


This is to certify that the
dissertation entitled
CONTROL OF MOBILE MANIPULATORS

presented by
JINDONG TAN

has been accepted towards fulfillment
of the requirements for

PH.D degree in ELECTRICAL & COMPUTER
ENGINEERING


Major professor

Date 05/08/02



PLACE IN RETURN BOX to remove this checkout from your record.
TO AVOID FINES return on or before date due.
MAY BE RECALLED with earlier due date if requested.

DATE DUE	DATE DUE	DATE DUE

CONTROL OF MOBILE MANIPULATORS

BY

JINDONG TAN

A Dissertation

Submitted to

Michigan State University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

2002

ABSTRACT

Control of Mobile Manipulators

By

Jindong Tan

This dissertation presents a new class of control and coordination schemes for mobile manipulators. A mobile manipulator, which consists of a mobile platform and a robot arm, provides a new concept and direction in robot applications due to its capability of infinite motion in a large workspace and dexterous manipulation. Mobile manipulators have various applications in rescue and hazardous fields, battlefield, homes, offices, hospitals, etc. However, motion planning and control of such a system is challenging. This dissertation first presents the design of non-time based tracking controller and its application to nonholonomic mobile platforms. The non-time based tracking control is a key step to ensure the robustness of the mobile manipulator to unexpected obstacles and coordination errors. Then, singularity analysis and motion control of robot arms is discussed. Robot arm singularity avoidance is considered as one of the criteria for coordinating the motion of the mobile platform and the robot arm. Using a hybrid system approach, a singularity-free robot motion controller has also been developed for controlling the robot arm. Further, a unified dynamic model for the integrated mobile platform and on-board manipulator arm is developed. This provides an efficient and convenient framework to design a mobile manipulator controller as well as plan its motions. Combining the non-time based planning and control method with the nonlinear feedback technique, motion and force controllers at task level are designed. An online non-time coordinating scheme is developed to resolve the kinematic redundancy. Based on the decoupled and linearized model, this

dissertation discusses an online non-time based coordination scheme. Force/position control along the same task direction is originally proposed in this dissertation. The proposed force/position control scheme is illustrated using a cart pushing control application, which requires both force control and motion control along the same task direction. An integrated task planning and control scheme while the mobile manipulator interacts with such moving objects is proposed. The proposed approaches are implemented and tested on a mobile manipulator consisting of a Nomadic XR4000 and a Puma 560 robot arm.

Copyright by[®]
JINDONG TAN
2002

To my mother, my grad parents, my brother and for the
memory of my father

ACKNOWLEDGEMENTS

Foremost, I gratefully acknowledge all the invaluable help from my advisor, Dr. Ning Xi, without whom this dissertation would not have been possible. This dissertation comes from numerous discussions in his office, from his keen insight knowledge, from his guidance to a fruitful research area, and his perseverance and support. Not only the knowledge I learnt here, but also the research experience and friendship I gained here will be beneficial to the rest of my life.

I would like to thank all my graduate committee members, Dr. Hassan Khalil, Dr. Fathi Salam, Dr. Robert Schlueter and Dr. Zhengfang Zhou. In addition to their invaluable time they spend on the committee meeting, their insightful comments and suggestions will enhance the technical soundness of this dissertation. I also would like to thank Dr. Wei Kang from Naval Postgraduate School for his valuable help.

I am grateful to my friends and colleagues from Robotics and Automation Lab. The discussions with Weihua Sheng, Imad Elhajj, Jizhong Xiao, Amit Goradia, Henning Hummert, Yu Sun and Heping Chen substantially contributed to my work and broaden my knowledge.

Finally, I would like to reiterate my gratitude to my parents, my grandparents, my brother and sister-in-law. Without their encouragement and continuous support for years, I would not have achieved what I have done today.

TABLE OF CONTENTS

1	INTRODUCTION	1
1.1	Motion Tracking Control of Mobile Robots	2
1.2	Singularity Analysis of Robot Arms	4
1.3	Motion Planning and Control of Mobile Manipulators	7
1.4	Mobile Manipulation Using Force Control	12
1.5	Contributions and Outline of this Dissertation	14
2	MOTION REFERENCE PROJECTION	17
2.1	Introduction to Event-Based Planning and Control	17
2.2	Reference Projection Method	20
2.3	Stability	23
2.3.1	Example	24
2.3.2	Error dynamics and stability	26
2.3.3	Feedback design	31
3	TRACKING CONTROL OF THE MOBILE PLATFORM	35
3.1	An Introduction to the Control of Mobile Platforms	35
3.2	Path Tracking Control for Mobile Robots	36
3.2.1	Mathematical Model of the Robot	36
3.2.2	Tracking Curves with no U-turn	38
3.2.3	Tracking Circles and Ellipses	42
3.3	Experiments and Results	45
3.3.1	Tracking Sinusoid Waves	45
3.3.2	Tracking Circles	48
3.3.3	Tracking Ellipses	48

4	SINGULARITY ANALYSIS AND SINGULARITY-FREE CONTROLLER OF THE ARM	53
4.1	Introduction	53
4.2	Robot Arm Singularities	56
4.2.1	Non-redundant Robot with Spherical Wrist	57
4.2.2	Non-redundant Robot with Non-spherical Wrist	61
4.2.3	Singularity Analysis for a Redundant Robot	62
4.2.4	Workspace and Singular Conditions	63
4.3	Hybrid Robot Motion Controller	64
4.3.1	Hybrid System Model	64
4.3.2	Design Procedure of a Hybrid Robot Motion Controller	65
4.3.3	Hybrid Motion Controller for a PUMA-like robot manipulator	68
4.4	Stability Analysis of the Hybrid Motion Controller	78
4.5	Experimental Implementation and Testing	83
5	CONTROL OF MOBILE MANIPULATORS	89
5.1	Introduction	89
5.2	Unified System Model	91
5.2.1	Kinematic model	91
5.2.2	Dynamics Model	93
5.3	Motion Control and Redundancy Resolution	95
5.3.1	Tracking Controller	95
5.3.2	Singularity Analysis and Redundancy Resolution	96
5.4	Experimental Implementation and Results	100
5.4.1	Hardware and Software Implementation	100
5.4.2	Path Tracking and Tele-operation	101
5.4.3	Path Tracking with Unexpected Obstacles	102

6	INTEGRATED SENSING AND CONTROL OF MOBILE MANIPULATOR	108
6.1	Introduction and Force Control Schemes Revisit	108
6.2	Output Force/Position Control of Redundant Robots	110
6.3	Integrated Task Planning and Control	113
6.3.1	Motion Planning of the Cart	114
6.3.2	Force Planning of the Cart	116
6.3.3	Integration of Sensing, Task Planning and Control	119
6.4	Experimental Implementation and Results	121
6.4.1	Pushing Along a Straight Line	122
6.4.2	Turning the Cart at a Corner	123
6.4.3	Pushing Along a Sine Wave	124
7	CONCLUSIONS	126

LIST OF TABLES

LIST OF FIGURES

1.1	A mobile manipulator	2
1.2	Path tracking of mobile manipulators	5
1.3	The mobile manipulator and nonholonomic cart system	12
2.1	The comparison	18
3.1	The configuration of a mobile robot.	37
3.2	Tracking a sinusoid curve with initial error.	46
3.3	Time based control vs Non-time based control.	47
3.4	Tracking a circle without initial error	49
3.5	Tracking a circle without initial error	49
3.6	Tracking an ellipse without initial error	50
3.7	Tracking an ellipse with initial error	50
3.8	Tracking an ellipse with present of unexpected obstacle	51
4.1	Singular configurations of Puma 560	60
4.2	Illustrative plot of the robot workspace and subspaces	64
4.3	Hybrid system controller framework	75
4.4	A typical switching sequence of the hybrid motion controller	81
4.5	Experimental Setup	83
4.6	At boundary singular configuration, with DLS method only. Dash line: Desired; Solid line: Actual	84
4.7	At boundary singular configuration, with hybrid motion controller. Dash line: Desired; Solid line: Actual	84
4.8	At interior singular configuration, with DLS only. Dash line: Desired; Solid line: Actual	85
4.9	At interior singular configuration, with hybrid motion controller. Dash line: Desired; Solid line: Actual	85

4.10	With both interior and wrist singular configurations. Dash line: De- sired; Solid line: Actual	86
4.11	With wrist singularity. Dash line: Desired; Solid line: Actual	86
5.1	Mobile manipulators	90
5.2	Event-based coordination between the arm and the mobile platform .	98
5.3	Hardware structure of the mobile manipulator	101
5.4	Software structure of the mobile manipulator	104
5.5	Multiple tasks path tracking	105
5.6	Multiple tasks path tracking	105
5.7	Tele-operation	106
5.8	Tele-operation	106
5.9	Teleoperation with unexpected obstacles	107
6.1	Hybrid Force/Position Control	109
6.2	Impedance Control	110
6.3	Hybrid Force/Position Control for Redundant Robots	111
6.4	Integrated Task Planning and Control for the Mobile Manipulation of a Nonholonomic Cart	114
6.5	A mobile manipulator and a nonholonomic cart	115
6.6	Configuration of the cart in the moving frame	120
6.7	Laser range data	121
6.8	Pushing the cart along a straight line	122
6.9	Pushing with unexpected obstacle	123
6.10	Turning the cart at an corner	124
6.11	Pushing the task along a sine wave	125

CHAPTER 1

INTRODUCTION

A mobile manipulator, which generally consists of a robot arm and a mobile platform, provides a new concept and direction in robot applications due to its capability of large workspace and dexterous manipulation. The robot arm is mounted on a mobile platform, as shown in Figure 1.1. A mobile manipulator combines the mobility of the platform and the dexterous manipulation capability of the robot arm. The manipulation capability on a mobile platform is called mobile manipulation[37]. The mobile manipulation capability is important for some autonomous mobile robot applications, which require both a large workspace and dexterous manipulation capability, such as reconnaissance and surveillance in rescue and hazardous fields, battlefield, homes, offices, hospitals, etc. The mobility of the (mobile) platform substantially increases the performance capabilities of the system. For example, the platform increases the size of the manipulator workspace and enables the manipulator to position the end-effector for executing the task efficiently, while a fixed base manipulator arm is limited by the size of its workspace. From a different point of view, the mobile manipulator has high degree of kinematic redundancy. Because of the kinematic redundancy, the functionality of the mobile manipulator is increased to a greater extent. The redundant degrees of freedom can be used to accomplish secondary tasks. For example, joint torques can be optimized, singular configurations of the manipulator can be avoided and hybrid force/position control can be achieved.

The objective of this dissertation is to investigate the modeling, redundancy resolution and control, integrated task planning and control of mobile manipulators. The unique force control properties while the mobile manipulator interacting with moving objects has also been probed. Among others, the following issues of mobile manipulators will be discussed:

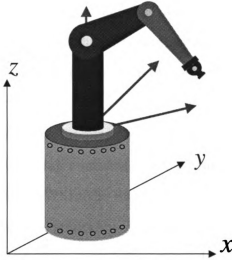


Figure 1.1: A mobile manipulator

- Motion tracking control of mobile robots
- Singularity analysis of robot arms
- Singularity-free tracking control of robot arms.
- Unified model of mobile manipulators and online redundancy resolution
- Decoupled force/position control along the same task direction
- Integrated tasking planning, sensing, and control of mobile manipulators

A quick survey of the ongoing research on related topics is given below.

1.1 Motion Tracking Control of Mobile Robots

Most of the tasks of a mobile robot, which generally equipped with a computer and sensors, are to move from one configuration to another configuration, or track certain path in its workspace. The research on mobile robots is generally focused on motion planning, localization, navigation, sensor fusion, tracking controller design. In this dissertation, a non-time based approach for the path tracking controller design of

mobile robots will be discussed. The non-time based robot controller enables the robot to handle unexpected events while tracking certain path. The non-time based approach can also be used for the coordination of the mobile platform and the robot arm.

In the control of autonomous mobile robots, unexpected events may change the speed and path of a vehicle during a mission (such as traffic light, unexpected obstacles, and system malfunction). When the unexpected event is over, the robot should recover its normal performance smoothly. A time-dependent trajectory controller drives a vehicle to its desired place at any given time. It speeds up the vehicle after a temporary slow down to catch up with the schedule. However, in many applications the important task is to track the desired path with the desired speed, keeping up with the schedule is not a first priority or not desired. A common approach for this problem is to use a high-level rule-based controller to decide the speed and the reset time. However, the non-time based controller proposed in this dissertation is a different approach. The proposed controller is independent of time. For the purpose of path tracking (with a desired speed depending on the location of the path), the controller recovers the normal path and speed from unexpected event without resetting time or speed. Furthermore, the algorithm introduces a reference projection, a function of state variables that replaces the time variable.

When multiple vehicles are involved in the same mission (such as formation control of multiple agents), reference projection is a tool that can be used for cooperated motion control of multiple vehicles. This idea of using reference projection for cooperated control was used in the formation control of multi-satellite systems ([71] and [34]). For the mobile manipulator, instead of using time as the reference, the motion of the mobile robot is referred to the trajectory of the end effector of the robot arm. The motion reference will be developed to coordinate the motion of the platform and the robot arm according to certain criteria such that the robot arm will always

working in a dexterous workspace, as shown in Figure 1.1.

Developing non-time based controller has attracted the attention of researchers from different fields. In [81], a non-time based controller called event-based design was introduced. It has successful applications in a variety of robot control problems ([81], [80], [76], [83] and [78]). In addition to robot control, [32] studies the tracking of a trimming trajectory for unmanned air vehicles using non-time based control. In [51], non-time based control of autonomous underwater vehicles is discussed. In [34], a time varying feedback is converted to a non-time based controller to achieve coordination among multiple satellites flying in formation. Although different terminologies were used in different applications, the author found that finding a path tracking controller independent of time is a common interest among a variety of application areas (see additional publications [22], [57], [59], [28], [25], [63], [33] and the references therein). The contribution of our work is to introduce a reference projection method in the design of non-time based controller.

1.2 Singularity Analysis of Robot Arms

A robot task is usually described in its task space. The direct implementation of a task level controller can provide significant application efficiency and flexibility to a robotic operation[68]. Implementation of task level controller becomes even more important when robots are working in coordination with humans[79]. The human intuition on a task is always represented in the task space. The existence of kinematic singularities is a major problem in the application of task level controller. While approaching a singular configuration, the task level controller may generate high joint torques, which result in instability and large errors in the task space. The task level controller is not only invalid at the singular configurations, but also in certain neighborhood of the singular configurations. Therefore, the singularity of robot arms is analyzed and a singularity-free path tracking controller is proposed. Different from a nonredundant

robot arm, a specific task for mobile manipulators can be performed by moving only the mobile platform or the manipulator, or moving the platform and the manipulator simultaneously, as depicted in Figure 1.2. It is ideal to coordinate the motion of the platform and the arm in such a way that tasks can be performed by moving them simultaneously. This involves the kinematic redundancy resolution of mobile manipulators. The objective to resolve the redundancy is to position the mobile base such that the robot arm will always be in a dexterous configuration and avoid singular configuration. The singularity analysis of robot arms is then used in the coordination of the mobile platform and the robot arm. A brief review of the research around this topic is given below.

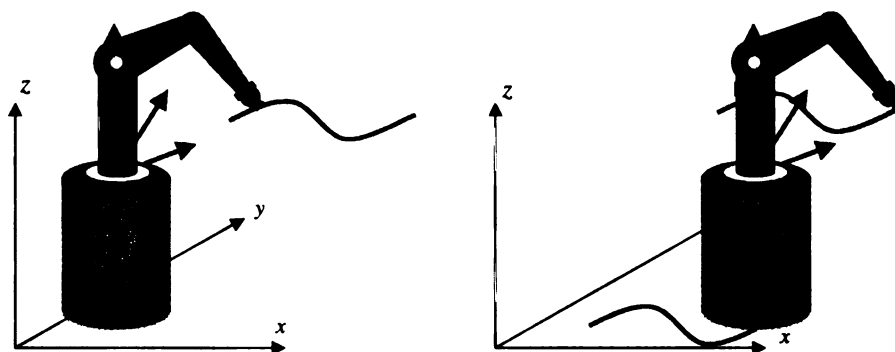


Figure 1.2: Path tracking of mobile manipulators

The research in robot arm singularities has attracted researchers' attention. The Singularity-Robust Inverse(SRI) [15, 17, 41, 47] method was developed to provide an approximate solution to the inverse kinematics problem around singular configurations. Since the Jacobian matrix becomes ill-conditioned around the singular configurations, the inverse (pseudoinverse) of the Jacobian matrix results in unreasonable torques in task level controller. Instead of using exact motion control resulting in large torques, the SRI uses Damped Least-Squares(DLS) to provide approximate motion control close to the desired Cartesian trajectory. Chiaverini, Siciliano and Egeland [17] refined the basic DLS method to improve tracking errors from the desired path by

varying the damping factors. Caccavale, Chiaverini and Siciliano elaborated the DLS method by considering the velocity and acceleration variables [11]. These methods reduce the torque applied to individual joints and achieve approximate motion. By allowing an error in the motion, the SRI method makes the robot pass close to the singular point but not go through it. Actually, it can be shown [41, 50] that the system is unstable at the singular points, which means either the singular points can not be actually reached or the robots can not start from the singular points. Obviously, this creates difficulties for many applications. Kirćanski *et al.*[41] theoretically proved that at the singular points, the resolved acceleration control of robots has unstable saddles. The instability of this method for redundant mechanisms is discussed in [50].

