the other is mounted on the mobile platform for navigation. It connects to two laser range finders. For the 270 degree laser range finder, it uses the serial port interface, and for the 3600 degree laser range finder, it uses the Ethernet. For the Point Grey binocular and omnidirectional cameras, the sensor computer connects to both them through the IEEE 1394 fireware interface. The sensor computer can also acquire information from other onboard sensors through the serial port interface including the global position system (GPS), inertial measurement unit (IMU), and tire pressure sensors on the four ATV tires. The sensory information are acquired and processed by the sensor computer and then provided to the control computer for potential usage in different applications.

Using this implementation design, the control of the mobile manipulator and data acquisition of the sensors can be efficiently implemented. The computation tasks for the control and data acquisition are evenly distributed to the two industrial computers and the system can thus run in an efficient way.

#### 6.4 Chapter Summary

This chapter introduces two types of mobile manipulators. One is the holonomic mobile manipulator consisted of a 4-caster-wheel holonomic mobile platform and a 6-DOF manipulator and the other one is the nonholonomic mobile manipulator consisted of a 4-wheel nonholonomic mobile platform and a 7-DOF manipulator. The system hardware of the two types of mobile manipulators are introduced. The implementation of system control software are also introduced in detail. The two types of mobile manipulators can both integrate the advantages of the mobile platforms and manipulators and their working spaces are significantly enlarged.

## Chapter 7

# **Conclusions and Future Work**

#### 7.1 Conclusions

The theoretical contributions of the study can be summarized into four aspects.

First, a whole-body control method for the mobile manipulators is developed. An online motion distribution method and an online coordinated control method are designed to solve the whole-body motion control accuracy problem of nonholonomic mobile manipulators under system performance differences and a online sensor-based redundancy resolution scheme is designed to solve the redundancy resolution problem. These methods enable the teleoperator to concentrate on the end-effector task while other tasks such as coordinated control and obstacle avoidance are autonomously handled by the mobile manipulator. Compared to the traditional separate control methods, it can significantly improve the accuracy and efficiency of teleoperation.

Second, a non-time based teleoperation and online coordination method is developed to solve the stability problem of multi-mobile-manipulator teleoperation systems under random communication delays and unexpected events. A non-time perceptive reference is designed as a new reference to replace the time in the system modeling and control. Through this design, the system stability under random communication delays and unexpected events could be ensured. It enables one teleoperator to simultaneously control multiple mobile manipulators to perform multi-robot tasks such as formation and co-transportation. Third, a method is developed to model the human teleoperator into the teleoperation system. A new concept named Quality of teleoperator (QoT) is proposed to represent the operation status of teleoperator. A QoT adaptive control method is designed to adjust the velocity and responsivity of the robotic system according to the QoT. Compared to the traditional teleoperation method, the QoT based method could facilitate the teleoperator to accomplish teleoperation tasks with easier, safer and more efficient operations. Through this design, the teleoperation efficiency and safety are enhanced under various operation status of the teleoperator.

Fourth, a natural language control method is developed for the intuitive and efficient teleoperation of mobile manipulator under unexpected events. A graph-based natural language processing method and a perceptive robot planning and control method are developed to online handle the unexpected events in the environment and robotic system. Compared to the conventional control manner of using joysticks, the new control method makes the communications between the human teleoperator and mobile manipulators more intuitive and efficient.

The application contributions of the study could be summarized into two aspects.

First, two types of mobile manipulators are developed. The first type contains two holonomic mobile manipulators which consist of a 4-caster-wheel holonomic mobile platform and a 6-DOF PUMA manipulator. The second type contains two nonholonomic mobile manipulator which consist of a 4-wheel nonholonomic mobile platform and a 7-DOF Schunk manipulator. The system hardware and software of these mobile manipulators are developed and implemented.

Second, the Internet-based mobile manipulator systems are developed for the two types of mobile manipulators. After implementing the theoretical designs on these systems, stable and efficient mobile manipulator teleoperation systems are developed.

#### 7.2 Future Research Work

#### 7.2.1 Human-Robot Shared Control for Teleoperation

In conventional teleoperation, the human teleoperator is responsible for most of the operations. The cognitive burdens for all decisions are placed on the human teleoperator. Because of some system uncertainties such as the communication constraints, feedback errors and the inexperience of the teleoperator, the actions generated by the teleoperator cannot always be guaranteed to be efficient. To address this problem, the robotic systems can be designed to have proper intelligence and autonomy such that they can recognize the teleoperator intentions through interactions with humans, and then generate decisions based on its own knowledge. Then a human-robot shared control mechanism can be designed to make the human teleoperator and robots work together to accomplish the tasks more efficiently.

Usually the human teleoperator is more intelligent and the mobile manipulators are more efficient and more accurate. Human teleoperator can make intelligent decisions and perform the tasks more advisably based their knowledge and experience. In contrast, the mobile manipulators can perform the tasks more quickly and more accurately. There could be several ways to combine the human's intelligence and robot's efficiency and accuracy.