The path tracking approach [38, 39, 48] augments the joint space by adding virtual joints to the manipulator and allowing self motion. Based on the predictor-corrector method of path following, this approach provides a satisfactory solution to the path tracking problem at singular configurations. Timing is not considered at the time of planning and it is reparameterized in solving the problem. However, when a timing is imposed on the path, it forces the manipulator to slow down in the neighborhood of singular configurations and to stop at the singular configuration.

The normal form approach in [46] provides a solution of inverse kinematics in the entire joint space. With appropriately constructed local diffeomorphic coordinate changes around the singularity, the solution of inverse kinematics can be found and then transformed back to the original coordinates. The normal form method involves heavy computation. Hence is extremely difficult to implement it in real-time. Duleba[23] proposed the the channel algorithm of transversal passing through singularities for non-redundant robot manipulators. The basic idea of channel algorithm is to smoothly change a component of the joint space as the neighborhood of singularity is entered. Thus a continuous joint coordinate can be obtained while tracking a desired path in a task space.

Another method has also been developed to incorporate the dynamic poles of the system by Sampei and Furuata [56]. They proposed a time re-scale transformation method for designing a robot controller, which achieves slow poles in the vicinity of singularity and fast poles in the regular area. It can be shown that this method results in similar error dynamics to the SRI method. For the application of switching control to the robot manipulators, Bishop and Spong [7] proposed a switching control method for redundancy resolution of redundant robots. The damped velocity method was used to maximize the manipulability. Torques and joint velocities are limited utilizing the redundancy of the robot.

1.3 Motion Planning and Control of Mobile Manipulators

Mobile manipulators provide numerous advantages compared with the fixed base manipulator, path planning and control of such a system are challenging. First, there is a redundancy resolution problem. Secondly, the dynamic model of a mobile manipulator is complex, and there is a high degree of coupling between the platform and the arm. The dynamics of the mobile platform and the manipulator arm are generally different. The former is heavy and has slow dynamics. The later is relatively light and has fast dynamics, *i.e.*, a wider bandwidth. Thirdly, many applications of the mobile manipulator require the robot to interact with its environment dynamically while providing motion, such as pushing, pulling, cutting, excavating. Implementation of these tasks demands the mobile manipulator to provide both output force control to overcome the resistance and friction of the objects, and motion control to track certain trajectory. Force control schemes for the mobile manipulator are therefore to be considered. The myriad use of the mobile manipulator and the difficulties in its control and motion planning bring a lot of research issues to light. Motion planning, modeling and control, coordination between the mobile platform and the manipulator arm, force control are topics of this dissertation. Previous work related to these topics

in the literature is reviewed.

A: Motion Planning. A solution to kinematic redundancy is to coordinate the mobile platform and robot manipulator to perform secondary tasks. Seraji [62] proposed an on-line coordinated motion control method by treating the mobile manipulator as a redundant mechanical systems. The mobility and manipulation are treated in a common framework. Secondary tasks for redundancy resolution are selected based on task specifications.

Egerstedt and Hu [24] proposed a coordinated virtual vehicle solution to the trajectory following problem for mobile manipulators. Given a desired trajectory for the end-effector, a trajectory for the base is planned in such a manner that the desired end effector position is within the specified workspace of the arm. The mobile platform and the manipulator then follow their respective reference trajectories. The two subsystems are controlled separately and coordinated.

The kinematic redundancy can also be resolved using criteria which are used to optimize application specific requirements, such as minimum torque output, minimum distance, *etc.* Chen and Zalzal [13] discussed trajectory generation while taking the dynamics of the system into consideration. The optimal trajectory generation problem with nonholonomic constraints and obstacle avoidance were formulated as a nonlinear multi-criteria optimization problem. Given the trajectory of the end-effector, near optimal trajectories for the mobile platform and manipulator joints are obtained by an efficient genetic algorithm. The algorithm considers the optimization of torque and obstacle avoidance.

When a mobile manipulator performs multiple tasks in a sequence, the final configuration of each task becomes the initial configuration of the subsequent task. The configuration between two tasks is called commutation configuration [54]. In the motion planning for a sequence of tasks, the motion trajectories together with commutation configuration need to be considered simultaneously. If the commutation configura-

tions are determined without considering the current task in a global perspective, a subsequent task may be rendered difficult. Most of the redundancy resolution approaches for executing a sequence of tasks are criteria based. For instance, Pin and Culioli [54] used multi-criteria optimization schemes for motion planning. The commutation configurations become boundary conditions for each task in the sequence. A variety of optimization criteria such as obstacle avoidance, reach and maneuverability are developed and the algorithms for optimization problems are discussed. However, the algorithm is off-line and nonholonomic constraints were not considered. Zhao, Ansari and Hou [85] formulated multiple task path planning as a nonlinear and non-polynomial optimization problem. The optimal sequence of platform positions and manipulator configurations are obtained by using genetic algorithms. Lee and Cho [42] proposed a different motion planning method for mobile manipulators to execute a sequence of tasks. They considered the motion trajectories and the commutation configurations simultaneously in a global cost function instead of treating them differently. A global optimization problem using the necessary conditions, which minimize the objective function, can be applied to both holonomic and nonholonomic systems.

From the above review, it can be seen that motion planning of the platform and the manipulator arm is crucial for the coordination of the mobile manipulator. The existing approaches to coordinate the platform and the robot arm, or in the other world, to resolve the kinematic redundancy, are either based on additional tasks or based on the optimization of criteria. For the additional tasks based approach, the end effector tasks may require an extension of the end effector out of its workspace. Since the motion of the platform and the manipulator is treated equally, mobility and manipulation ability are not optimized. For the criteria optimization based approach, a near optimal path for both the platform and the manipulator arm can be obtained. However, it is generally computationally intensive and not eligible for realtime control. Since most of the optimization algorithms for redundancy resolution is computation

intensive and have to be used off-line, this dissertation presents an online coordination scheme based on the event-based approach. It optimizes the manipulation capability of the robot arm and can avoid unexpected obstacles. The kinematic redundancy resolution will be investigated based on the analysis of the robot arm singularities in this dissertation.

B: Modeling and Control. To achieve effective and practical control, various dynamic models and control schemes are proposed in the literature. The control strategies are generally classified into two categories. One is decentralized control of the mobile platform and the manipulator arm. The controller for each part is constructed separately and then the interaction between the platform and the manipulator is considered. The other is a unified approach for the control of both the mobile platform and the robot arm. In this case, the mobile manipulator is treated as a redundant manipulator.

By considering the mobile robot and the manipulator as two subsystems, Liu and Lewis [43] developed a decentralized robust controller for a mobile manipulator. The reaction forces are considered in the dynamic models of the two subsystems. However, only the translation of the mobile robot is considered in the paper. Chung and Velinsky [18, 19] also derived the dynamic model of nonholonomic mobile robots and manipulators separately. The kinematic redundancy is resolved by decomposing the mobile manipulator into two subsystems, the mobile platform and the manipulator. A robust interaction control algorithm, in which suitable controllers are designed for the two subsystems, is developed. The algorithm is employed to minimize the adverse effect of the wheel slip on the tracking performance.

Yamamoto and Yun [84] studied a two-linked planar mobile manipulator subject to nonholonomic constraints. The dynamics model was derived and reformulated according to the nonholonomic constraints. Output feedback linearization is used to simplify the nonlinear model and derive the controller. The mobile platform and

the manipulator are coordinated based on the concept of a preferred operating region. However, if the manipulator was commanded to follow a spatial trajectory in the task space, it is difficult to keep the arm in a preferred region during a task execution. Colbaugh, Trabatti and Glass [20] considered the dynamics model for redundant nonholonomic mechanical systems and reformulated it into two parts, a reduced dynamics model and a kinematic relationship which considers the nonholonomic constraints. Adaptive control schemes were developed based on the two parts. The mobile platform and the arm are coordinated such that the distance between the end-effector and the mobile base does not exceed a pre-specified value.

Two problems are brought to light for the control structure of the mobile manipulator system, one is to ensure the system output regardless of the output of the subsystems. It is therefore desirable to model the mobile manipulator such that motion planning and force/position control can be considered in a unified task space coordinate frame. The other is the system stability in the case of unexpected obstacles or desynchronization of the two subsystems. The non-time based controller design for the coordination is investigated for this purpose.

C: Force Control Schemes. Robot manipulator force control schemes such as external force control, hybrid position/force control, impedance control have been applied to the force control of the mobile manipulators. Umeda *et. al.* [73] discussed the hybrid position/force control of mobile manipulators. In order to obtain desired hybrid control with respect to the end-effector's tasks, the concept of performance indices based on an equivalent mass matrix is introduced. A weight matrix is also proposed to obtain adequate torque distribution. Antonelli, Sarkar and Chiaverini [3] presented a force control scheme for underwater vehicle manipulator systems. The environment is modeled as a frictionless and elastically complaint plane. External control structure is used and kinematic redundancy is considered. Perrier, Cellier and Dauchez [53] compared the robustness of hybrid position/force control structure

and external force control structure with respect to external disturbances for a mobile manipulator. The effect of mobile platform velocities on the force control structures were discussed.

Due to the kinematic redundancy, the force control and position control of mobile manipulators can be decoupled along the same task direction. This dissertation will discuss the decoupled force/position control based on a linearized and decoupled model.

1.4 Mobile Manipulation Using Force Control

The integrated task planning and control of a mobile manipulator performing the task of a moving object has relatively received less attention. Based on the decoupled force/position control of mobile manipulator, this dissertation presents an integrated task planning and control approach for the task of the mobile manipulator pushing a nonholonomic cart. As shown in figure 1.3, the mobile manipulator and nonholo-

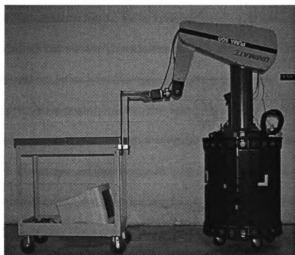


Figure 1.3: The mobile manipulator and nonholonomic cart system

nomic cart system is similar to a tracker-trailer system. In a tracker-trailer system, there is generally a steering mobile robot and one or more passive trailer(s) connected

together by rigid joints. The tracking control and open loop motion planning of such a nonholonomic system has been discussed in the literature. The trailer system was controlled to track certain trajectory using a linear controller based on the linearized model[22]. Exact linearization and nonlinear control of the track-trailer system has been discussed in [58] and experiments on backward steering the trailer have been reported in [40]. Instead of pulling the trailer, the tracker pushes the trailer to track certain trajectories in the backward steering problem. Fire truck steering is another example for pushing a nonholonomic system. A chained form has been used in the open loop motion planning of such a system [10]. Based on the chained form, motion planning for steering a nonholonomic system has been investigated in [45, 70]. Time varying nonholonomic control strategies for the chained form can stabilize the tracker and trailers system to certain configurations[60].

The planning and control of the mobile manipulator and nonholonomic cart system is different from a tracker-trailer system. They are not linked by a rigid joint, but the robot arm. Therefore, the mobile manipulator has more flexibility and control while maneuvering the nonholonomic cart. In a tracker-trailer system, control and motion planning were considered based on the kinematic model, and the trailer is steered by the motion of the tracker. In a mobile manipulator and nonholonomic cart system, the mobile manipulator can manipulate the cart by a regulated output force. The nonholonomic cart can be controlled at a dynamic level by the output force of the mobile manipulator. However, this requires full integration of the motion planning and force planning of the cart, and the planning and control of mobile manipulators.

Though ample literature discusses force control in the context of manipulator arms, little of it considers the force control of mobile manipulator and none of it discusses the out force and position control along the same task direction. This dissertation considers the output force control and position control simultaneously making use of the kinematic redundancy.

1.5 Contributions and Outline of this Dissertation

This goal of this dissertation is to investigate new control and coordination algorithms for a mobile manipulator. The contributions to the control and coordination strategy of mobile manipulators can be grouped into four parts. The first part discusses a motion reference projection method for non-time based tracking controller design and its application to nonholonomic mobile platforms. The proposed design method for non-time based tracking controller converts a controller designed using traditional time-based approach to a new non-time based controller. This approach significantly simplifies the design procedure to achieve a non-time based controller. This method is applied to tracking controllers of mobile platforms. The experimental results have clearly demonstrated the advantage of the non-time based tracking controller in the presence of unexpected obstacles. Applying this method to the mobile manipulators, a common motion reference based on the measurable system output and desired path can coordinate the motion of the platform and the manipulator arm. The advantage of the non-time based tracking controller while coordinating with robot is also shown. More importantly, the results provides an efficient and systematic approach to design non-time based tracking controller for general nonlinear dynamical systems. This part is discussed in Chapter 2, Chapter 3 and part of Chapter 5

The second part presents singularity analysis of manipulator arms and a hybrid system design of singularity-free motion controller for manipulator. Since the mobile platform creates redundancy to mobile manipulator system, the redundancy enables the manipulator arm to keep its optimal configurations and avoid singular configurations. Chapter 4 analyzes the singular configuration of robot arms, which will provides a mechanisms for the coordination of the mobile platform and the robot arm. A singularity-free motion controller is also designed for the manipulator using a hybrid system approach.

The third part formulates the unified control of mobile manipulators and discusses

the redundancy resolution. A unified dynamic model for the mobile manipulator is derived and nonlinear feedback is applied to linearize and decouple this model. The advantage of this model lies in that the kinematic redundancy and dynamic properties of the platform and the robot arm have been considered in the unified model. Based on this model, mobile manipulator tasks such as path tracking control, force/position control can be easily designed. Based on the analysis of the robot arm, an online event-based coordination scheme is proposed for the cooperation of the platform and the robot arm. This online approach is suitable for a point-to-point task, multiple segment tasks and teleoperation. The mobile manipulator system is stable even in the case of appearance of unexpected obstacles and the manipulation capability can adapt to a best possible values during the operation. This part is discussed in Chapter 5.

The forth part discusses the force/position control while the mobile manipulation interacting with moving objects. Based on the decoupled and linearized model, force/position control along the same task direction is discussed utilizing the kinematic redundancy of mobile manipulators. For tasks that require both force and position control along the same task direction, such as pushing a nonholonomic cart, an integrated task planning and control for mobile manipulators is proposed. The approach for manipulating the nonholonomic cart integrates the planning and control of the mobile manipulator, the motion planning and force planning of the nonholonomic cart, and the estimation of the cart configuration based on the sensors of the mobile manipulators. The integrated task planning and control enables the robot to interact with moving objects and complete complicated tasks by the output force. The approaches discussed in this dissertation can be used for the motion planning and control of mobile manipulators which can find applications in manufacturing, hospitals, homes and offices. Experimental results have demonstrated the advantage of the integrated task planning and control approach. This part of the work is presented in

Chapter 6

Chapter 7 concludes this dissertation.

CHAPTER 2

MOTION REFERENCE PROJECTION

2.1 Introduction to Event-Based Planning and Control

The basic idea of event-based planning and control [77] is to introduce the concept of a motion reference, a parameter of the desired path, as shown in Figure 2.1(b). This parameter is directly relevant to the sensing measurement and the task. Instead of time, the control input is parameterized by the action reference parameter. Since the action reference is a function of the real time measurement, the values of the desired vehicle states are functions of the measured data. As a result, for any given instant of time, the desired input is a function of system output. This creates a mechanism to adjust or modify the plan based on the measurements. Thus, the planning becomes a closed loop real-time process. The plan (desired path) can be perceived as an “abstract” entity, like a function. It only has a “real” value when measurements are made. The non-time based planning and control scheme and the traditional time-based planning and control scheme are compared in Figure 2.1(b). In this dissertation, event-based and non-time based are used alternatively.

In the non-time based planning and control scheme, the function of the Motion Reference block in Figure 1(b) is to compute the action reference parameter s on-line, based on sensory measurements. The planner generates the desired value to the system, according to the on-line computed action reference parameter s . In addition, the motion reference parameter is calculated near or at the same rate as the feedback control. In other words, the action plan is adjusted at very high rate, which enables the planner to handle unexpected or uncertain discrete and continuous events. For example, the event-based planning and control scheme has been successfully applied to deal with unexpected obstacles in a robot motion [81]. If the robot is blocked by obstacles, the action reference also stops evolving. As a result the desired input will

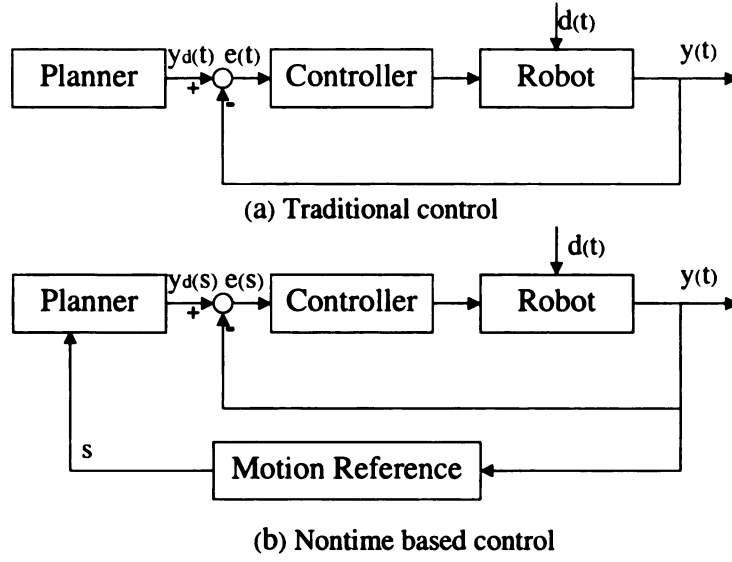


Figure 2.1: The comparison

be constant and the error will remain unchanged. In this case, if the conventional time-based motion plan were implemented, the system error would keep increasing as the planner generates the desired input according to the original fixed plan. The system would therefore become unstable. To solve this problem with the traditional method, one has to incorporate a higher-level monitoring function such as “move until touch” or some kind of compliant motion strategy, both requiring extra sensing (force/torque or touch/contact sensing, see [44] and [16]). Furthermore, once the obstacle is removed, the traditional methods require planning again to complete the task. But our non-time based planning and control in this case does not require any extra force-torque or contact sensing, or planning again. The experimental results in [81] clearly demonstrate the feature.