The first way is to combine them in the lower control level. The human teleoperator can send control commands to the robot and at the same time the robot autonomy could also generate control commands. There could be three ways to use these commands: using the human teleoperator's control commands only, using the robot autonomy's control only and fusing both commands from the human teleoperator and the robot autonomy. These three ways could be online selected according to different task requirements

The second way is to combine them in the higher task level. Since the human teleoperator and the robot autonomy may be good at doing difficult tasks, based on the characteristics and difficulties of different tasks, the tasks could be allocated to either the human teleoperator or the robot autonomy. Sometimes, one task could also be decomposed into multiple subtasks which could be allocated to the human teleoperator and the robot autonomy respectively.

For both combinations in the lower control level and higher task level, the quality of teleoperator (QoT) would be useful. Based on the QoT, it would be possible determine how much the human teleoperator should be involved in the shared control.

Therefore, our future work will focus on designing a human-robot shared control method for the teleoperation system to integrate the intelligence of the human teleoperator and the efficiency and accuracy of the mobile manipulators such that the teleoperation performance could be further enhanced.

#### 7.2.2 Human-Robot Dialog for Teleoperation

In our design of teleoperation of mobile manipulators through natural language, we have proposed a natural language processing method and a planning and control method using the perceptive feedback to handle some unexpected events in the robotic system and the environment. In the future work, we will add human-robot dialog into the natural language control system for efficiently handling system exceptions with uncertainties and ambiguities.

For most existing natural language controlled robotic systems, the high-level processes and the low-level process are connected in an open-loop manner. This formalism usually needs to assume that the low-level processes could absolutely accomplish the outputs of the high-level processes and there are no exceptions in the system. If the assumption is violated, the system will encounter problems.

In the previous natural language control design, we mainly focus on the natural language processing and the planning and control in the robotic system. We assume that the output of the semantic processing could provide enough information to construct the task graph. In reality, to obtain this graph, the dialog between the human and robot may be required.

In addition, to further handle the unexpected events and system uncertainties, the human-robot dialog is also necessary. We have integrated the perceptive feedback into the natural language processing to handle some exceptions in the system. It makes the natural language processing become a closed-loop process by using the perceptive feedback. The task is represented by a goal graph and the perceptive feedback is represented by a perceptive graph. A graph matching and searching algorithm is designed based on these two graphs to generate the natural language processing output to make sure that the task can be correctly achieved with the maximal probability. However, the natural language method in this design could not always best handle all unexpected events due to some uncertainties and ambiguities in the the system. To better handle these unexpected events with uncertainties and ambiguities, a dialog between the human and the robot will be necessary. In the dialog, the proposed natural language processing method could provide important information for formulating the questions from the robot to the human teleoperator.

Therefore, for the future work, we will integrate the human-robot dialog with the designed natural language controlled robotic systems for better handling the unexpected events with uncertainties and ambiguities. The ultimate goal is to make the mobile manipulators be able to communicate with the human teleoprator through natural language like a human.

# REFERENCES

## REFERENCES

- M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. John Wiley & Sons New York, 2006.
- [2] I. Elhajj, N. X. N. Xi, W. K. F. W. K. Fung, Y.-H. L. Y.-H. Liu, Y. Hasegawa, and T. Fukuda, "Supermedia-enhanced Internet-based telerobotics," *Proceedings of* the IEEE, vol. 91, 2003.
- [3] J. Vertut and P. C. Coeffet, "Robot technology; vol. 3a teleoperation and robotics evolution and development," 1986.
- [4] G. Clement, J. Vertut, R. Fournier, B. Espiau, and G. Andre, "An overview of cat control in nuclear services," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, vol. 2, pp. 713–718, IEEE, 1985.
- [5] C. Secchi, S. Stramigioli, and C. Fantuzzi, "Digital passive geometric telemanipulation," in *ICRA*, pp. 3290–3295, 2003.
- [6] S. B. Skaar and C. F. Ruoff, Teleoperation and robotics in space. No. 161, Aiaa, 1994.
- [7] T. Imaida, Y. Yokokohji, M. Oda, T. Yoshikawa, et al., "Ground-space bilateral teleoperation of ets-vii robot arm by direct bilateral coupling under 7-s time delay condition," *Robotics and Automation, IEEE Transactions on*, vol. 20, no. 3, pp. 499–511, 2004.
- [8] J. Funda, R. H. Taylor, B. Eldridge, S. Gomory, and K. G. Gruben, "Constrained cartesian motion control for teleoperated surgical robots," *Robotics and Automation*, *IEEE Transactions on*, vol. 12, no. 3, pp. 453–465, 1996.
- [9] A. J. Madhani, G. Niemeyer, and J. K. Salisbury, "The black falcon: a teleoperated surgical instrument for minimally invasive surgery," in *Intelligent Robots and Systems*, 1998. Proceedings., 1998 IEEE/RSJ International Conference on, vol. 2, pp. 936–944, IEEE, 1998.
- [10] J. Marescaux, J. Leroy, M. Gagner, F. Rubino, D. Mutter, M. Vix, S. E. Butner, and M. K. Smith, "Transatlantic robot-assisted telesurgery," *Nature*, vol. 413, no. 6854, pp. 379–380, 2001.