For a path tracking task, the most natural thought is to pick the distance traveled along the given path as the motion reference s . After s is chosen, the motion along

the given path can be described as [77]

$$\begin{cases} \frac{ds}{dt} = v \\ \frac{dv}{dt} = a \end{cases} \quad (2.1)$$

where v and a are velocity and acceleration along the motion reference s . In order to obtain a non-time based trajectory plan, it is necessary to perform a transformation of variable by replacing the independent scalar variable t in eq. (2.1) by the independent scalar variable s , which is computed from system output. Basically, the event-based trajectory planning is to find the motion profile as a function of the given event-based motion reference s , $v = v(s)$ and $a = a(s)$, subject to the kinematic and dynamic constraints of the robot.

Different from [77], which focuses on applications of robotic arms, the framework here is based on general dynamical systems of moving vehicles, including robotic arms, ground vehicles, aircraft, spacecraft, and surface and underwater vehicles. Except for some ideas used in [77], the viewpoint adopted in this dissertation is different from some existing non-time based approach. Instead of developing design algorithms based on the specific features of the desired path ([32]), the controllers are first designed in the time domain using any existing tracking controller design method. Then, the time variable in the feedback is substituted by a special transformation called a motion reference projection. The final resulting feedback becomes non-time based feedback. It drives the system asymptotically approaching the desired path.

This approach has the following four advantages. First, the model of the system is a general nonlinear dynamical control system. The method is applicable to a wide range of tracking control problems of autonomous vehicles or robotic arms. It has the potential of performing cooperated control of multiple vehicles with different models moving in formation ([71] and [34]). Second, a transformation is provided to transform

a time dependent controller into a non-time based controller. It bridges the traditional state trajectory tracking with the non-time based design. The results enable us to design non-time based feedback using the existing, well known methods of feedback design in time domain such as LQR and H_∞ control, and feedback linearization. The third advantage is the flexibility in choosing motion references. The arc length of the desired path is often used as the motion reference. However, there are many curves for which the formula of computing the arc length is complicated. Parameters other than arc length can be used as motion references in the design to simplify the controller. In the proposed method, the motion reference can be any parameter. The fourth advantage of our method is that the transformation from the state space to the action reference needs not to be the orthogonal projection. For many curves, orthogonal projection requires on-line numerical solution of minimization. Our method allows the designer to use any mapping satisfying a projection condition. A good choice of the reference projection can dramatically reduce the on-line computational load.

In the following, the concept of reference projection method is first introduced. The stability of the non-time based controller is studied and a general method of non-time based tracking control feedback design is developed.

2.2 Reference Projection Method

Mechanical systems and electrical systems are modeled by differential equations in which the free variable is time t . A desired trajectory is usually modeled as a function of time, which is generally denoted by $x_d(t)$. The variable x represents the state of the system. Controllers are designed so that the trajectory of the system $x(t)$ 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. However, in many applications of autonomous vehicles,

a feedback controller independent of time is preferred. Furthermore, the desired path is not necessarily defined as a function of time. It is defined by parametric equations. The parameter is called *motion reference* or *action reference*. It is denoted by s . For instance, a desired path can be defined by a parametric equation $x_d(s)$, where s is the arc length, or s is the projection of x_d to a coordinate axis. In this section, we prove a result, which transforms a time dependent feedback controller into a time independent controller to asymptotically track a desired path driven by action reference. A system is defined by the equation

$$\begin{aligned} \dot{x} &= f(x, u), \quad x \in \mathbb{R}^n, u \in \mathbb{R}^m, \\ y &= h(x), \quad y \in \mathbb{R}^k \end{aligned} \tag{2.1}$$

where u is the control input, and x is the state of the system. The desired path usually does not involve all the state variables explicitly. For example, to make a mobile robot following a curve, the desired path is defined by the position of the system. The state variables such as the orientation of the vehicle, the angle of the front wheels do not appear explicitly in the desired path. Suppose that $y = h(x)$ represents the output of the system. The desired path is given by a parametric equation

$$y = y_d(s),$$

where s is the motion reference.

The first step in the controller design is to define a corresponding path in time domain. If s is not time, then a strictly increasing function $s = v(t)$ is assigned. How to pick the function $v(t)$ depends on the desired speed. For example, if the problem requires that the system is operated so that s is increasing at a constant speed v_0 , then

$$v(t) = v_0 t.$$

Notice that the desired trajectory is defined by two functions, $y_d(s)$ and $s = v(t)$, not by a single desired trajectory $y_d(t)$. We do this on purpose because moving a vehicle along a curve, and how fast a vehicle moves are two relevant but different issues in the planning. They should not be mixed together. In the implementation of the tracking control, $v(\cdot)$ can be a proportional to time t , such as $v(t) = v_0 t$. It can also be related to the present of obstacles and the task specifications. Denote d as the distance from the robot to the obstacle in the moving direction, $v(\cdot)$ can be designed as $v(d, t)$ such that obstacle avoidance can be considered.

The second step is to find a feedback, $u(x, t)$, to track the path $y_d(v(t))$. There exist many well-known design algorithms in the literature of control theory. The feedback satisfies

$$\lim_{t \rightarrow \infty} (h(x(t)) - y_d(v(t))) = 0.$$

Furthermore, if the initial condition is on the desired path, then the trajectory of the controlled system follows the path. More specifically, there exists an initial condition of the system x_0 such that $h(x(t)) = y_d(t)$. Denote this path by $x_d(s)$ or $x_d(v(t))$.

The third step is to find a suitable transformation, $s = \gamma(x)$, from the state space to the reference s . The transformation satisfies

$$\gamma(x_d(s)) = s \tag{2.2}$$

For example, given any state x_0 , let $x_d(s_0)$ be the orthogonal projection from x_0 to $x_d(s)$. If $\gamma(x_0) = s_0$, then it satisfies (2.2). The transformation satisfying (2.2) is called a *reference projection*. However, orthogonal projection is not the only way to define γ . In section 4.1, the projection to x -axis is adopted as γ , which is a simpler formula than orthogonal projection for the specific application studied in that section.

The last step is to construct the non-time based feedback. A *virtual time* is defined

by

$$T(x) = v^{-1}(\gamma(x)). \quad (2.3)$$

It is worthy noting that virtual time is given as a function of system output. The system output is finally a function of time, the actually system operation is considered in the virtual time $T(x)$. By substitute the virtual time $T(x)$ for real time t , the non-time based feedback is

$$u(x) = u(x, T(x)) \quad (2.4)$$

where $\tilde{u}(x, t)$ is the feedback found in the second step, $\gamma(x)$ is a reference projection. The closed-loop system is

$$\dot{x} = f(x, u(x, T)). \quad (2.5)$$

2.3 Stability

In the above design process, the feedback $u(x, t)$ drives the system asymptotically approaching $x_d(t)$. The non-time based design is given by substituting the time variable t in $u(x, t)$ by the virtual time $v^{-1}(\gamma(x))$. Because of (2.2), the trajectory $x(t)$ of (2.5) equals $x_d(t)$ if $x(0)$ is on the desired trajectory x_d . Based on our analysis and simulations, the non-time based feedback is likely to drive the system asymptotically approaching $x_d(s)$. However, the substitution $t = T(x)$ may change the stability of the system in some cases as shown in the following example. In this section, sufficient conditions are given to guarantee the asymptotic stability of the tracking error under non-time based feedback.

2.3.1 Example

Two feedbacks are given in this example for the same tracking problem. One feedback result in a stable non-time based controller, and the other one result in an unstable controller. The control system is linear,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -2 & -2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ -1 \end{bmatrix} u \quad (2.6)$$

It is a controllable system. The desired path is the line $x_2 = -1$, i.e.

$$x_d(s) = \begin{bmatrix} s \\ -1 \end{bmatrix}. \quad (2.7)$$

Suppose that the velocity on the desired path is 1, so $s = t$. A time dependent feedback is

$$u = 2t - 1. \quad (2.8)$$

It is easy to check that the error $x(t) - x_d(t)$ satisfies

$$\dot{e} = \begin{bmatrix} -2 & -2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} e. \quad (2.9)$$

It is asymptotically stable. The simplest reference projection for the desired path is the orthogonal projection

$$\gamma(x) = x_1.$$

Therefore, the virtual time is $T = x_1$. The control system under non-time based control is

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} -2 & -2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ -1 \end{bmatrix} (2x_1 - 1) \\ &= \begin{bmatrix} -2x_2 - 1 \\ x_2 + 1 \end{bmatrix}. \end{aligned} \quad (2.10)$$

The second equation $\dot{x}_1 = x_2 + 1$ is unstable. So, the system does not approach the desired path under the non-time based controller, although the non-time based feedback is converted from an exponentially stable feedback (2.8).

On the other hand, if the time-variant feedback is

$$u = 2t - 1 + \frac{3}{2}(x_1 - t) + \frac{3}{2}(x_2 + 1).$$

It is easy to check that the time dependent feedback drives the system approaching $x_d(t)$. Furthermore, the non-time based feedback is

$$u(x) = 2x_1 - 1 + \frac{3}{2}(x_2 + 1).$$

The system under the non-time based control is

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}x_2 + \frac{1}{2} \\ -\frac{1}{2}(x_2 + 1) \end{bmatrix} \quad (2.11)$$

The second equation guarantees that $\lim x_2(t) = -1$. So, the non-time based control drives the system asymptotically approaching x_d .

The example implies that not all the stable feedback $u(x, t)$ can be converted into stable non-time based controllers. In the following, we derive conditions under which

the non-time based controller is guaranteed to be stable.

2.3.2 Error dynamics and stability

Two error dynamics are derived in this section for the stability analysis. The first error dynamics is under the time-variant feedback $u(x, t)$. The error of the system

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

is defined by

$$e = x(t) - x_d(t). \quad (2.12)$$

It satisfies

$$\begin{aligned} \dot{e} &= f(x, u(x, t)) - f(x_d, u(x_d, t)) \\ &= f(e + x_d, u(e + x_d, t)) - f(e + x_d, u(e + x_d, t)). \end{aligned} \quad (2.13)$$

The desired trajectory $x_d(t)$ can be considered as a function of time t . The error dynamics is a time-variant system. Define

$$E(e, t) = f(e + x_d, u(e + x_d, t)) - f(e + x_d, u(e + x_d, t)). \quad (2.14)$$

The error dynamics under $u(x, t)$ is

$$\dot{e} = E(e, t). \quad (2.15)$$

Under the non-time based feedback $u(x, T(x))$, the error is defined by

$$\bar{e} = x(t) - x_d(T). \quad (2.16)$$

The system is not tracking the trajectory as a function of real time. The controller suppose to drive the system toward the desired state at the virtual time. The error dynamics is changed due to the derivative of $T(x)$.

$$\begin{aligned}\dot{\bar{e}} &= f(x, u(x, T)) - f(x_d(T), u(x_d(T), T)) \frac{\partial T}{\partial x} f(x, u(x, T)) \\ &= f(x, u(x, T)) - f(x_d(T), u(x_d(T), T)) - f(x_d(T), u(x_d(T), T)) \left(\frac{\partial T}{\partial x} f(x, u(x, T)) - 1 \right)\end{aligned}\tag{2.17}$$

From (2.14) and the relation

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

we have

$$\dot{\bar{e}} = E(\bar{e}, T) - f(x_d(T), u(x_d(T), T))(\dot{T}(x) - 1).\tag{2.18}$$

$$\dot{x} = f(x, u(x, T(x)))\tag{2.19}$$

System (2.19) is treated as part of the error dynamics. System (2.18) is not a complete system because $T(x)$ and \dot{T} depend on (2.19), which is the vehicle dynamics under non-time based feedback.

The two errors e and \bar{e} defined by (2.12) and (2.16) follows different dynamics. The stability of e does not automatically imply the stability of \bar{e} . If we treat \dot{T} as an output of (2.19), then its performance is critical for the stability of (2.18). In this dissertation, a δ neighborhood of $x_d(s)$ is the set of all x such that the distance from the point x to the curve x_d is less than δ . In this dissertation, we assume

Assumption A1: The vector fields and functions $f(x, u(x, t))$, $u(x, t)$, $\frac{\partial f}{\partial x}$, $\frac{\partial f}{\partial u}$, $\frac{\partial u}{\partial x}$,

$\frac{\partial u}{\partial t}, \frac{\partial T}{\partial x}$ are all bounded for $t > 0$ and x in a δ neighborhood of x_d .

The nonlinear function $f(x, t)$ represent the rate of change of the state variable. It is reasonable to require in **A1** that the speed, acceleration of the state variables of a mechanical system is bounded near the desired path. **A1** also assumes that $u(x, t)$ and its derivative are bound. It implies that the controller can react to the change of state and time variable at a limited rate.

Theorem 2.3.1 *Suppose (2.15) is uniformly exponentially stable in $\|e\| < \delta$. Suppose the system (2.19) has solution for $t > 0$ and the solution satisfies*

$$\lim_{t \rightarrow \infty} \dot{T}(x(t)) - 1 = 0$$

uniformly and exponentially if $\|\bar{e}(0)\| < \delta$. Then, there is a $\delta_1 > 0$ so that

$$\lim_{t \rightarrow \infty} \bar{e}(t) = 0$$

provided $\|\bar{e}(0)\| < \delta_1$.

Proof. From the converse theorem of stability, there exists a Lyapunov function $V(e, t)$

$$\begin{aligned} \alpha_1 \|e\|^2 &\leq V(e, t) \leq \alpha_2 \|e\|^2, \quad t \geq 0, \\ \left| \frac{\partial V}{\partial e}(e, t) \right| &\leq \alpha_3 \|e\|, \quad t \geq 0, \\ \frac{\partial V}{\partial t} + \frac{\partial V}{\partial e} E(e, t) &= -\|e\|^2, \quad t \geq 0. \end{aligned} \tag{2.20}$$

in a neighborhood $|e| < \delta$. Under the non-time based controller, the error is replaced by \bar{e} in (2.16). The Lyapunov function for \bar{e} is $V(\bar{e}, T(x(t)))$. Its derivative in the

direction of (2.18)-(2.19) is

$$\begin{aligned}
\dot{V}(\bar{e}, T) &= \frac{\partial V}{\partial T} \dot{T} + \frac{\partial V}{\partial \bar{e}} \left(E(\bar{e}, T) - f(x_d, u(x_d, T))(\dot{T} - 1) \right) \\
&= \frac{\partial V}{\partial T} + \frac{\partial V}{\partial \bar{e}} E(\bar{e}, T) + \left(\frac{\partial V}{\partial T} - \frac{\partial V}{\partial \bar{e}} f(x_d, u(x_d, T)) \right) (\dot{T} - 1) \\
&= -\|\bar{e}\|^2 + \left(\frac{\partial V}{\partial T} - \frac{\partial V}{\partial \bar{e}} f(x_d, u(x_d, T)) \right) (\dot{T} - 1)
\end{aligned} \tag{2.21}$$

From **A1**, we know that $E(e, t)$ is bounded in $\|e\| < \delta$. So, the third equation in (2.20) implies that $\frac{\partial V}{\partial t}(e, t)$ is bounded in $0 \leq t \leq \infty$ and $\|e\| \leq \delta$. From the second equation of (2.20), it is known that $\frac{\partial V}{\partial x}(e, t)$ is also bounded. Therefore,

$$\frac{\partial V}{\partial T}(\bar{e}, T) - \frac{\partial V}{\partial \bar{e}}(\bar{e}, T) f(x_d, u(x_d, T)) \tag{2.22}$$

is bounded in $0 \leq t \leq \infty$ and $\|\bar{e}\| \leq \delta$. Since $\dot{T}(x) - 1$ is uniformly exponentially convergent to zero, we have

$$\left(\frac{\partial V}{\partial T} - \frac{\partial V}{\partial \bar{e}} f(x_d, u(x_d, T)) \right) (\dot{T} - 1) < M e^{-\lambda t} \tag{2.23}$$

for some $M > 0$ and $\lambda > 0$, provided $\|\bar{e}\| < \delta$. Equations (2.21) and (2.23) implies

$$\begin{aligned}
\dot{V}(\bar{e}, T) &< -\|\bar{e}\|^2 + M e^{-\lambda t} \\
&< -\frac{1}{\alpha_2} V(\bar{e}, T) + M e^{-\lambda t}, \quad \text{for } \|\bar{e}\| < \delta.
\end{aligned} \tag{2.24}$$

At any fixed time t_0 , if $\bar{e}(t_0) < \delta$, then

$$\dot{V}(\bar{e}, T) < -\frac{1}{\alpha_2} V(\bar{e}, T) + M e^{-\lambda t_0} e^{-\lambda(t-t_0)}, \quad \text{for } t > t_0 \tag{2.25}$$

This implies

$$\frac{d}{dt} \left(e^{\frac{1}{\alpha_2}(t-t_0)} V(\bar{e}, T) \right) < M e^{-\lambda t_0} e^{(-\lambda + \frac{1}{\alpha_2})(t-t_0)}. \quad (2.26)$$

The integration of the inequality over the time interval $[t_0, t]$ yields

$$e^{\frac{1}{\alpha_2}(t-t_0)} V(\bar{e}, T) - V(\bar{e}(t_0), T(x(t_0))) \leq M e^{-\lambda t_0} \frac{1}{-\lambda + \alpha_1^{-1}} \left(e^{(-\lambda + \frac{1}{\alpha_2})(t-t_0)} - 1 \right) \quad (2.27)$$

It is equivalent to

$$V(\bar{e}, T) \leq e^{-\frac{1}{\alpha_2}(t-t_0)} V(\bar{e}(t_0), T(x(t_0))) + \frac{M e^{-\lambda t_0}}{-\lambda + \alpha_1^{-1}} \left(e^{-\lambda(t-t_0)} + e^{-\frac{1}{\alpha_2}(t-t_0)} \right), \quad (2.28)$$

provided $t \geq t_0$ and $\|\bar{e}(t)\| < \delta$. There exists $t_0 > 0$ so that

$$\frac{M e^{-\lambda t_0}}{-\lambda + \alpha_1^{-1}} \left(e^{-\lambda(t-t_0)} + e^{-\frac{1}{\alpha_2}(t-t_0)} \right) < \frac{\alpha_1}{2} \delta^2, \quad (2.29)$$

for $t \geq t_0$ and $\|\bar{e}\| < \delta$. If $\|\bar{e}(t_0)\| < \frac{\alpha_1}{2\alpha_2} \delta^2$, (2.20) and (2.28) imply

$$\begin{aligned} \alpha_1 \|\bar{e}\|^2 &\leq V(\bar{e}, T()) \\ &\leq V(\bar{e}(t_0), T(x(t_0))) + \frac{\alpha_1}{2} \delta^2 \\ &\leq \alpha_2 \|\bar{e}(t_0)\|^2 + \frac{\alpha_1}{2} \delta^2 \\ &\leq \alpha_2 \frac{\alpha_1}{2\alpha_2} \delta^2 + \frac{\alpha_1}{2} \delta^2 \\ &\leq \alpha_1 \delta^2. \end{aligned} \quad (2.30)$$

Therefore, $\|\bar{e}(t)\| \leq \delta$ for all $t \geq t_0$, provided

$$\|\bar{e}(t_0)\| < \frac{\alpha_1}{2\alpha_2} \delta^2. \quad (2.31)$$

Thus, (2.28) holds for $t \geq t_0$ if the initial error satisfies (2.31). From (2.28), it is easy to derive that $\lim_{t \rightarrow \infty} V(\bar{e}, T) = 0$, provided (2.31) holds. Therefore, $\lim_{t \rightarrow \infty} \bar{e}(t) = 0$. Because the right side of (2.18) is uniformly bounded in $\|\bar{e}\| < \delta$, there exists $\delta_1 > 0$ so that $\|\bar{e}(0)\| < \delta_1$ implies (2.31). This proves that

$$\lim_{t \rightarrow \infty} \bar{e}(t) = 0$$

if $\|\bar{e}(0)\| < \delta_1$. □

2.3.3 Feedback design

Given a time-variant tracking feedback $u(x, t)$ and a reference projection, one needs to check the performance of the following system with dummy output $y^* = \dot{T}(x) - 1$,

$$\begin{aligned} \dot{x} &= f(x, u(x, T(x))) \\ y^* &= \frac{\partial T}{\partial x} f(x, u(x, T(x))) - 1 \end{aligned} \tag{2.32}$$

If the dummy output y^* exponentially converges to zero, the non-time based control drives the vehicle approaching the desired path. In this section, a design method is introduced if $T(x)$ has relative degree. The feedback guarantees that y^* is exponentially stable. In the following, we assume that the desired path satisfies

$$\begin{aligned} y_{d1} &= s \\ y_{d2} &= y_{d2}(s) \\ &\vdots \\ y_{dk} &= y_{dk}(s). \end{aligned} \tag{2.33}$$

Or equivalently, the first output variable y_{d1} serves as the action reference. This is true for many applications as shown in the next section. A natural reference projection is

$$\gamma(x) = y_1(x) \quad (2.34)$$

where y_1 is the first output of the original system (2.1). We also assume that in a neighborhood of the desired path, reference projection $\gamma(x) = y_1$ has relative degree, i.e. $\gamma(x)$ satisfies

$$\begin{aligned} \frac{\partial}{\partial u} L_{f(x,u)}^{i-1}(\gamma(x)) &= 0, \quad 1 \leq i \leq r, \\ \frac{\partial}{\partial u_1} L_{f(x,u)}^r(\gamma(x)) &\neq 0 \end{aligned} \quad (2.35)$$

where $L_{f(x,u)}$ is the Lie operator defined by

$$L_{f(x,u)}(\gamma(x)) = \frac{\partial \gamma}{\partial x} f(x, u).$$

The following design method follows the idea of feedback linearization (see, for instance, [30]).

If the desired speed on the desired path is defined by $s = v(t)$, then the virtual time is $T = v^{-1}(\gamma(x))$. The feedback is defined based on the virtual time $T(x)$. It is easy to check that $T(x)$ also has relative degree. Define $\xi_i = L_{f(x,u)}^{i-1}(T(x))$ for $1 \leq i \leq r$. The equation (2.35) implies

$$\begin{aligned} \dot{\xi}_1 &= \xi_2 \\ \dot{\xi}_2 &= \xi_3 \\ &\vdots \\ \dot{\xi}_r &= b(x) + a_1(x)u_1 + a_2(x)u_2 + \cdots + a_m(x)u_m, \quad a_1(x) \neq 0. \end{aligned} \quad (2.36)$$

The feedback for u_1 has the form

$$\begin{aligned} u_1(x, t, u_2, \dots, u_m) = & -\frac{1}{a_1(x)}(b(x) + a_1(x)u_1 + a_2(x)u_2 + \dots + a_m(x)u_m) \\ & + k_1(\xi_1 - t) + k_2(\xi_2 - 1) + k_3\xi_3 + \dots + k_r\xi_r. \end{aligned} \quad (2.37)$$

The gain $\begin{bmatrix} k_1 & k_2 & \dots & k_r \end{bmatrix}$ satisfies that

$$s^r - k_r s^{r-1} - k_{r-1} s^{r-2} - \dots - k_1 \text{ is Hurwitz,} \quad (2.38)$$

$$s^{r-1} - k_r s^{r-2} - k_{r-1} s^{r-3} - \dots - k_2 \text{ is also Hurwitz if } r > 1. \quad (2.39)$$

In this system, the dummy output $y^* = \dot{T} - 1$ is completely determined by (2.36)-(2.37), which has the following form

$$\begin{aligned} \dot{\xi}_1 &= \xi_2 \\ \dot{\xi}_2 &= \xi_3 \\ &\vdots \\ \dot{\xi}_r &= k_1(\xi_1 - t) + k_2(\xi_2 - 1) + k_3\xi_3 + \dots + k_r\xi_r \end{aligned} \quad (2.40)$$

Because (2.38) is Hurwitz, the control guarantees that both $\xi_1 - t = T(x) - t$ and $\xi_2 - 1 = \dot{T} - 1$ exponentially approach zero. If t is replaced by T , the variable $\xi_1 - t$ is constant zero. The closed-loop system (2.40) becomes

$$\begin{aligned} \dot{\xi}_1 &= \xi_2 \\ \dot{\xi}_2 &= \xi_3 \\ &\vdots \\ \dot{\xi}_r &= k_2(\xi_2 - 1) + k_3\xi_3 + \dots + k_r\xi_r \end{aligned}$$

Because (2.39) is Hurwitz, $\xi_2 - 1 = \dot{T} - 1$ exponentially approach zero under $u_1 = u_1(x, T, u_2, \dots, u_m)$. Furthermore, the tracking error $y_1 - y_{d1}(T)$ satisfies

$$\begin{aligned} y_1 - y_{d1}(T) &= \gamma(x) - v^{-1}(T) \\ &= v^{-1}(T) - v^{-1}(T) \\ &= 0 \end{aligned}$$

Therefore, the tracking of y_1 is guaranteed. Applying (2.37) to the original system, we have

$$\dot{x} = f(x, u_1(x, t, u_2, \dots, u_m), u_2, \dots, u_m). \quad (2.41)$$

For this system, the control inputs are u_2, u_3, \dots, u_m and the desired path to track is $\{y_{d2}(s), y_{d3}(s), \dots, y_{dk}(s)\}$. The feedback for $u_i, i \geq 2$, can be designed using any tracking control design algorithms. Then, replace the time variable t in the feedback to obtain a non-time based controller. The stability is guaranteed because (2.37) drives $\dot{T} - 1$ approaching zero.

To summarize, we assume that $y_{d1}(s) = s$ is the action reference. The first output $y_1(x)$ is defined to be the reference projection $\gamma(x)$. Assume that $\gamma(x)$ has relative degree, then feedback u_1 is defined by (2.37) satisfying (2.38)-(2.39). The other feedbacks are designed to tracking y_i , for $i \geq 2$. The final non-time based feedback is to replace t by $T(x)$ in all the feedbacks. It is guaranteed that the non-time based feedback drives the tracking error to zero.

CHAPTER 3

TRACKING CONTROL OF THE MOBILE PLATFORM

3.1 An Introduction to the Control of Mobile Platforms

This chapter presents a novel method of non-time based controller design for path tracking of moving platforms. The purpose of this proposed controller is to design an event based controller which can coordinate the motion of mobile platform and manipulator arm. In this chapter, the state variables of the mobile platform are adopted as the motion reference. In the control of mobile manipulators, a global state variable is used as the motion reference of both the mobile platform and the manipulator arm.

The key to the proposed control method is the introduction of a suitable motion reference other than time, which is directly related to the desired path and the measurable system output. It enables the construction of control systems with integrated planning capability, in which planning becomes a real-time closed-loop process. As a result, the control system is capable of coping with unexpected events and uncertainties. This new design method converts a controller designed with traditional time-based approaches to a non-time based controller using a reference projection. It significantly simplifies the design procedure for a non-time based controller. The method is exemplified by the design and experiment for the tracking control of car-like mobile robots. Based on the properties of the desired path, two car-like mobile robot controllers are proposed. According to the type of the desired trajectory, the time based controller is mapped to a non-time based controller by choosing appropriate motion reference. The controllers are implemented and tested for different trajectories such as sine waves, circles and ellipses. Lab experiment on mobile platform are carried out in an environment with unexpected obstacles and unexpected stop for at arbitrary time.

3.2 Path Tracking Control for Mobile Robots

3.2.1 Mathematical Model of the Robot

In this section, the general design algorithm is applied to the model of mobile robot shown in Figure 3.1. The rear wheels are aligned with the vehicle while the front wheels are allowed to spin about the vertical axes. The robot is a rear wheel driven robot. The system is constrained by allowing the wheels to roll and spin, but not slip. Let (x, y, θ, ϕ) denote the configuration of the robot, parameterized by the location of the rear wheels. The dynamical model of the mobile robot can be represented as [45]:

$$\begin{aligned}\dot{x} &= u_1 \cos \theta \\ \dot{y} &= u_1 \sin \theta \\ \dot{\theta} &= \frac{u_1}{l} \tan \phi \\ \dot{\phi} &= u_2,\end{aligned}\tag{3.1}$$

where u_1 corresponds to the forward velocity of the rear wheels of the robot and u_2 corresponds to the angular velocity of the steering wheels, the angle of the robot body with respect to the horizontal axes is θ , the steering angle with respect to the robot body is ϕ , (x, y) is the location of the rear wheels, l is the length between the front and the rear wheels.

Let $X_d = [x_d(t), y_d(t), \theta_d(t), \phi_d(t)]$ denote the desired states of the mobile robot at time instance t , the purpose of the path tracking control is to design a controller $u_1 \doteq u_1(X_d, t), u_2 = u_2(X_d, t)$ such that the robot will asymptotically track the desired path $y_d = f(x_d)$. By using the reference projection method, the time based feedback laws can be converted into the time invariant feedback $\tilde{u}_1(X_d, T(x))$ and $\tilde{u}_2(X_d, T(x))$. The virtual time $T(x)$ is designed by the non-time based motion reference s .

The path control of a nonholonomic robot has been discussed in the literature.

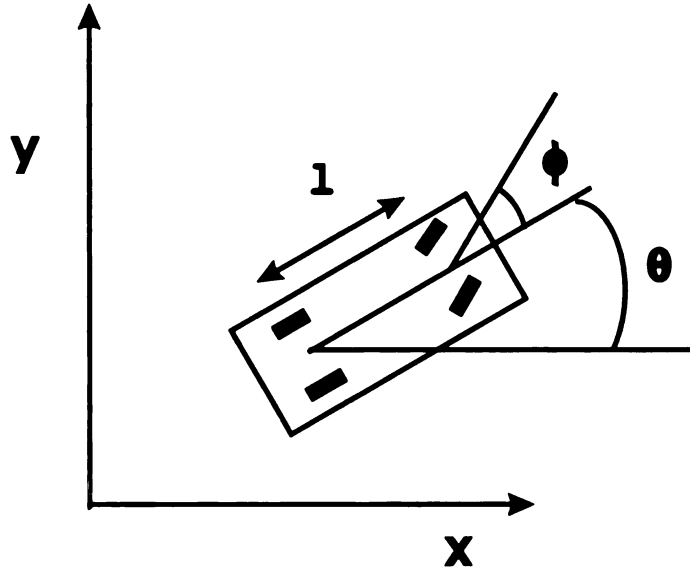


Figure 3.1: The configuration of a mobile robot.

The system model can be linearized based on certain path and a feedback control was designed using the LQR approach[22]. The disadvantage for this class of methods lie in that the controller is path specified. The controller is dependent on the characteristics of the path. Exact linearization approach, such as the chained form, has been developed and used in the open loop path planning [45] and feedback control design [21, 60]. However, singular points exist in the derivation of chained form, the feedback controller based on the chained form therefore cannot track certain path. In this section, two classes of paths are discussed. One is curve without U-turns, an exact linearization and feedback control design similar to chained form is described. For a curve with U turns, an exact linearization is presented and feedback controller is designed. Both the time based controllers are converted to non-time based controllers using the reference projection approach.

3.2.2 Tracking Curves with no U-turn

Suppose that the desired path is the following function of time

$$x = x_d(t), \quad y = y_d(t).$$

In this section, a desired path without U-turn is assumed, i.e. for any value x_d , there is at most one value y_d such that (x_d, y_d) is on the desired path. This implies that $\dot{x}_d(s) > 0$. Let's first find a feedback for u_1 . Since

$$\frac{d}{dt}(x - x_d) = u_1 \cos \theta - \dot{x}_d,$$

the control input u_1 can be defined by

$$u_1 = \frac{1}{\cos \theta}(\dot{x}_d - (x - x_d)). \quad (3.2)$$

Under (3.2), the first equation of (3.1) becomes

$$\frac{d}{dt}(x - x_d) = -(x - x_d).$$

Its solution asymptotically approaches to zero as $t \rightarrow \infty$.

The following change of coordinates is defined to achieve the linearization of the

last three equations in (3.1),

$$\begin{aligned}
z_1 &= y \\
z_2 &= \sin \theta \\
z_3 &= \frac{1}{l} \cos \theta \tan \phi \\
\alpha_1 &= -\frac{u_1}{l^2} \sin \theta (\tan \phi)^2 \\
\beta_1 &= \frac{\cos \theta}{l \cos^2 \phi} \\
v &= \alpha_1 + \beta_1 u_2
\end{aligned} \tag{3.3}$$

The last three equations in (3.1) are transformed into

$$\begin{aligned}
\dot{z}_1 &= u_1 z_2 \\
\dot{z}_2 &= u_1 z_3 \\
\dot{z}_3 &= v
\end{aligned} \tag{3.4}$$

Define

$$\begin{aligned}
e_1 &= z_1 - y_d \\
e_2 &= z_2 - \frac{\dot{y}_d}{u_1} \\
e_3 &= z_3 - \frac{1}{u_1} \frac{d}{dt} \left(\frac{\dot{y}_d}{u_1} \right)
\end{aligned} \tag{3.5}$$

the error dynamics of the last three equations is represented by

$$\begin{aligned}
\dot{e}_1 &= u_1 e_2 \\
\dot{e}_2 &= u_1 e_3 \\
\dot{e}_3 &= v - \frac{d}{dt} \left(\frac{1}{u_1} \frac{d}{dt} \left(\frac{\dot{y}_d}{u_1} \right) \right)
\end{aligned} \tag{3.6}$$

The right side of the last equation in (3.6) satisfies

$$\frac{d}{dt} \left(\frac{1}{u_1} \frac{d}{dt} \left(\frac{\dot{y}_d}{u_1} \right) \right) = \frac{\ddot{\ddot{y}}_d}{u_1^2} - \frac{3\ddot{y}_d \dot{u}_1}{u_1^3} + \frac{3\dot{y}_d \dot{u}_1^2}{u_1^4} - \frac{\dot{y}_d \ddot{u}_1}{u_1^3} \tag{3.7}$$

Notice that \ddot{u}_1 contains the term $\dot{\phi}$, which depends on u_2 . Therefore, the expression (3.7) has the form

$$\alpha_2(x, y, \theta, \phi, t) + \beta_2(x, y, \theta, \phi, t)u_2 \quad (3.8)$$

To stabilize (3.6), v is defined by

$$\begin{aligned} v = & \alpha_2(x, y, \theta, \phi, t) + \beta_2(x, y, \theta, \phi, t)u_2 \\ & + u_1(a_1e_1 + a_2e_2 + a_3e_3) \end{aligned} \quad (3.9)$$

Then, (3.6) is equivalent to

$$\begin{aligned} \dot{e}_1 &= u_1e_2 \\ \dot{e}_2 &= u_1e_3 \\ \dot{e}_3 &= u_1(a_1e_1 + a_2e_2 + a_3e_3) \end{aligned} \quad (3.10)$$

Select the values of $a_i, i = 1, 2, 3$, so that all the eigenvalues of

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ a_1 & a_2 & a_3 \end{bmatrix}$$

are on the left half plane. Since $\dot{x}_d(s) > 0$, $u_1(t) > 0$ in a neighborhood of $(x_d(t), y_d(t))$, $\int_0^t u_1(\tau)d\tau$ increases. The solution of (3.10) is

$$e(t) = e_0 \exp(A \int_0^t u_1(\tau)d\tau).$$

Since A is stable, $\lim_{t \rightarrow \infty} e(t) = 0$ as $\int_0^t u_1(\tau)d\tau$ increases. Therefore, $y - y_d(t) \rightarrow 0$ when $t \rightarrow \infty$. By solving for u_2 from (3.9) and the last equation in (3.3), u_2 can be found.