- [11] N. Diolaiti and C. Melchiorri, "Teleoperation of a mobile robot through haptic feedback," in *Haptic Virtual Environments and Their Applications, IEEE International* Workshop 2002 HAVE, pp. 67–72, IEEE, 2002.
- [12] S.-G. Hong, J.-J. Lee, and S. Kim, "Generating artificial force for feedback control of teleoperated mobile robots," in *Intelligent Robots and Systems*, 1999. IROS'99. Proceedings. 1999 IEEE/RSJ International Conference on, vol. 3, pp. 1721–1726, IEEE, 1999.
- [13] T. Makiishi and H. Noborio, "Sensor-based path-planning of multiple mobile robots to overcome large transmission delays in teleoperation," in Systems, Man, and Cybernetics, 1999. IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on, vol. 4, pp. 656–661, IEEE, 1999.
- [14] N. C. Mitsou, S. V. Velanas, and C. S. Tzafestas, "Visuo-haptic interface for teleoperation of mobile robot exploration tasks," in *Robot and Human Interactive Communication, 2006. ROMAN 2006. The 15th IEEE International Symposium on*, pp. 157–163, IEEE, 2006.
- [15] M. Sitti, B. Aruk, H. Shintani, and H. Hashimoto, "Scaled teleoperation system for nano-scale interaction and manipulation," *Advanced Robotics*, vol. 17, no. 3, pp. 275– 291, 2003.
- [16] N. Durlach, "The potential of teleoperation for entertainment and education," Presence-Teleoperators and Virtual Environments, vol. 6, no. 3, pp. 350–351, 1997.
- [17] S. Westerberg, I. R. Manchester, U. Mettin, P. La Hera, and A. Shiriaev, "Virtual environment teleoperation of a hydraulic forestry crane," in *Robotics and Automation*, 2008. ICRA 2008. IEEE International Conference on, pp. 4049–4054, IEEE, 2008.
- [18] Q. Ha, M. Santos, Q. Nguyen, D. Rye, and H. Durrant-Whyte, "Robotic excavation in construction automation," *Robotics & Automation Magazine*, *IEEE*, vol. 9, no. 1, pp. 20–28, 2002.
- [19] B. Bayle, J.-Y. Fourquet, and M. Renaud, "Manipulability of wheeled mobile manipulators: Application to motion generation," *The International Journal of Robotics Research*, vol. 22, no. 7-8, pp. 565–581, 2003.
- [20] A. Dietrich, T. Wimbock, A. Albu-Schaffer, and G. Hirzinger, "Singularity avoidance for nonholonomic, omnidirectional wheeled mobile platforms with variable footprint," in *Robotics and Automation (ICRA)*, 2011 IEEE International Conference on, pp. 6136–6142, IEEE, 2011.

- [21] S. Soylu, B. J. Buckham, and R. P. Podhorodeski, "Dexterous task-priority based redundancy resolution for underwater-manipulator systems," *CSME Transactions*, vol. 4, no. 31, pp. 519–533, 2007.
- [22] H. Seraji, "A unified approach to motion control of mobile manipulators," The International Journal of Robotics Research, vol. 17, no. 2, pp. 107–118, 1998.
- [23] F. G. Pin, K. A. Morgansen, F. A. Tulloch, C. J. Hacker, and K. B. Gower, "Motion planning for mobile manipulators with a non-holonomic constraint using the fsp (full space parameterization) method," *Journal of robotic systems*, vol. 13, no. 11, pp. 723– 736, 1996.
- [24] J. F. Gardner and S. A. Velinsky, "Kinematics of mobile manipulators and implications for design," *Journal of Robotic Systems*, vol. 17, no. 6, pp. 309–320, 2000.
- [25] O. Egeland, "Task-space tracking with redundant manipulators," Robotics and Automation, IEEE Journal of, vol. 3, no. 5, pp. 471–475, 1987.
- [26] A. Liegeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 7, no. 12, pp. 868–871, 1977.
- [27] A. De Luca and G. Oriolo, "The reduced gradient method for solving redundancy in robot arms," *Robotersysteme*, vol. 7, no. 2, pp. 117–122, 1991.
- [28] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *Robotics and Automation, IEEE Transactions on*, vol. 13, no. 3, pp. 398–410, 1997.
- [29] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *Journal of dynamic systems, measurement, and control*, vol. 108, no. 3, pp. 163–171, 1986.
- [30] J. L. Adams, "An investigation of the effects of the time lag due to long transmission distances upon remote control. phase i. tracking experiments," tech. rep., DTIC Document, 1961.
- [31] T. B. Sheridan and W. R. Ferrell, "Remote manipulative control with transmission delay," *Human Factors in Electronics, IEEE Transactions on*, no. 1, pp. 25–29, 1963.
- [32] R. Anderson and M. W. Spong, "Bilateral control of teleoperators with time delay," Automatic Control, IEEE Transactions on, vol. 34, no. 5, pp. 494–501, 1989.

- [33] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," Automatica, vol. 42, no. 12, pp. 2035–2057, 2006.
- [34] D. Lee and M. W. Spong, "Passive bilateral teleoperation with constant time delay," *Robotics, IEEE Transactions on*, vol. 22, no. 2, pp. 269–281, 2006.
- [35] S. Munir and W. J. Book, "Internet-based teleoperation using wave variables with prediction," *Mechatronics, IEEE/ASME Transactions on*, vol. 7, no. 2, pp. 124–133, 2002.
- [36] S. Munir and W. J. Book, "Wave-based teleoperation with prediction," in American Control Conference, 2001. Proceedings of the 2001, vol. 6, pp. 4605–4611, IEEE, 2001.
- [37] Y. Kamei and C. Ishii, "Passivity based bilateral control with motion scaling for robotic forceps teleoperation system with time delay," in *Control, Automation and Systems*, 2008. ICCAS 2008. International Conference on, pp. 1657–1662, IEEE, 2008.
- [38] A. Bemporad, "Predictive control of teleoperated constrained systems with unbounded communication delays," in *Decision and Control*, 1998. Proceedings of the 37th IEEE Conference on, vol. 2, pp. 2133–2138, IEEE, 1998.
- [39] J. Sheng and M. W. Spong, "Model predictive control for bilateral teleoperation systems with time delays," in *Electrical and Computer Engineering*, 2004. Canadian Conference on, vol. 4, pp. 1877–1880, IEEE, 2004.
- [40] P. Buttolo, P. Braathen, and B. Hannaford, "Sliding control of force reflecting teleoperation: Preliminary studies," *Presence*, vol. 3, no. 2, pp. 158–172, 1994.
- [41] J. H. Park and H. C. Cho, "Sliding mode control of bilateral teleoperation systems with force-reflection on the internet," in *Intelligent Robots and Systems*, 2000.(IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on, vol. 2, pp. 1187– 1192, IEEE, 2000.
- [42] G. M. Leung, B. A. Francis, and J. Apkarian, "Bilateral controller for teleoperators with time delay via μ-synthesis," *Robotics and Automation*, *IEEE Transactions on*, vol. 11, no. 1, pp. 105–116, 1995.
- [43] M. Boukhnifer, A. Ferreira, and J. Fontaine, "Scaled teleoperation controller design for micromanipulation over internet," in *Robotics and Automation*, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on, vol. 5, pp. 4577–4583, IEEE, 2004.

- [44] L. Eusebi and C. Melchiorri, "Force reflecting telemanipulators with time-delay: Stability analysis and control design," *Robotics and Automation*, *IEEE Transactions on*, vol. 14, no. 4, pp. 635–640, 1998.
- [45] D. Taoutaou, S.-I. Niculescu, and K. Gu, "Robust stability of teleoperation schemes subject to constant and time-varying communication delays," in *Advances in Time-Delay Systems*, pp. 327–338, Springer, 2004.
- [46] T. Suzuki, T. Sekine, T. Fujii, H. Asama, and I. Endo, "Cooperative formation among multiple mobile robot teleoperation in inspection task," in *Proceedings of the 39th IEEE International Conference on Decision and Control*, vol. 1, pp. 358–363, Citeseer, 2000.
- [47] O. Reinoso, A. Gil, L. Paya, and M. Julia, "Mechanisms for collaborative teleoperation with a team of cooperative robots," *Industrial Robot: An International Journal*, vol. 35, no. 1, pp. 27–36, 2008.
- [48] D. Lee, O. Martinez-Palafox, and M. W. Spong, "Bilateral teleoperation of multiple cooperative robots over delayed communication networks: Application," in *Robotics* and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on, pp. 366–371, IEEE, 2005.
- [49] R. Wegner and J. Anderson, "Agent-based support for balancing teleoperation and autonomy in urban search and rescue," *International Journal of Robotics and Au*tomation, vol. 21, no. 2, pp. 120–128, 2006.
- [50] I. Farkhatdinov and J.-H. Ryu, "Teleoperation of multi-robot and multi-property systems," in *Industrial Informatics*, 2008. INDIN 2008. 6th IEEE International Conference on, pp. 1453–1458, IEEE, 2008.
- [51] U. Tumerdem and K. Ohnishi, "Multi-robot teleoperation under dynamically changing network topology," in *Industrial Technology*, 2009. ICIT 2009. IEEE International Conference on, pp. 1–6, IEEE, 2009.
- [52] K. Kinugawa and H. Noborio, "A shared autonomy of multiple mobile robots in teleoperation," in *Robot and Human Interactive Communication*, 2001. Proceedings. 10th IEEE International Workshop on, pp. 319–325, IEEE, 2001.
- [53] Y. Cheung, J. H. Chung, and N. P. Coleman, "Semi-autonomous formation control of a single-master multi-slave teleoperation system," in *Computational Intelligence in Control and Automation, 2009. CICA 2009. IEEE Symposium on*, pp. 117–124, IEEE, 2009.