The feedback is

$$u_2 = \frac{\alpha_2 - \alpha_1 + u_1(a_1e_1 + a_2e_2 + a_3e_3)}{\beta_1 - \beta_2} \quad (3.11)$$

In summary, suppose the desired path is

$$x_d(v(t)) = t, \quad y_d(v(t)) = \sin t \quad (3.12)$$

The non-time based control u_1 and u_2 can be computed as following

$$\begin{aligned} u_1 &= \frac{\dot{x}_d - (x - x_d)}{\cos \theta}, \\ u_2 &= \frac{\alpha_2 - \alpha_1}{\beta_1 - \beta_2} + \frac{u_1}{\beta_1 - \beta_2}(a_1e_1 + a_2e_2 + a_3e_3) \end{aligned} \quad (3.13)$$

The coefficients a_1 , a_2 and a_3 are obtained from the linear control design.

Remark. A circle or an ellipse does not satisfy the condition $\dot{x}_d > 0$. $\theta = \pm \frac{\pi}{2}$ are the singular points of the system. The controller (3.11) does not apply. This case is discussed in the next subsection. \triangleleft

The feedback (3.2,3.11) depends on time. However, it can be converted into a time-invariant feedback using reference projection as proved in § 2.2. Using the method developed in § 2.2, the feedback (3.2,3.11) is changed to time-invariant feedbacks for the tracking of the curve $y = \sin x$. The key step is to find a motion reference for the task. The design method proposed in this dissertation is not based on a specific motion reference. It works for arbitrary choice of the reference. Different from circles and lines, arc length and orthogonal projection are not the best choice if the desired path is $y = \sin x$, because the computation of arc length for this curve is complicated. In this case, it will be shown in § 3.3 that $s = x$ is a much simpler choice of the action reference. The lab experiment based on this controller is shown in § 3.3.

3.2.3 Tracking Circles and Ellipses

As remarked in the previous section, the controller (3.13) fails to track trajectories having U-turns. For a class of trajectories such as circles and ellipses, another controller is proposed in this dissertation for the mobile robot. Based on the same state equation (3.1) and the desired trajectory $(x_d(t), y_d(t))$, the system is first transformed into a polar form. A controller is then derived based on the polar axis.

By defining

$$\begin{aligned} r &= \sqrt{x^2 + y^2} \\ \psi &= \tan^{-1} \frac{y}{x} \end{aligned} \tag{3.14}$$

which is equivalent to

$$\begin{aligned} x &= r \cos \psi \\ y &= r \sin \psi \end{aligned} \tag{3.15}$$

the system dynamic model in polar axis is

$$\begin{aligned} \dot{r} &= \frac{1}{r}(x\dot{x} + y\dot{y}) \\ &= \frac{1}{r}(r \cos \psi u_1 \cos \theta + r \sin \psi u_1 \sin \theta) \\ &= u_1 \cos(\theta - \psi) \end{aligned} \tag{3.16}$$

$$\begin{aligned} \dot{\psi} &= \frac{1}{1 + (y/x)^2} \frac{\dot{y}x - x\dot{y}}{x^2} \\ &= \frac{1}{r^2}(u_1 \sin \theta r \cos \psi - u_1 \cos \theta r \sin \psi) \\ &= \frac{u_1}{r} \sin(\theta - \psi) \end{aligned} \tag{3.17}$$

System model (3.1) is then transformed into

$$\begin{aligned}
\dot{\psi} &= \frac{u_1}{r} \sin(\theta - \psi) \\
\dot{r} &= u_1 \cos(\theta - \psi) \\
\dot{\theta} &= \frac{u_1}{l} \tan \phi \\
\dot{\phi} &= u_2
\end{aligned} \tag{3.18}$$

The desired trajectory in Cartesian space is also transformed into polar axis and denoted by $r_d, \psi_d, \theta_d, \phi_d$ respectively. Considering only the first equation of (3.18), u_1 is designed to stabilize ψ . By defining

$$u_1 = \frac{r}{\sin(\theta - \psi)} (\dot{\psi}_d - k_1(\psi - \psi_d)), \tag{3.19}$$

where k_1 is greater than zero, the first equation (3.1) becomes

$$\frac{d}{dt}(\psi - \psi_d) = -k_1(\psi - \psi_d), \tag{3.20}$$

which is asymptotically stable.

Substituting (3.19) into the second equation of (3.18) and defining $e_r = r - r_d$, it becomes

$$\dot{r} = r \operatorname{ctan}(\theta - \psi) (\dot{\psi}_d - k_1(\psi - \psi_d)). \tag{3.21}$$

Taking $\operatorname{ctan}(\theta - \psi)$ as the control input of this equation, a feedback controller which stabilize (3.21) could be

$$\operatorname{ctan}(\theta - \psi) = \frac{1}{r(\dot{\psi}_d - k_1(\psi - \psi_d))} (\dot{r}_d - k_2 e_r). \tag{3.22}$$

where $k_2 > 0$. Therefore,

$$\begin{aligned}\theta &= \psi + \arctan\left(\frac{1}{r(\dot{\psi}_d - k_1(\psi - \psi_d))}(\dot{r}_d - k_2 e_r)\right) \\ &= g_1(r, \psi, t).\end{aligned}\tag{3.23}$$

For the third equation of (3.18), defining $e_\theta = \theta - g_1$, the error dynamics of θ can be derived,

$$\begin{aligned}\dot{\theta} &= \frac{u_1}{l} \tan \phi \\ \dot{e}_\theta &= \frac{u_1}{l} \tan \phi - \dot{g}_1.\end{aligned}\tag{3.24}$$

Taking ϕ as the control input, the feedback controller for (3.24) is

$$\begin{aligned}\phi &= \arctan\left(\frac{l}{u_1}(\dot{g}_1 - k_3 e_\theta)\right) \\ &= \arctan\left(\frac{l \sin(\theta - \psi)}{r(\dot{\psi}_d - k_1(\psi - \psi_d))}(\dot{g}_1 - k_3 e_\theta)\right) \\ &= g_2(r, \theta, \psi, t).\end{aligned}\tag{3.25}$$

For the fourth equation, let $e_\phi = \phi - g_2$,

$$\dot{e}_\phi = u_2 - \dot{g}_2\tag{3.26}$$

the controller of the system is finally designed by

$$u_2 = \dot{g}_2 - k_4 e_\phi\tag{3.27}$$

To summarize, the time depended feedbacks of system (3.18) are

$$\begin{aligned}
u_1 &= \frac{r}{\sin(\theta - \psi)}(\dot{\psi}_d - k_1(\psi - \psi_d)) \\
u_2 &= \dot{g}_2 - k_4 e_\phi \\
g_2 &= \arctan\left(\frac{\sin(\theta - \psi)}{r(\dot{\psi}_d - k_1(\psi - \psi_d))}(\dot{g}_1 - k_3 e_\theta)\right) \\
g_1 &= \psi + \arctan\left(\frac{1}{r(\dot{\psi}_d - k_1(\psi - \psi_d))}(\dot{r}_d - k_2 e_r)\right),
\end{aligned} \tag{3.28}$$

where $k_i, i = 1, 2, 3, 4$ are greater than zero.

The feedbacks in (3.28) are also time dependent controllers. However, it can be converted into a time-invariant feedback using reference projection as proved in § 2.2. Since the design method proposed in this dissertation is not based on a specific motion reference, different references according to different curves are used. As it is shown in § 3.3, a combination of the state variables can be used for convenience.

3.3 Experiments and Results

The non-time based controllers have been implemented on a Nomadic XR4000 mobile robot. The mobile robot has four wheels. Each wheel can be individually steered. However, in order to implement the above control scheme, only two front wheels are steered in the experiments. The other two wheels maintain Straightforward configuration. The distance between front and rear wheels is 0.3053 *m*. In the following subsections, three experimental results are given. In § 3.3.1, the results of controller (3.13) are shown. In § 3.3.2 and § 3.3.3, the results of controller (3.28) are reported.

3.3.1 Tracking Sinusoid Waves

The desired trajectory is given by:

$$y_d = \sin(s) \quad x_d = s,$$

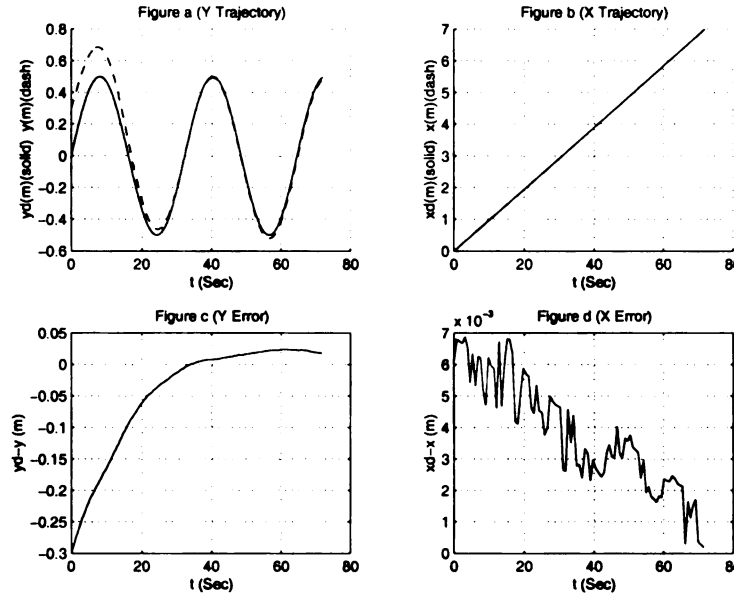


Figure 3.2: Tracking a sinusoid curve with initial error.

where s is the motion reference. The velocity along the desired path is defined by $s = vt$, where v is a constant. Notice that the desired velocity is a constant in the direction x . But the desired velocity on the sinusoid wave is not a constant. This change of desired velocity can be tracked by a non-time based controller using the reference projection method from § 2.2.

In the controller, the closed loop poles are assigned to be $-3, -5 + i, -5 - i$. The reference projection is $s = x$, and the synthetic time is $t = \frac{1}{v}x$. It is easy to check that the relative degree of the reference projection is 1. Therefore, the feedback in § 3.2.2 satisfies the design requirements of § 2.3.3. It is converted to a non-time based controller and then applied the Normadic XR4000 mobile robot. If the initial position of the robot is not on the desired path as shown in Figure 3.2, the controller is able to command the robot to approach the desired path. There exists an initial error of $0.3m$ in the experiment. But the tracking error converges to zero after the robot starts moving. This experiment demonstrates the ability of the controller to rapidly overcome initial tracking error. It is worthy noting that the XR4000 is a holonomic robot, and it is simulated as nonholonomic robot in the experiment.

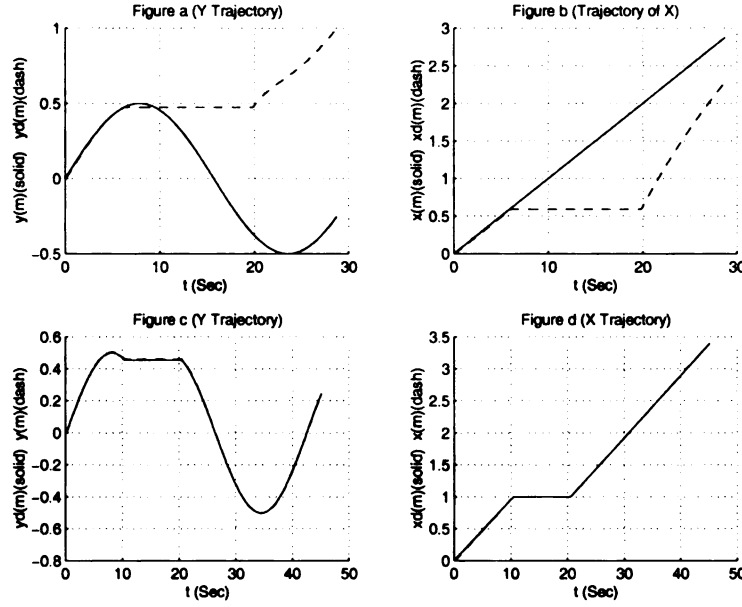


Figure 3.3: Time based control vs Non-time based control.

The interesting result of the experiment is presented in Figure 3.3. While tracking a pre-planned desired path, the robot was stopped by an unexpected obstacle at the 10th sec. Figure 3.3a and 3.3b show the results of a time-based controller. Its desired input was a function of the time. Therefore the desired position would keep changing even the robot was blocked. As a result, the tracking error kept increasing until the system became unstable. However, for the non-time based controller, the results are completely different. As shown in Figure 3.3c and 3.3d, when the robot is blocked, the action reference remains constant because the reference projection depends on the position of the robot. Therefore, the tracking error remains constant. Furthermore, after the obstacle is removed at the 20th sec, the robot automatically resumes the original motion without re-planning. This experiment demonstrates the capability of the non-time based controller for coping with unexpected events. It should be noticed that the non-time based controller is not simply replacing the time by a non-time based motion reference. The new motion reference is determined by sensory measurements. It reflects the key information of the state of the system. The

reference projection acting as an on-line real-time planner.

3.3.2 Tracking Circles

This experiment is done under the control of (3.28) with feedback gains $k_1 = 1, k_2 = 0.4, k_3 = 8, k_4 = 3$. The desired path is

$$\begin{aligned} x_d^2 + y_d^2 &= a^2 \\ y_d &= a \sin(s), x_d = a \cos(s) \end{aligned}$$

where s is the non-time based reference. It represents the angle corresponding to the tracked arc of the robot. The reference is computed directly by the location of the robot, (x, y) . In the polar coordinates model of the vehicle, the reference projection has relative degree 1. Given a constant relation between the reference s and the time t , $s = vt$, the synthetic time is $t = \frac{1}{a} \text{atan}_2(y, x)$, where $\text{atan}_2(\cdot, \cdot)$ maps (y, x) to $(0, 2\pi)$. The radius of the circle in the experiment is $0.5m$. Figure 3.4 and 3.5 recorded the experimental results of tracking a circle by the controller (3.28). From Figure 3.4a-3.4d and Figure 3.5a-3.5d, it can be seen that the static tracking error is less than $1mm$.