- [54] N. Xi and T. J. Tarn, "Stability analysis of non-time referenced internet-based telerobotic systems," *Robotics and Autonomous Systems*, vol. 32, no. 2, pp. 173–178, 2000.
- [55] S. Hirche, A. Bauer, and M. Buss, "Transparency of haptic telepresence systems with constant time delay," in *Control Applications*, 2005. CCA 2005. Proceedings of 2005 IEEE Conference on, pp. 328–333, IEEE, 2005.
- [56] I. Polat and C. W. Scherer, "Stability analysis for bilateral teleoperation: An iqc formulation," *Robotics, IEEE Transactions on*, vol. 28, no. 6, pp. 1294–1308, 2012.
- [57] T. Itoh, K. Kosuge, and T. Fukuda, "Human-machine cooperative telemanipulation with motion and force scaling using task-oriented virtual tool dynamics," *Robotics and Automation, IEEE Transactions on*, vol. 16, no. 5, pp. 505–516, 2000.
- [58] I. Elhajj, N. Xi, W. K. Fung, Y.-H. Liu, Y. Hasegawa, and T. Fukuda, "Supermediaenhanced internet-based telerobotics," *Proceedings of the IEEE*, vol. 91, no. 3, pp. 396– 421, 2003.
- [59] P. H. Chang and J. Kim, "Telepresence index for bilateral teleoperations," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 42, no. 1, pp. 81–92, 2012.
- [60] M. R. Rosekind, D. F. Neri, and D. F. Dinges, "From laboratory to flightdeck- promoting operational alertness," *Fatigue and duty time limitations- An international review*, p. 7, 1997.
- [61] H. I. Son, L. L. Chuang, A. Franchi, J. Kim, D. Lee, S.-W. Lee, H. Bulthoff, and P. R. Giordano, "Measuring an operator's maneuverability performance in the haptic teleoperation of multiple robots," in *Intelligent Robots and Systems (IROS)*, 2011 *IEEE/RSJ International Conference on*, pp. 3039–3046, IEEE, 2011.
- [62] M. Eriksson and N. P. Papanikolopoulos, "Driver fatigue: a vision-based approach to automatic diagnosis," *Transportation Research Part C: Emerging Technologies*, vol. 9, no. 6, pp. 399–413, 2001.
- [63] M. Patel, S. Lal, D. Kavanagh, and P. Rossiter, "Applying neural network analysis on heart rate variability data to assess driver fatigue," *Expert Systems with Applications*, vol. 38, no. 6, pp. 7235–7242, 2011.
- [64] M. Grosse-Wentrup, K. Gramann, and M. Buss, "Adaptive spatial filters with predefined region of interest for eeg based brain-computer-interfaces," in Advances in neural information processing systems, pp. 537–544, 2006.