3.3.3 Tracking Ellipses

This experiment is also done under (3.28). To track an ellipse, the desired path is given by

$$\frac{x_d^2}{a^2} + \frac{y_d^2}{b^2} = 1$$

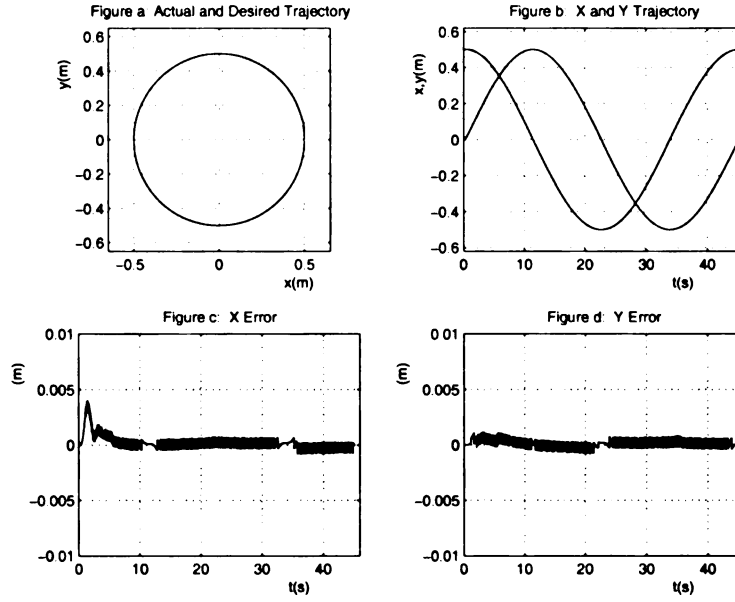


Figure 3.4: Tracking a circle without initial error

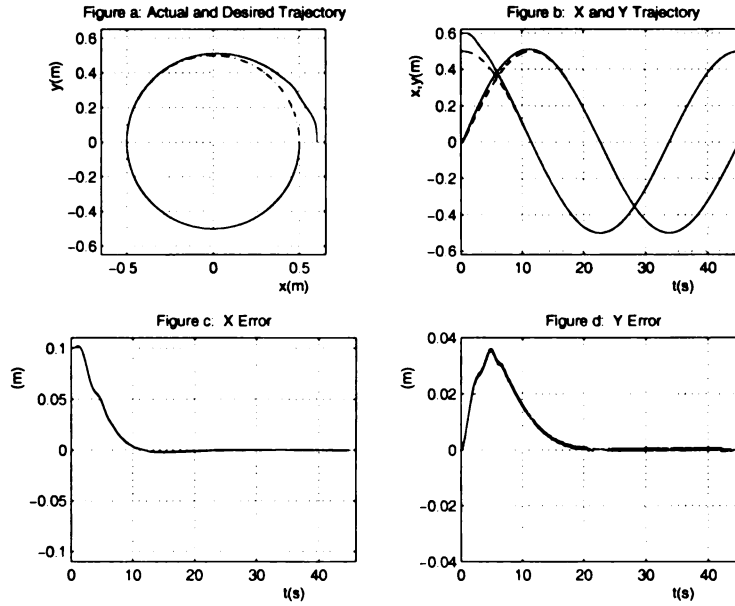


Figure 3.5: Tracking a circle without initial error

where s is the non-time based reference. Instead of the original state variables in the system, $s = \psi$ is chosen as the motion reference in this experiment. It is computed by the measured output x and y by $\psi = \text{atan}_2(y, x)$. For simplicity in the lab experiment, a constant relation between the reference and the time is chosen, though a variable

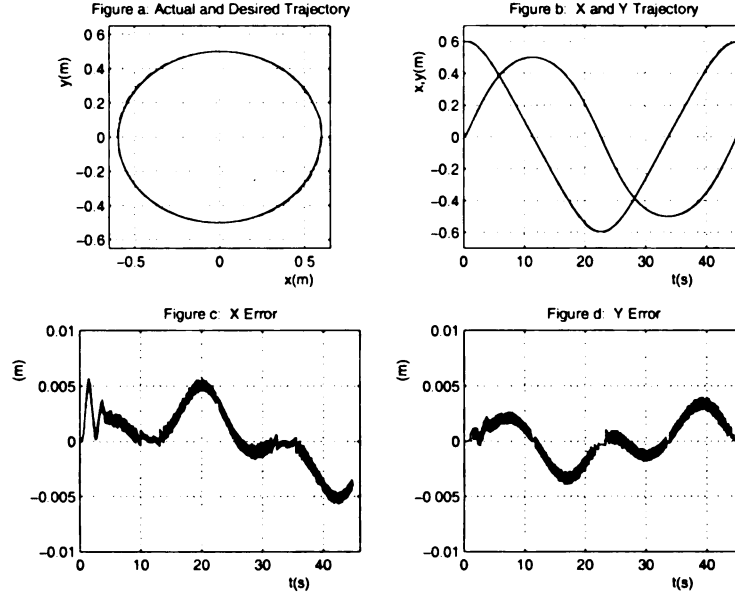


Figure 3.6: Tracking an ellipse without initial error

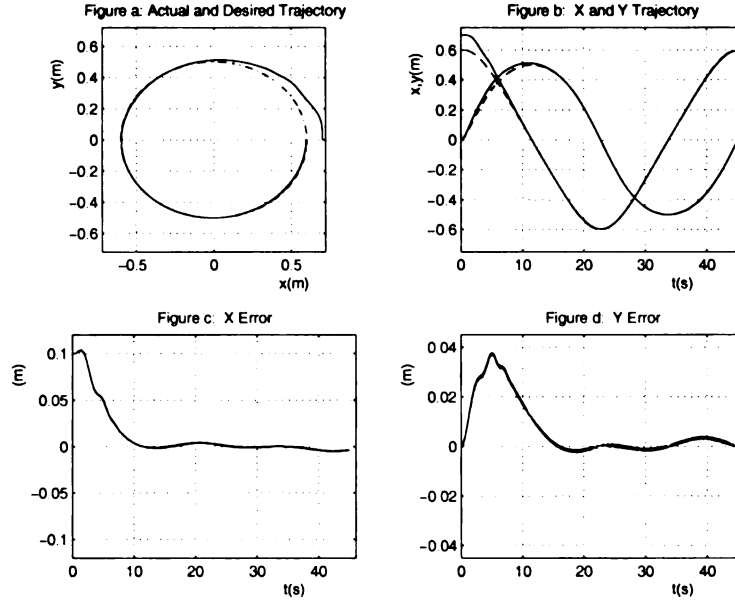


Figure 3.7: Tracking an ellipse with initial error

relation depending on the curvature of the path would be better. The relation is defined by $\psi = vt$. The synthetic time is $t = \frac{1}{v}\psi$. It should be noticed that the motion reference here is not a existed path parameter of the ellipse, i.e., the relation, $x = a \cos \psi, y = b \sin \psi$ are not true.

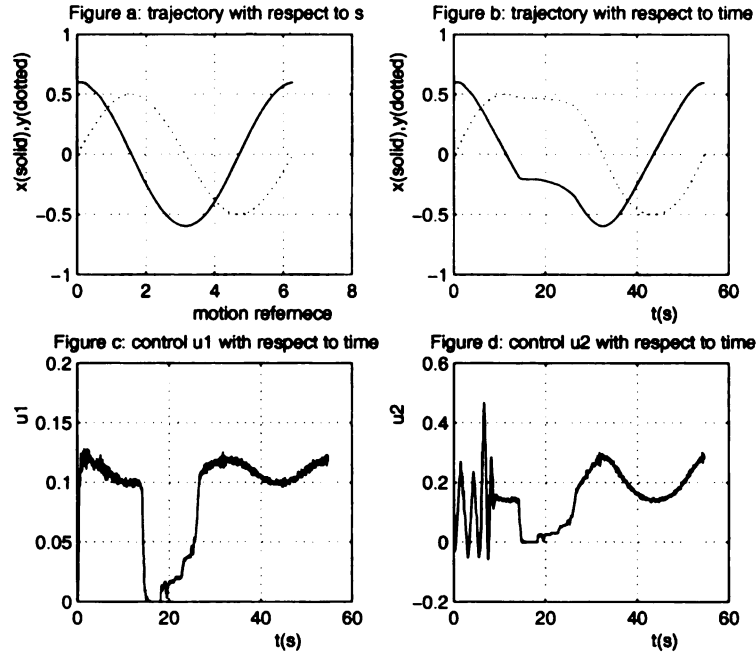


Figure 3.8: Tracking an ellipse with present of unexpected obstacle

To test the control law on a trajectory with large curvature, a small ellipse with $a = 0.6m, b = 0.5m$ is used. The feedback gains are selected as $k_1 = 1, k_2 = 0.4, k_3 = 8, k_4 = 3$ in the experiments. From the experiment result (Figure 3.6 and 3.7), it can be seen that the robot approaches the desired trajectory asymptotically and the static tracking error is bounded by $5mm$.

An interesting result is shown in Figure 3.8. A unexpected obstacle occurred at time $t = 16sec$. When an obstacle is detected by the laser sensor equipped on the mobile robot, it is seen that the robot is slowed down. This is done by relating the motion reference to the distance with respect to the obstacle. Instead of using a constant v_0 to relate the motion reference and time t , the distance from the obstacle to the robot, d , is also considered into the relation. For simplicity of the implementation, three situations are considered. When d is large enough, the distance to the obstacle is not considered in the design of the motion reference. When d is small enough, s will keep constant to suspend the system. It is scaled down gradually when the obstacle is approaching. Of course it can also be designed as continuous function of the distance

to the obstacle. From figure 3.8(a) and (b), it is seen that the mobile robot almost stopped when an obstacle is close. When the obstacle is removed gradually, the robot resumes its action and the originally path is tracked. Figures 3.8(c) and (d) show the control commands. This experiment has demonstrated that the relation $s = v(t)$ can be designed to accommodate the sensor information without any effect on the design of the time based controller.

CHAPTER 4

SINGULARITY ANALYSIS AND SINGULARITY-FREE CONTROLLER OF THE ARM

4.1 Introduction

One advantage of the mobile manipulator is that the kinematic redundancy can be used to avoid singular configuration of the manipulator arm. The purpose of investigating robot arm singularity is to ensure the robot arm dexterity by positioning the mobile platform and/or ensure the stability of the robot arm while approaching singular configurations. This chapter first analyzes singular configurations of the manipulator arm, and then presents a hybrid system design of robot arm controller for singularityless robot motion control. To ensure feasible robot motion in the neighborhood of kinematic singularities as well as at the singularities, a hybrid robot motion controller is designed to integrate joint level control and task level control. The hybrid control system has a two-layered hierarchical structure, a discrete layer and a continuous layer. The workspace is partitioned into subspaces based on the singular configurations of the robot. Switching between continuous controllers is involved when the robot travels across subspaces. The discrete layer is comprised of the switching logic and the continuous layer includes the controllers used in various subspaces. With the hybrid controller, the robot not only can pass by the singular configurations, but also can go through and stay at the singular configurations. The stability of the hybrid system is analyzed using Multiple Lyapunov Function theory. The singularity-free motion control for mobile manipulators will be presented in the Chapter 5.

The existence of robot arm singularities can be seen and analyzed from its task level controller. The dynamic model for a robot arm with n joints in the joint space

can be written as

$$D(q)\ddot{q} + c(q, \dot{q}) + g(q) = u, \quad (4.1)$$

where $q = \{q_1, q_2, \dots, q_n\}^T$ is the $n \times 1$ vector of joint displacements, u is the $n \times 1$ vector of applied torques, $D(q)$ is the $n \times n$ positive definite manipulator inertia matrix, $c(q, \dot{q})$ is the $n \times 1$ centripetal and coriolis terms, and $g(q)$ is the $n \times 1$ vector of gravity term. A robot task can be defined in either the joint space or the task space. Robot controllers can be designed accordingly.

For a robot task specified in the n dimensional joint space, denoted by $q^d, \dot{q}^d, \ddot{q}^d$, a computed torque controller at joint level can be formulated as:

$$u_1 = D(q)(\ddot{q}^d + K_{Vq}e_{q2} + K_{Pq}e_{q1}) + c(q, \dot{q}) + g(q), \quad (4.2)$$

where K_{Vq} and K_{Pq} are feedback gain matrices, $e_{q1} = q^d - q$, and $e_{q2} = \dot{q}^d - \dot{q}$. The error dynamics in joint space can be described by

$$\begin{aligned} \dot{e}_{q1} &= e_{q2} \\ \dot{e}_{q2} &= -K_{Pq}e_{q1} - K_{Vq}e_{q2}. \end{aligned} \quad (4.3)$$

It is straightforward to prove that system (4.3) is asymptotically stable for appropriate gain matrices K_{Pq}, K_{Vq} [6]. Irrespective of the current robot configuration, the system (4.3) can track any reasonable trajectories given as $q^d, \dot{q}^d, \ddot{q}^d$. In other words, controller (4.2) is valid in the entire workspace.

However, a robot task is generally represented by the desired end-effector position and orientation, a robot controller in task space is therefore desirable. To develop a task level controller, the robot model (4.1) needs to be represented in the form of task level variables. Let $x_e = \{x_1, x_2, \dots, x_m\}^T$ represent the position and orientation of

the end effector in the task space, the relation between joint space variable q and task space variable x_e can be represented by $x_e = h_e(q)$, where h denotes a vector consisting of m scalar functions. In general case, $m = 6$ and $x_e \in \mathbb{R}^m$ under certain assumption. The n dimensional joint velocity and m dimensional end effector task velocity can be related by the Jacobian equation: $\dot{x}_e = J_e \dot{q}$, where J_e is the Jacobian matrix. For a nonredundant robot, $n = m$, J_e is a square matrix and the relation between joint level and task level variables is quite straightforward. For a kinematically redundant manipulator, $n > m$, which means the manipulator has more degrees of freedom than necessary to execute a task. The concept of extended task space for kinematically redundant manipulators was introduced in [12]. Based on the kinematic redundancy resolution algorithms, an independent constraint task vector $x_c = h_c(q) \in \mathbb{R}^{n-m}$ can be chosen. Therefore, for a redundant robot, the task involves not only the end effector motion, but also the achievement of additional performances by utilizing the kinematic redundancy. In the mapping $\dot{x}_c = J_c \dot{q}$, the constraint task Jacobian matrix J_c can be chosen as a matrix consisting of the orthonormal vectors that span the null space of J_e . The Jacobian equation in the extended task space can then be represented by $\dot{x} = J \dot{q}$, where $x = \{x_e^T, x_c^T\}^T$ is task level variables in the extended task space and

$$J = \begin{bmatrix} J_e \\ J_c \end{bmatrix}$$

is the Jacobian matrix in the extended task space. It is worth noting that $J_e J_c^T = 0$ [12]. In the extended task space, the redundant manipulators can be treated as nonredundant manipulators. Therefore, the acceleration in task and joint space can be related by $\ddot{x} = \dot{J} \dot{q} + J \ddot{q}$. Considering (4.1), an robot dynamic model in the term of extended task space variables can then be described as follows:

$$DJ^{-1}(\ddot{x} - \dot{J}\dot{q}) + c(q, \dot{q}) + g(q) = u \quad (4.4)$$

where the Jacobian matrix is $n \times n$. Given the desired path in task space, x^d , the robot controller in task space can be described as:

$$u_2 = DJ^{-1}(\ddot{x}^d - \dot{J}\dot{q} + K_{Vx}e_{x2} + K_{Px}e_{x1}) + c(q, \dot{q}) + g(q), \quad (4.5)$$

where

$$e_{x1} = x^d - x, e_{x2} = \dot{x}^d - \dot{x}.$$

The error dynamics in task space can be described as:

$$\begin{aligned} \dot{e}_{x1} &= e_{x2} \\ \dot{e}_{x2} &= -K_{Px}e_{x1} - K_{Vx}e_{x2}, \end{aligned} \quad (4.6)$$

It is also easy to show that system (4.6) is locally asymptotically stable for appropriate gain matrices K_{Px}, K_{Vx} [6].

4.2 Robot Arm Singularities

By comparing the controllers in (4.2) and in (4.5), it can be seen that the existence of controller (4.5) depends on the existence of J^{-1} , while (4.2) does not. At the singular configuration or in its vicinity, the Jacobian J is singular or ill-conditioned. The determinant of J^{-1} could be very large, which will result in unreasonable large joint torques. The robot configurations such that $|J| = 0$ are called robot singular configurations. Theoretically, the robot singular configurations and their corresponding analytical singularity conditions can be obtained based on the analysis of the Jacobian J . To better understand the kinematic redundancy and singularity, consider the Singular Value Decomposition(SVD) of the matrix J

$$J = U\Sigma V^T = \sum_{i=1}^n \sigma_i u_i v_i^T, \quad (4.1)$$

where U is the $(n \times n)$ orthonormal matrix of the output singular vectors u_i , V is the $(n \times n)$ orthonormal matrix of the input singular vectors v_i , and Σ is a diagonal matrix whose elements are the singular values of J , $\sigma_1, \sigma_2, \dots, \sigma_n$. At a singular configuration, one of the singular value σ_i is 0. Thus,

$$u_i^T J(q) = u_i^T U \Sigma V^T = 0.$$

This means that motion in the joint space will result in zero motion in the output direction u_i of the task space. This direction in the task space is called degenerate direction.

Each singular configuration of the robot can be represented by a singular condition in terms of the robot parameters and joint variables. The singular conditions can be used to check the proximity of robot configuration to the singular configuration. Since singular configurations may varies for mechanical structure, the singular conditions of manipulators are analyzed based on three classes of mechanical structures: (a) nonredundant manipulator with spherical wrist; (b) nonredundant manipulator with non-spherical wrist; (c) redundant manipulators. Corresponding to the singular configurations, a vector of singular conditions, denoted by Γ , can be derived.

4.2.1 Non-redundant Robot with Spherical Wrist

A general task for an end effector motion requires six degrees of freedom, therefore, $m = 6$ can be assumed. For a 6-DOF robot manipulator, which consists of a 3-DOF forearm and a 3-DOF spherical wrist, $J = J_e$ is a 6×6 matrix. Due to the mechanical structure of the spherical wrist, the motion of the position and orientation can be decoupled. As a result, the Jacobian can be decoupled in such that the singular configurations resulting from the arm and the wrist are separated. According to

[15],[67], the Jacobian J for a PUMA-like robot can be decoupled into:

$$\begin{aligned}
 J_e &= \begin{bmatrix} I & W \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \Psi \end{bmatrix} \begin{bmatrix} J_{11} & 0 \\ J_{21} & J_{22} \end{bmatrix} \\
 &= \begin{bmatrix} I & W \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \Psi \end{bmatrix} J_w
 \end{aligned} \tag{4.2}$$

where

$$\begin{aligned}
 J_w &= \begin{bmatrix} J_{11} & 0 \\ J_{21} & J_{22} \end{bmatrix}, \\
 \Psi &= \begin{bmatrix} -\frac{s_O s_A}{c_A} & \frac{c_O s_A}{c_A} & 1 \\ -c_O & -s_O & 0 \\ -\frac{s_O}{c_A} & \frac{c_O}{c_A} & 0 \end{bmatrix}, \\
 W &= \begin{bmatrix} 0 & d_6 \cdot e_k & -d_6 \cdot e_j \\ -d_6 \cdot e_k & 0 & d_6 \cdot e_i \\ d_6 \cdot e_j & -d_6 \cdot e_i & 0 \end{bmatrix},
 \end{aligned}$$

$$J_{11} = \begin{bmatrix} -d_4 s_1 s_{23} - a_1 s_1 c_2 - a_3 s_1 c_{23} - d_2 c_1 & d_4 c_1 c_{23} - a_2 c_1 s_2 - a_3 c_1 s_{23} \\ d_4 c_1 s_{23} + a_1 c_1 c_2 + a_3 c_1 c_{23} - d_2 s_1 & d_4 s_1 c_{23} - a_2 s_1 s_2 - a_3 s_1 s_{23} \\ 0 & -d_4 s_{23} - a_2 - a_3 c_{23} \\ & c_1 (d_4 c_{23} - a_3 s_{23}) \\ & s_1 (d_4 c_{23} - a_3 s_{23}) \\ & -d_4 s_{23} - a_3 c_{23} \end{bmatrix},$$

$$J_{21} = \begin{bmatrix} 0 & -s_1 & -s_1 \\ 0 & -c_1 & -c_1 \\ 1 & 0 & 0 \end{bmatrix},$$

$$J_{22} = \begin{bmatrix} c_1 s_{23} & -c_1 c_{23} s_4 - s_1 c_4 & c_1 c_4 c_{23} s_5 - s_1 s_4 s_5 + c_1 s_{23} c_5 \\ s_1 s_{23} & -s_1 c_{23} s_4 - c_1 c_4 & s_1 c_4 c_{23} s_5 + c_1 s_4 s_5 + s_1 s_{23} c_5 \\ c_{23} & s_{23} s_4 & c_{23} c_5 - s_{23} c_4 s_5 \end{bmatrix},$$

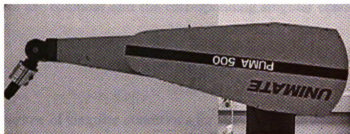
$$e_i = c_1(c_{23}c_4s_5 + s_{23}c_5) - s_1s_4s_5,$$

$$e_j = s_1(c_{23}c_4s_5 + s_{23}c_5) + c_1s_4s_5,$$

$$e_k = c_{23}c_5 - s_{23}c_4s_5.$$

Here s_i , c_i , s_{ij} and c_{ij} stand for $\sin(q_i)$, $\cos(q_i)$, $\sin(q_i + q_j)$ and $\cos(q_i + q_j)$ respectively. a_2, a_3, d_2, d_4 are robot joint parameters, $(O, A, T)^T$ denotes an orientation representation (Orientation, Attitude, Tool angles) [67]. Since the singularity of Ψ caused by $c_A = 0$ can be overcome by choosing other definition of the angle representation ([76],[26]) and W is not singular, the singularity analysis can be obtained by checking J_w . Observing J_{11} and J_{22} of J_w , it can be seen that J_{11} involves only q_1, q_2, q_3 , which correspond to the arm joints, and J_{22} involves only q_4, q_5, q_6 , which correspond to the wrist joints. Singular configurations caused by $J_{11} = 0$ are called arm singularities and singular configurations caused by $J_{22} = 0$ are called wrist singularities.

Since the determinant of J_{11} is $(d_4c_3 - a_3s_3)(d_4s_{23} + a_2c_2 + a_3c_{23})$, two singular configurations can be obtained. One is called boundary singular configuration, as



(a) Boundary singular configuration



(b) Interior singular configuration



(c) Wrist singular configuration

Figure 4.1: Singular configurations of Puma 560

shown in Figure 1, (a), which corresponds to:

$$\gamma_b = d_4 c_3 - a_3 s_3 = 0. \quad (4.3)$$

This situation occurs when the elbow is fully extended or retracted. When fully extended, there is no way to move the end effector forward, this is the degenerate direction. The other situation is called interior singular configuration, as shown in Figure 1, (b), the corresponding singular condition is

$$\gamma_i = d_4 s_{23} + a_2 c_2 + a_3 c_{23} = 0. \quad (4.4)$$

In addition, the wrist singularity, as shown in Figure 1, (c), can be identified by checking the determinant of the matrix J_{22} . The wrist singularity happens when two

wrist joint axes are collinear and the corresponding singular condition is

$$\gamma_w = -s_5 = 0. \quad (4.5)$$

Therefore, a vector of singular conditions for a PUMA-like non-redundant arm with spherical wrist can be given as:

$$\Gamma = \begin{bmatrix} d_4 c_3 - a_3 s_3 \\ d_4 s_{23} + a_2 c_2 + a_3 c_{23} \\ -s_5 \end{bmatrix}$$

By checking the value of singular conditions $\gamma \in \Gamma$, one can decide the proximity to the robot singular configurations. Denoting the manipulability as $\sqrt{JJ^T}$, it can be seen that $\sqrt{JJ^T}$ is proportional to $|\gamma_b \cdot \gamma_i \cdot \gamma_w|$.

4.2.2 *Non-redundant Robot with Non-spherical Wrist*

A robot manipulator with non-spherical wrist generally does not have a decoupled mechanical structure, though constructing a non-spherical wrist is easier. The inverse kinematics of a manipulator with non-spherical wrist generally cannot be computed in closed form and numerical computation needs to be applied. However, the singular configurations of a non-redundant robot with non-spherical wrist can also be obtained by considering the Jacobian [72]. For instance, the determinant of the Jacobian of a manipulator with non-spherical wrist can be derived as given in [72]:

$$|J| = k a_2 a_3 s_3 s_5 (a_1 + a_2 c_2 + a_3 c_{23} - d_5 s_{234}),$$

where a_1, a_2, a_3, d_5 are robot parameters. The singular configurations can be decoupled into three kinds of singular configurations: (i) shoulder singularity, $a_1 + a_2 c_2 +$

$a_3c_{23} - d_5s_{234}$; (ii) elbow singularity, $s_3 = 0$; and (iii) wrist singularity, $s_5 = 0$. The singular condition vector is represented by

$$\Gamma = \begin{bmatrix} a_1 + a_2c_2 + a_3c_{23} - d_5s_{234} \\ s_3 \\ s_5 \end{bmatrix},$$

which is used to check the proximity to the singular configurations.

4.2.3 Singularity Analysis for a Redundant Robot

Generally, a redundant manipulator which has more than necessary degree of freedom is more flexible and dexterous than a nonredundant manipulator, though the singularity analysis of a redundant manipulator is more complex. A singularity could be caused by the singular value of J_e , which is called a kinematic singularity, or the singular value of J_c , which is called an algorithm singularity.

For a redundant manipulator with spherical wrist, the singular configuration of robot arm can be obtained by checking the Jacobian matrix J_e [14]. The position singularity and orientation singularity can also be decoupled due to the mechanical structure. Some of the configurations are boundary configurations, which are non-escapable, whereas some are interior singularities, which are escapable [5]. However, a singular condition vector Γ can still be obtained by analyzing the analytical representation of the Jacobian matrix [14].

For a given constraint task, the singular configurations of J_c can also be determined in advance. However, with a full rank J_e and J_c , there are robot configurations for which $J = [J_e^T, J_c^T]^T$ loses rank. All these singular configurations caused by the choice of J_c are called algorithmic singular configurations. For a given J_c , the algorithm singular conditions can be determined. Therefore, given a redundancy resolution algorithm and an extended task space, the singular condition of matrix J can be

determined.

4.2.4 *Workspace and Singular Conditions*

The workspace of a robot, which is a complex volume calculated from the limit values of the joint variables, can be described by two concepts: reachable workspace and dexterous workspace. A reachable workspace is the volume in which every point can be reached by a reference point on the manipulator end effector. The dexterous workspace is a subset of the reachable workspace, and it does not include the singular configurations and their vicinities. The dexterous workspace is therefore a volume within which every point can be reached by a reference point on the manipulator's end effector in any desired orientation. Unfortunately the dexterous workspace is not a connected space. The singular configurations partition the dexterous space into disconnected subspaces. For example, a wrist singular condition (4.5) of a nonredundant robot can be satisfied at almost any end effector position in the workspace. Thus, for almost any position in the workspace, there exists an orientation of the end effector which satisfies $\gamma_w = 0$. A robot path may therefore be composed of several segments of paths in the dexterous subspaces and several segments of paths in the vicinity of singular configurations. In the dexterous subspaces, there is no problem to control the robot at task level. When the task requires the robot to go from one dexterous subspace to another dexterous subspace, the robot end effector needs to go through the vicinity of singular configurations. At these configurations, the manipulator loses one or more degrees of freedom and the determinant of J approaches zero. In order to generate a reasonable motion in task space, the joint motion constraint may be violated.

Based on the above analysis, the robot workspace, denoted by Ω , can be divided into two kinds of subspaces, the dexterous subspace denoted by Ω_0 and the vicinity of singular configurations denoted by Ω_1 and Ω_2 . Ω_2 specifies a smaller vicinity of the

singular configuration. The definition of Ω_1 and Ω_2 are given as follows:

$$\begin{aligned}\Omega_2 &= \{q \in \mathbb{R}^n, \gamma \in \Gamma \mid |\gamma| < \alpha\} \\ \Omega_1 &= \{q \in \mathbb{R}^n, \gamma \in \Gamma \mid \alpha \leq |\gamma| \leq \beta, \gamma \notin \Omega_2\},\end{aligned}\tag{4.6}$$

where $\beta > \alpha > 0$. Figure 4.2 shows an illustrative plot of the subspaces.

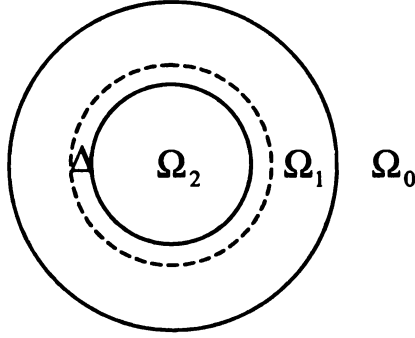


Figure 4.2: Illustrative plot of the robot workspace and subspaces

It is worth noting that $\Omega_0 \cup \Omega_1 \cup \Omega_2 = \Omega$ and $\Omega_0 \cap \Omega_1 \cap \Omega_2 = \emptyset$. A Δ region called dwell region is defined in Ω_1 to avoid the chattering when switching occurs. The definition of a dwell region is given as:

$$\Delta = \{q \in \mathbb{R}^n, \gamma \in \Gamma \mid \alpha < |\gamma| < \alpha + \delta, \gamma \notin \Omega_2\},\tag{4.7}$$

where δ is a constant greater than zero.

4.3 Hybrid Robot Motion Controller

4.3.1 Hybrid System Model

Hybrid control systems involve both continuous and discrete dynamic systems. The evolution of such systems is given by equations of motion that generally depend on both continuous and discrete variables. The continuous dynamics of such systems is generally modeled by several sets of differential or difference equations while the

discrete dynamics describes switching logic of the continuous dynamics. For example, robot controllers (4.2) and (4.5) can switch in the workspace according to certain switching logic.

A general form of the hybrid robot motion controller is defined as follows. The continuous state variable is joint angle q , or the end-effector position and orientation x . The discrete state variables, denoted by $m \in \mathbb{R}^l$ or $m \in \{m_1, m_2, \dots, m_l\}$, represent the switching conditions. The hybrid system model of the robot can be described by

$$\begin{aligned} D(q)\ddot{q} + c(q, \dot{q}) + g(q) &= u \\ m(t) &= f(\Gamma(q(t)), m(t^-)), \end{aligned} \tag{4.1}$$

and the hybrid robot motion controller is given by

$$u(t) = u(x(t), q(t), m(t)). \tag{4.2}$$

The dynamics, f , is governed by the singular conditions. Depending on the current discrete state of the robot and continuous states, f generates the next discrete states. t^- denotes that $m(t)$ is piecewise continuous from the right. f discretizes the continuous states and switches the local controllers. Max-Plus algebra [4] is used to describe the discrete event evolution. It can provide an analytical representation of a discrete event system. The detail design of (4.2) is given in the next subsections.

4.3.2 Design Procedure of a Hybrid Robot Motion Controller

The first step in the hybrid motion controller design is the singularity analysis of the manipulator and definition of the singular and nonsingular regions. For mechanical structures, the kinematic and algorithmic singular configurations and the corresponding singular condition vector Γ can be obtained. The analysis of the singular conditions Γ and the singular values is an important step in the design of the

switching conditions.

The second step is the design of the switching functions based on the definition of the subspaces. Two switching function vectors are generally necessary. One is the switching conditions between the regular region Ω_0 and region Ω_1 , the other is the switching function between region Ω_1 and region Ω_2 . The switching functions can be formulated as:

$$\begin{aligned} m_{s1} &= \text{sgn}(\beta - |\gamma|) \oplus 0, \quad \forall \gamma \in \Gamma \\ m_{s2} &= m_{s2}(t^-)^{\text{sgn}(\alpha + \delta - |\gamma| \oplus 0)} \oplus \text{sgn}(\alpha - |\gamma|) \oplus 0, \quad \forall \gamma \in \Gamma \end{aligned} \quad (4.3)$$

where $\text{sgn}(\cdot)$ is a signum function. The switching function is a function of the singular conditions represented by Max-plus Algebra. The Max-Plus algebra, widely used in the Discrete Event Systems, is defined as $m(t) \in \mathfrak{R}_{max}^n$ where

$$\mathfrak{R}_{max} = \mathbb{R} \cup \{-\infty\}$$

\oplus : Max operation

\otimes : Plus operation.

Here the operations, \oplus and \otimes are defined as: $a \oplus b = \max\{a, b\}$, and $a \otimes b = a + b$. It can be proved that $\{\mathfrak{R}_{max}^n : \oplus, \otimes\}$ is an idempotent and commutative semi-field with zero element $\alpha = -\infty$ and identity element $e = 0$ [4]. It can be seen that $m_{s1} = 0$ in Ω_0 , $m_{s1} = 1$ in Ω_1 and $m_{s2} = 1$ in Ω_2 . It is worth noting that m_{s2} is not only a function of singular condition vector Γ , but also a function of the current status of $m_{s2}(t^-)$. It means that the switching condition for switching into a subspace and switching out a subspace may have different values. This is designed for eliminating the chattering characteristic of switching controller.

The third step is the design of the continuous controllers. Three continuous controllers are to be designed. In subspace Ω_0 , the inverse of J always exists and the task level control (4.5) is effective in this subspace. In subspace Ω_2 , the joint level

controller is used.

In region Ω_1 , though it is close to a singular configuration, a feasible solution of inverse Jacobian can be obtained by pseudo-inverse or singular robust inverse(SRI). Controller (4.5) can still be used after substituting $J^\#$ for J^{-1} , where $J^\#$ is a kind of pseudo-inverse represented by:

$$J^\# = (J^T \cdot J + \lambda m_{s1})^{-1} J^T, \quad (4.4)$$

where λ is the damping factor. Instead of using a constant damping factor, λ is generally defined a function of the singular value σ [17]:

$$\lambda = \lambda(\sigma) = -\frac{\lambda_{max}}{\sigma_{min}}\sigma + \lambda_{max}.$$

It is generally hard to compute the singular value online, the singular condition $\gamma \in \Gamma$ is used in this design:

$$\lambda = \lambda(\gamma) = -\frac{\lambda_{max}}{\gamma_{min}}\gamma + \lambda_{max},$$

where the parameters λ_{max} and γ_{min} are chosen experimentally. The controller in region Ω_1 can then be represented by

$$u_3 = DJ^\#(\ddot{x}^d - \dot{J}\dot{q} + K_{Vx}e_{x2} + K_{Px}e_{x1}) + c + g. \quad (4.5)$$

The stability analysis of controller (4.5) is discussed in [11].

However, it can be proved that the controller based on pseudo-inverse method will cause instability in subspace Ω_2 [41, 50]. Though subspace Ω_2 could be very small, it cuts the dexterous workspace into pieces. If a controller can not make the robot travel through subspace Ω_2 , the singular configurations will greatly restrict the dexterous workspace. Thus the joint level control is enabled in subspace Ω_2 .

With the continuous controllers in different subspaces and the switch functions,

the next step is to formulate the hybrid robot motion controller in the entire workspace utilizing the switching conditions m_{s1} and m_{s2} :

$$u = (I - m_{s2})u_3 + m_{s2}u_1, \quad (4.6)$$

where the switch function m_{s1} is within u_3 . Based on the stability analysis of the continuous controllers in their respective region, the stability of the hybrid motion controller in the entire workspace should also be investigated. Section 4.4 discusses the stability of such a system.

The last step of the hybrid robot motion controller design is the path planning of the continuous controllers. Given desired path in the task space, x^d , controllers (4.5) and (4.5) are complete. To complete controller (4.2), x^d should be converted into q^d in region Ω_2 . The mapping should be considered in the context of the mechanical structure.

4.3.3 Hybrid Motion Controller for a PUMA-like robot manipulator

For a Puma-like robot manipulator, the continuous controllers (4.2) and (4.5) can be used in region Ω_2 and Ω_0 respectively. The controller design in region Ω_1 and the switching functions are considered based on the characteristics of a PUMA-like robot manipulator.