- [65] D. Heger, F. Putze, and T. Schultz, "Online workload recognition from eeg data during cognitive tests and human-machine interaction," in KI 2010: Advances in Artificial Intelligence, pp. 410–417, Springer, 2010.
- [66] M. Y. Bekkedal, J. Rossi III, and J. Panksepp, "Human brain eeg indices of emotions: delineating responses to affective vocalizations by measuring frontal theta event-related synchronization," *Neuroscience & Biobehavioral Reviews*, vol. 35, no. 9, pp. 1959–1970, 2011.
- [67] J. R. Millan, F. Renkens, J. Mouriño, and W. Gerstner, "Noninvasive brain-actuated control of a mobile robot by human eeg," *Biomedical Engineering, IEEE Transactions* on, vol. 51, no. 6, pp. 1026–1033, 2004.
- [68] A. Ferreira, T. F. Bastos-Filho, M. Sarcinelli-Filho, F. A. Cheein, J. Postigo, and R. Carelli, "Teleoperation of an industrial manipulator through a tcp/ip channel using eeg signals," in *Industrial Electronics, 2006 IEEE International Symposium on*, vol. 4, pp. 3066–3071, IEEE, 2006.
- [69] A. Chella, E. Pagello, E. Menegatti, R. Sorbello, S. M. Anzalone, F. Cinquegrani, L. Tonin, F. Piccione, K. Prifitis, C. Blanda, et al., "A bci teleoperated museum robotic guide," in *Complex, Intelligent and Software Intensive Systems, 2009. CISIS'09. In*ternational Conference on, pp. 783–788, IEEE, 2009.
- [70] I. Iturrate, J. M. Antelis, A. Kubler, and J. Minguez, "A noninvasive brain-actuated wheelchair based on a p300 neurophysiological protocol and automated navigation," *Robotics, IEEE Transactions on*, vol. 25, no. 3, pp. 614–627, 2009.
- [71] K. Kiguchi and Y. Hayashi, "A study of emg and eeg during perception-assist with an upper-limb power-assist robot," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 2711–2716, IEEE, 2012.
- [72] Y. Chae, J. Jeong, and S. Jo, "Toward brain-actuated humanoid robots: Asynchronous direct control using an eeg-based bci," *Robotics, IEEE Transactions on*, vol. 28, no. 5, pp. 1131–1144, 2012.
- [73] W. S. Kim and A. K. Bejczy, "Demonstration of a high-fidelity predictive/preview display technique for telerobotic servicing in space," *Robotics and Automation, IEEE Transactions on*, vol. 9, no. 5, pp. 698–702, 1993.
- [74] R. Lindemann, S. Tosunoglu, and D. Tesar, "Construction and demonstration of a nine-string, six-degree-of-freedom force-reflecting joystick for telemanipulation," *Final*

Report to NASA Johnson Space Center, Grant No. NAG9-188, Department of Mechanical Engineering, University of Texas, Austin, Texas, 1987.

- [75] P. Batsomboon, S. Tosunoglu, and D. W. Repperger, "A survey of telesensation and teleoperation technology with virtual reality and force reflection capabilities," *International Journal of Modelling and Simulation*, vol. 20, no. 1, pp. 79–88, 2000.
- [76] S. Waldherr, R. Romero, and S. Thrun, "A gesture based interface for human-robot interaction," Autonomous Robots, vol. 9, no. 2, pp. 151–173, 2000.
- [77] A. Malima, E. Ozgur, and M. Çetin, "A fast algorithm for vision-based hand gesture recognition for robot control," in *Signal Processing and Communications Applications*, 2006 IEEE 14th, pp. 1–4, IEEE, 2006.
- [78] J. Triesch and C. Von Der Malsburg, "A gesture interface for human-robotinteraction," in 2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), pp. 546–546, IEEE Computer Society, 1998.
- [79] C. Breazeal, A. Brooks, J. Gray, G. Hoffman, C. Kidd, H. Lee, J. Lieberman, A. Lockerd, and D. Mulanda, "Humanoid robots as cooperative partners for people," *Int. Journal of Humanoid Robots*, vol. 1, no. 2, pp. 1–34, 2004.
- [80] J. Y. Chai, L. She, R. Fang, S. Ottarson, C. Littley, C. Liu, and K. Hanson, "Collaborative effort towards common ground in situated human-robot dialogue," in *Proceedings* of the 2014 ACM/IEEE international conference on Human-robot interaction, pp. 33– 40, ACM, 2014.
- [81] S. Lauria, G. Bugmann, T. Kyriacou, J. Bos, and E. Klein, "Converting natural language route instructions into robot executable procedures," in *Robot and Human Interactive Communication*, 2002. Proceedings. 11th IEEE International Workshop on, pp. 223–228, IEEE, 2002.
- [82] A. J. Martignoni III and W. D. Smart, "Programming robots using high-level task descriptions," in *Proceedings of the AAAI Workshop on Supervisory Control of Learning* and Adaptive Systems, pp. 49–54, 2004.
- [83] M. N. Nicolescu and M. J. Mataric, "Learning and interacting in human-robot domains," Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, vol. 31, no. 5, pp. 419–430, 2001.

- [84] M. Kloetzer and C. Belta, "Temporal logic planning and control of robotic swarms by hierarchical abstractions," *Robotics, IEEE Transactions on*, vol. 23, no. 2, pp. 320–330, 2007.
- [85] C. Liu, J. Walker, and J. Chai, "Ambiguities in spatial language understanding in situated human robot dialogue," in AAAI 2010 Fall Symposium on Dialogue with Robots, 2010.
- [86] M. Skubic, D. Perzanowski, S. Blisard, A. Schultz, W. Adams, M. Bugajska, and D. Brock, "Spatial language for human-robot dialogs," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 34, no. 2, pp. 154–167, 2004.
- [87] S. Konrad and B. H. Cheng, "Facilitating the construction of specification patternbased properties," in *Requirements Engineering*, 2005. Proceedings. 13th IEEE International Conference on, pp. 329–338, IEEE, 2005.
- [88] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas, "Temporal-logic-based reactive mission and motion planning," *Robotics, IEEE Transactions on*, vol. 25, no. 6, pp. 1370–1381, 2009.
- [89] W. Takano and Y. Nakamura, "Statistically integrated semiotics that enables mutual inference between linguistic and behavioral symbols for humanoid robots," in *Robotics* and Automation, 2009. ICRA'09. IEEE International Conference on, pp. 646–652, IEEE, 2009.
- [90] M. Ralph and M. A. Moussa, "Toward a natural language interface for transferring grasping skills to robots," *Robotics, IEEE Transactions on*, vol. 24, no. 2, pp. 468–475, 2008.
- [91] X. He, T. Ogura, A. Satou, and O. Hasegawa, "Developmental word acquisition and grammar learning by humanoid robots through a self-organizing incremental neural network," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 37, no. 5, pp. 1357–1372, 2007.
- [92] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas, "Translating structured english to robot controllers," Advanced Robotics, vol. 22, no. 12, pp. 1343–1359, 2008.
- [93] Y. Jia, N. Xi, and E. Nieves, "Coordination of a nonholonomic mobile platform and an on-board manipulator," in *Robotics and Automation (ICRA)*, 2014 IEEE International Conference on, pp. 4356–4361, IEEE, 2014.

- [94] Y. Jia, N. Xi, Y. Cheng, and S. Liang, "Coordinated motion control of a nonholonomic mobile manipulator for accurate motion tracking," in *Intelligent Robots and Systems* (IROS), 2014 IEEE/RSJ International Conference on, pp. 1635–1640, IEEE/RSJ, 2014.
- [95] Y. Jia, X. Li, Y. Jiang, and N. Xi, "Sensor-based redundancy resolution for a mobile robotic manipulator," in Unmanned Systems Technology XIII Conference, SPIE Defense, Security, and Sensin, pp. 101–105, SPIE, 2011.
- [96] H. Zhang, Y. Jia, and N. Xi, "Sensor-based redundancy resolution for a nonholonomic mobile manipulator," in *Intelligent Robots and Systems (IROS)*, 2012 IEEE/RSJ International Conference on, pp. 5327–5332, IEEE, 2012.
- [97] Y. Jia, N. Xi, and J. Buether, "Design of single-operator-multi-robot teleoperation systems with random communication delay," in *Intelligent Robots and Systems (IROS)*, 2011 IEEE/RSJ International Conference on, pp. 171–176, IEEE, 2011.
- [98] N. Xi, Event-based planning and control for robotic systems. PhD thesis, Washington University, 1993.
- [99] N. Xi, T. Tarn, and A. K. Bejczy, "Intelligent planning and control for multirobot coordination: an event-based approach," *Robotics and Automation, IEEE Transactions* on, vol. 12, no. 3, pp. 439–452, 1996.
- [100] N. Xi and T. J. Tarn, "Integration of heterogeneity for human-friendly robotic operations," *Journal of Intelligent and Robotic Systems*, vol. 25, no. 4, pp. 281–293, 1999.
- [101] T.-J. Tarn and N. Xi, "Eventbased planning and control for robotic systems: Theory and implementation," in *Essays on Mathematical Robotics*, pp. 31–59, Springer, 1998.
- [102] J. Tan, N. Xi, and W. Kang, "Non-time based tracking controller for mobile robots," in *Electrical and Computer Engineering*, 1999 IEEE Canadian Conference on, vol. 2, pp. 919–924, IEEE, 1999.
- [103] J. Tan, N. Xi, A. Goradia, and W. Sheng, "Coordination of human and mobile manipulator formation in a perceptive reference fame," in *Robotics and Automation*, 2003. *Proceedings. ICRA'03. IEEE International Conference on*, vol. 1, pp. 374–379, IEEE, 2003.
- [104] Y. Jia, N. Xi, F. Wang, Y. Wang, and X. Li, "Controlling telerobotic operations adaptive to quality of teleoperator and task dexterity," in *Intelligent Robots and Systems* (IROS), 2011 IEEE/RSJ International Conference on, pp. 184–189, IEEE, 2011.

- [105] Y. Jia, N. Xi, Y. Wang, and X. Li, "Online identification of quality of teleoperator (qot) for performance improvement of telerobotic operations," in *Robotics and Automation* (ICRA), 2012 IEEE International Conference on, pp. 451–456, IEEE, 2012.
- [106] "Emotiv brain computer interface technology. retrieved april 22, 2014, from http://www.emotiv.com,"
- [107] P. S. INVENTADO, R. LEGASPI, M. SUAREZ, and M. NUMAO, "Predicting student emotions resulting from appraisal of its feedback," *Res. Pract. Technol. Enhanced Learning*, vol. 6, pp. 107–133, 2011.
- [108] J. Gonzalez-Sanchez, M.-E. Chavez-Echeagaray, R. Atkinson, and W. Burleson, "Affective computing meets design patterns: a pattern-based model for a multimodal emotion recognition framework," in *Proceedings of the 16th European Conference on Pattern Languages of Programs*, p. 14, ACM, 2012.
- [109] M. Brown and C. J. Harris, "Neurofuzzy adaptive modelling and control," 1994.
- [110] W.-k. Fung, N. Xi, W.-t. Lo, B. Song, Y. Sun, Y.-h. Liu, and I. H. Elhajj, "Task driven dynamic qos based bandwidth allocation for real-time teleoperation via the internet," in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 2, pp. 1094–1099, IEEE, 2003.
- [111] D. Aarno, S. Ekvall, and D. Kragic, "Adaptive virtual fixtures for machine-assisted teleoperation tasks," in *Robotics and Automation*, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on, pp. 1139–1144, IEEE, 2005.
- [112] J. Kofman, X. Wu, T. J. Luu, and S. Verma, "Teleoperation of a robot manipulator using a vision-based human-robot interface," *Industrial Electronics, IEEE Transactions* on, vol. 52, no. 5, pp. 1206–1219, 2005.
- [113] K. Hauser, R. Alterovitz, N. Chentanez, A. Okamura, and K. Goldberg, "Feedback control for steering needles through 3d deformable tissue using helical paths," *Robotics science and systems: online proceedings*, p. 37, 2009.
- [114] D.-J. Kim, R. Hazlett-Knudsen, H. Culver-Godfrey, G. Rucks, T. Cunningham, D. Portee, J. Bricout, Z. Wang, and A. Behal, "How autonomy impacts performance and satisfaction: Results from a study with spinal cord injured subjects using an assistive robot," Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, vol. 42, no. 1, pp. 2–14, 2012.

- [115] A. D. Dragan, S. S. Srinivasa, and K. C. Lee, "Teleoperation with intelligent and customizable interfaces," *Journal of Human-Robot Interaction*, vol. 2, no. 2, pp. 33– 57, 2013.
- [116] W.-k. Fung, N. Xi, W.-t. Lo, and Y.-H. Liu, "Improving efficiency of internet based teleoperation using network qos," in *Robotics and Automation*, 2002. Proceedings. I-CRA'02. IEEE International Conference on, vol. 3, pp. 2707–2712, IEEE, 2002.
- [117] N. J and W. SJ, Numerical Optimization (2nd Edition). New York: Springer-Verlag, 2006.
- [118] Y. Jia, N. Xi, J. Chai, Y. Cheng, R. Fang, and L. She, "Perceptive feedback for natural language control of robotic operators," in *Robotics and Automation (ICRA)*, 2014 IEEE International Conference on, pp. 4356–4361, IEEE, 2014.
- [119] M. Steedman and J. Baldridge, "Combinatory categorial grammar," Non-Transformational Syntax Oxford: Blackwell, pp. 181–224, 2011.
- [120] M. Scheutz, P. Schermerhorn, J. Kramer, and D. Anderson, "First steps toward natural human-like hri," Autonomous Robots, vol. 22, no. 4, pp. 411–423, 2007.
- [121] P. Gorniak and D. Roy, "Situated language understanding as filtering perceived affordances," *Cognitive Science*, vol. 31, no. 2, pp. 197–231, 2007.
- [122] J. Peltason, H. Rieser, S. Wachsmuth, and B. Wrede, "On grounding natural kind terms in human-robot communication," *KI-Künstliche Intelligenz*, vol. 27, no. 2, pp. 107–118, 2013.
- [123] C. Liu, R. Fang, L. She, and J. Y. Chai, "Modeling collaborative referring for situated referential grounding," in *The 14th Annual SIGdial Meeting on Discourse and Dialogue*, 2013.
- [124] W.-H. Tsai and K.-S. Fu, "Subgraph error-correcting isomorphisms for syntactic pattern recognition," Systems, Man and Cybernetics, IEEE Transactions on, no. 1, pp. 48–62, 1983.
- [125] W. Zhang, State-space search: Algorithms, complexity, extensions, and applications. Springer, 1999.
- [126] P. J. Ramadge and W. M. Wonham, "The control of discrete event systems," Proceedings of the IEEE, vol. 77, no. 1, pp. 81–98, 1989.

- [127] J. Goryca and R. Hill, "Formal synthesis of supervisory control software for multiple robot systems," in American Control Conference (ACC), 2013, pp. 125–131, IEEE, 2013.
- [128] J.-C. Latombe, "Robot motion planning, chapter," 1996.
- [129] Y. Jia and N. Xi, "Coordinated formation control for multi-robot systems with communication constraints," in Advanced Intelligent Mechatronics (AIM), 2011 IEEE/ASME International Conference on, pp. 158–163, IEEE, 2011.
- [130] W. Hahn and A. P. Baartz, Stability of motion, vol. 422. Springer, 1967.