Continuous controller design in Ω_1

In this subspace, the robot is in the neighborhood of singular configurations. Using direct J^{-1} in task level will result in large torques. The modified damped least squares method is used in the hybrid robot motion controller in this subspace. According to

(4.2), the inverse Jacobian J^{-1} in (4.5) can be represented by

$$J^{-1} = J_w^{-1} \cdot \begin{bmatrix} I & 0 \\ 0 & \Psi \end{bmatrix}^{-1} \cdot \begin{bmatrix} I & W \\ 0 & I \end{bmatrix}^{-1}, \quad (4.7)$$

where

$$J_w = \begin{bmatrix} J_{11} & 0 \\ J_{21} & J_{22} \end{bmatrix}. \quad (4.8)$$

In the neighborhood of singular configurations, J_w is ill-conditioned and the inverse of J_w can be computed by the damped least squares technique [47]. Since a uniformed damping factor λ in the entire workspace may cause unnecessary position and orientation errors in the regular region, a damping vector based on the analysis of the inverse of J_w [15] [47] can be defined as:

$$\Lambda(t) = \begin{bmatrix} \lambda_1(t) \\ \lambda_2(t) \\ \lambda_3(t) \\ \lambda_4(t) \\ \lambda_5(t) \\ \lambda_6(t) \end{bmatrix} = \begin{bmatrix} k_i \oplus 0 \\ k_i \oplus k_b \oplus 0 \\ k_b \oplus 0 \\ k_w \oplus 0 \\ 0 \\ k_w \oplus 0 \end{bmatrix} \quad (4.9)$$

where

$$\begin{aligned} k_b &= k_{b0}(1 - |\gamma_b|/\beta_b) \\ k_i &= k_{i0}(1 - |\gamma_i|/\beta_i) \\ k_w &= k_{w0}(1 - |\gamma_w|/\beta_w). \end{aligned}$$

Λ is a vector of the positive damping factors depending on the proximity to the singular configurations. Denoting the singular value decomposition of J_w as

$$J_w = \sum_{i=1}^6 \sigma_i u_i v_i^T = U \cdot \Sigma_1 \cdot V^T \quad (4.10)$$

where $v_i, u_i, i = 1, \dots, 6$ orthonormal bases of \mathbb{R}^6 . $\sigma_i, i = 1, \dots, 6$ are the singular values of J_w . U and V are orthonormal matrices and Σ_1 is diagonal. The pseudoinverse of J_w is:

$$J_w^\# = (J_w^T \cdot J_w + V \text{diag}(\Lambda) m_{s1}(t) V^T)^{-1} \cdot J_w^T \quad (4.11)$$

where $m_{s1}(t)$ is the switching condition. The inverse of J based on $J_w^\#$ is called $J^\#$. Here $\text{diag}(v)$ denotes a matrix whose diagonal elements are vector v , and the remaining elements are zeroes. It is worth noting that (4.11) is different from (4.4) to reflect the characteristic of Puma-like robot.

The controller of the robot in subspace Ω_0 and Ω_1 can be synthesized by:

$$u_3 = DJ^\#(\ddot{x}^d - \dot{J}\dot{q} + K_{Vx}e_{x2} + K_{Px}e_{x1}) + c + g. \quad (4.12)$$

Here $x = \{p_x, p_y, p_z, O, A, T\}$, where $(p_x, p_y, p_z)^T$ denotes the position of the end-effector in the task space.

Substitute (4.12) into (4.4), the error dynamics of the system can be obtained as:

$$\begin{aligned} \dot{e}_{x1} &= e_{x2} \\ \dot{e}_{x2} &= (I - JJ^\#)(\ddot{x}^d - \dot{J}\dot{q}) - JJ^\#(K_{Px}e_{x1} + K_{Vx}e_{x2}) \end{aligned} \quad (4.13)$$

To analyze the stability of the system, the term $JJ^\#$ in the error dynamics (4.13) is a key component. Considering (4.2), (4.7) and (4.10), $JJ^\#$ can then be simplified as

follows:

$$\begin{aligned}
k &= JJ^\# \\
&= \text{diag} \left[\frac{\sigma_1^2}{\sigma_1^2 + m_1} \quad \frac{\sigma_2^2}{\sigma_2^2 + m_2} \quad \frac{\sigma_3^2}{\sigma_3^2 + m_3} \quad \frac{\sigma_4^2}{\sigma_4^2 + m_4} \quad \frac{\sigma_5^2}{\sigma_5^2 + m_5} \quad \frac{\sigma_6^2}{\sigma_6^2 + m_6} \right]^T
\end{aligned}$$

Defining

$$k_{\min} = \min_{q,i} \left\{ \frac{\sigma_i^2}{\sigma_i^2 + m_i} \right\},$$

it can be seen that k_{\min} becomes smaller and smaller when the robot configuration is approaching singular configurations. The stability of the controller (4.12) in subspace Ω_0 and Ω_1 depends on k . In subspace Ω_0 , $m_{s1} = 0$, $k = I$, the error dynamics (4.13) in region Ω_0 becomes

$$\begin{aligned}
\dot{e}_{x1} &= e_{x2} \\
\dot{e}_{x2} &= -K_{Px}e_{x1} - K_{Vx}e_{x2}
\end{aligned} \tag{4.14}$$

It is obvious that system (4.14) is asymptotically stable in subspace Ω_0 for appropriate gain matrices. While in subspace Ω_1 , the error dynamics of the system can be described by

$$\begin{aligned}
\dot{e}_{x1} &= e_{x2} \\
\dot{e}_{x2} &= (I - k)(\ddot{x}^d - \dot{J}\dot{q}) - k(K_{Px}e_{x1} + K_{Vx}e_{x2})
\end{aligned} \tag{4.15}$$

According to the discussion in [11],[41], a candidate Lyapunov function of (4.15) can be defined as

$$V = \frac{1}{2}[e_{x1}^T K e_{x1} + (\mu e_{x1} + e_{x2})^T (\mu e_{x1} + e_{x2})] \tag{4.16}$$

where K is positive definite matrix and μ is positive constant. The derivative of

the candidate Lyapunov Function can be derived by

$$\begin{aligned}
\dot{V} &= e_{x1}^T K e_{x2} + (\mu e_{x1} + e_{x2})^T (\mu e_{x2} + (I - k)(\ddot{x}^d - \dot{J}\dot{q}) - k(K_{Px} e_{x1} + K_{Vx} e_{x2})) \\
&= -\mu e_{x1}^T k K_{Px} e_{x1} - e_{x2}^T (k K_{Vx} - \mu I) e_{x2} \\
&\quad + (\mu e_{x1} + e_{x2})^T (I - k)(\ddot{x}^d - \dot{J}\dot{q}) + e_{x1}^T (I - k)(K_{Px} + \mu K_{Vx}) e_{x2}
\end{aligned}$$

Here $K = K_{Px} + \mu K_{Vx} - \mu^2 I$. μ can be chosen such that $K_{Vx} - \mu I$ and K are positive definite. Denoting vector $v = \dot{J}\dot{q}$, an upper bound of the vector term can be given by[11]:

$$||v|| \leq \frac{r}{\sigma_{min}^2} (v_M^2 + 2v_M ||e_{x2}|| + ||e_{x2}||^2),$$

where v_M denotes the maximum norm of the desired end effector velocity, σ_{min} denotes the minimum of the singular value of matrix J , and

$$r = \max_q \sqrt{\sum_{i=1}^6 ||N_i||^2},$$

where $N_i = \frac{\partial J_i}{\partial q}$ and J_i is the i th row of matrix J . Thus

$$\begin{aligned}
\dot{V} &\leq -k_{min} \lambda_{min} ||e_x||^2 + (1 - k_{min}) \cdot ||e_x|| (k_1 + k_2 ||e_x|| + k_3 ||e_x||^2) \\
&= -(1 - \theta) k_{min} \lambda_{min} ||e_x||^2 + (1 - k_{min}) \cdot ||e_x|| (k_1 + (k_2 - k_4) ||e_x|| + k_2 ||e_x||^2) \\
&= -(1 - \theta) k_{min} \lambda_{min} ||e_x||^2, \quad \forall \quad (k_1 + (k_2 - k_4) ||e_x|| + k_2 ||e_x||^2) \leq 0.
\end{aligned}$$

where

$$k_1 = a_M + \sqrt{1 + \mu^2} \frac{r}{\sigma_{min}^2} v_M^2, \quad k_2 = \sqrt{1 + \mu^2} \frac{2r}{\sigma_{min}^2} v_M + ||(I - k)(K_{Px} + \mu K_{Vx})||,$$

$$k_3 = \sqrt{1 + \mu^2} \frac{r}{\sigma_{min}^2}, \quad k_4 = \frac{\theta k_{min} \lambda_{min}}{1 - k_{min}},$$

and λ_{min} is the minimum singular value of μK_{P_x} and $K_{V_x} - \mu I/k_{min}$. It can be shown that $(k_1 + (k_2 - k_4)||e_x|| + k_2||e_x||^2) \leq 0$ can be satisfied by regulating k_4 . From the above derivation, it can be seen that the stability depends on the value of k_{min} . In subspace Ω_0 , $k_{min} = 1$, \dot{V} is negative definite and the system is asymptotically stable. In subspace Ω_1 , $k_{min} \lambda_{min} - (1 - k_{min})\eta\sqrt{1 + \mu^2} \leq 0$, the system is uniformly asymptotically ultimate bounded. The bound is ζ_1 where $\zeta_1 < \zeta_2$ are the root of $k_{min} \lambda_{min} - (1 - k_{min})\eta\sqrt{1 + \mu^2} = 0$.

However, in subspace Ω_2 , the value of k_{min} is close to zero, and $-k_{min} \lambda_{min} ||e_x||^2 + (1 - k_{min}) \cdot ||e_x|| (k_1 + k_2 ||e_x|| + k_3 ||e_x||^2)$ may be greater than zero. The system stability can not be ensured. When the robot approaches singular configurations, *i.e.*, in region Ω_2 , some of the elements of k become very small. Due to this, the corresponding elements of the gain matrices become small. Furthermore, at the singular configurations, $k_{min} = 0$. Assuming σ_i is zero, the i th element of (4.13) can be written as:

$$\begin{aligned} \dot{e}_{x1i} &= e_{x2i} \\ \dot{e}_{x2i} &= \ddot{x}_i^d - (\dot{J}\dot{q})_i \end{aligned}$$

It can be seen that the system is an open loop system. Though it can be proved the system is ultimately bounded[41], large errors in the task space are expected in the vicinity of singular configurations.

Controller design in subspace Ω_2

To achieve stable control in region Ω_2 , a switching control is needed so that joint level controller (4.2) can be enabled in Ω_2 . Two discrete variables, m_7 and m_8 , are defined

to represent arm and wrist singularities respectively,

$$\begin{aligned} m_7(t) &= m_7(t^-)^{\text{sgn}(\alpha_i + \delta - |\gamma_b|) \oplus \text{sgn}(\alpha_b + \delta - |\gamma_b|) \oplus 0} \oplus \text{sgn}(\alpha_i - |\gamma_i|) \oplus \text{sgn}(\alpha_b - |\gamma_b|) \oplus 0 \\ m_8(t) &= m_8(t^-)^{\text{sgn}(\alpha_i + \delta - |\gamma_w|) \oplus 0} \oplus \text{sgn}(\alpha_w - |\gamma_w|) \oplus 0. \end{aligned} \quad (4.17)$$

Here $m_7(t^-)$ and $m_8(t^-)$ represent the values of m_7 and m_8 before a time instance t respectively. α_i, α_b and α_w are used to specify subspace Ω_2 . In this design, α and β take different values for different singular configurations. Depending on the values of γ_b, γ_i and γ_w , m_7 and m_8 will determine what kind of continuous controllers should be used. It is worth noting that in region Δ , the values of m_7 and m_8 not only depend on γ_b, γ_i and γ_w , but also depend on the previous value of $m_7(t^-)$ and $m_8(t^-)$. The Δ region, which is defined in (4.7), is called a dwell region. It is introduced to avoid chattering along the switching surface. In other words, the switching surfaces for switching into joint level control and switching out of joint level control are different. After the controller switches into joint space control, it can not switch back to task space control unless the robot reaches a larger region $\Omega_2 \cup \Delta$. This strategy can effectively avoid chattering and ensure the system stability.

A switching matrix, m_{s2} , is defined based on m_7 and m_8 as:

$$m_{s2}(t) = \begin{bmatrix} m_7(t)I_{3 \times 3} & 0 \\ 0 & (m_7(t) \oplus m_8(t))I_{3 \times 3} \end{bmatrix}. \quad (4.18)$$

Hybrid robot motion controller

The singularityless robot motion controller can be represented by

$$u = (I - m_{s2})u_3 + m_{s2}u_1 \quad (4.19)$$

The above controller involves both continuous and switching controls, as shown

in Figure 4.3. It forms a two-layered hierarchical hybrid controller, a continuous layer and a discrete layer. The switching conditions are dependent on the proximity to the singular configurations and the previous controller status. When the robot approaches the singular configurations, the hybrid controller first uses damped least squares to achieve an approximate motion of the end effector. It will then switch into joint level control if the robot reaches the singular configuration.

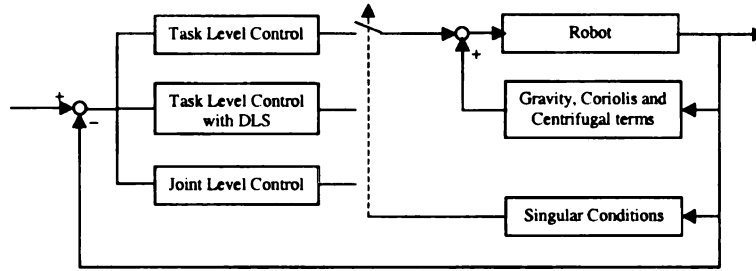


Figure 4.3: Hybrid system controller framework

For a robot with a spherical wrist, it is worth noting that if the position and orientation control are decoupled, joint level control and task level control can be enabled simultaneously. This happens when $m_7 = 0$ and $m_8 = 1$. It means that the position is controlled at task level and the orientation is controlled at joint level. However, the position and orientation are generally coupled if the tool center of the end effector is used to indicate the desired position. For a given task $x^d = \{p_x^d, p_y^d, p_z^d, O^d, A^d, T^d\}$, the desired position $\{p_x^d, p_y^d, p_z^d\}$ is coupled with the desired orientation $\{O^d, A^d, T^d\}$. However, it can be easily seen that after proper transformation of a given task, the desired position at the wrist center can be taken as the desired position at task level. It is only related to the joint angles q_1, q_2, q_3 and can be obtained from x^d [67]. The orientation control will only depend on q_4^d, q_5^d, q_6^d . The end effector position and orientation can therefore be decoupled and can be controlled separately. Since the wrist singular configuration may happen at almost any end effector position, the separated position control and orientation control can ensure a relatively larger continuous dex-

terous workspace for position control.

Path planning for switching

In the regular region Ω_0 and region Ω_1 , the task is represented in Cartesian task space. The task, however, in region Ω_2 needs to be transformed into joint space since a joint level controller is used. Given x^d in task space, joint level command q^d needs to be calculated in subspace Ω_2 . Substantial research has been done on this topic [15, 47, 75, 56, 49]. By the relation between task space and joint space variables,

$$x = h(q),$$

a first order approximation of the kinematic function at a configuration q can be written as:

$$dx = J(q)dq,$$

where dx and dq represent motion increment in task space and joint space respectively. At robot singular configurations, $\text{rank}(J(q)) < n$. When a manipulator passes through a singular configuration, a certain degrees of freedom in the singular direction or the degenerate direction is lost. Therefore, there would be no feasible solution for the manipulator to move into the singular direction. For a path close to a singular configuration, an exact solution is possible but very large joint motions are required if the assigned dx has components along directions which become degenerate at the singular configuration. To balance the accuracy and feasibility of the joint space motion, the following criteria is used to find the inverse kinematics:

$$\min_q \{w_1 \|dx - J(q)dq\|^2 + w_2 \|q^d - q\|^2\}.$$

where w_1 and w_2 are positive weight factors. This criteria minimizes the deviation after the task is mapped into joint space, and tries to find q^d such that the least joint movement is needed. An optimal solution to this can be written as [47]:

$$dq = w_1(w_1 J^T(q)J(q) + w_2)^{-1} J(q)^T dx$$

In the computation of dq , it can be found that the joint space variable can be divided into two subspaces: joint variable that can be computed by normal inverse kinematics and joint variable that cannot [15, 75]. For example, at the wrist singular configuration, Figure 4.1(c), only q_4^d and q_6^d cannot be obtained by the inverse kinematics; they are q_2^d and q_3^d at the boundary singularity, Figure 4.1 (a); and q_1^d and q_2^d at the interior singularity, Figure 4.1 (b). Thus, the desired joint level command can be grouped into two subsets: $q^{(1)}$ denotes the joints that can be obtained by inverse kinematics, $q^{(2)}$ denotes the joints that cannot be obtained by inverse kinematics. By checking the values of singular conditions $\gamma_b, \gamma_i, \gamma_w$, the elements of $q^{(2)}$ can be determined. To compute these joint level commands $q^{(2)}$, the following criteria are considered:

$$\min_{q^{(2)}} \{w_1 \|dx - J(q)dq\|^2 + w_2 \|q^d - q\|^2\}.$$

Motion planning should also be considered in the joint space. In the vicinity of a singular configuration, a pure geometrical path can always be tracked, but a trajectory which includes both geometrical and temporal specifications cannot be tracked. In the other word, the geometrical parameter and the temporal parameter can be considered separately, as discussed in [56, 49]. It is easy to understand that an appropriate dx would result in a large dq in the neighborhood of singular configurations. A new path can be planned to satisfy the velocity and acceleration constraints. The continuity of joint velocities and task level velocities are also considered in the planning. After switching from task level control to joint level, the initial velocity for every joint is

the joint velocity prior to switching. If the control switches from joint level control to task level control, the velocity continuity can be ensured by taking $\dot{x} = J\dot{q}$ as the initial velocity in task space.

4.4 Stability Analysis of the Hybrid Motion Controller

In Section 4.3, it has been shown that task level controller (4.12) is uniformly ultimate bounded in the workspace except subspace Ω_2 . And joint level controller (4.2) is asymptotically stable in the entire workspace. This section discusses the stability of controller (4.19) when switching between (4.12) and (4.2) is involved.

Theorem Suppose controller (4.2) is asymptotically stable in workspace and (4.12) is uniformly ultimate bounded in region Ω_1 , there exists a constant $\delta > 0$, as shown in Figure 4.2 and (4.7), such that the hybrid robot motion controller (4.19) is ultimate bounded in the entire workspace of the robot. \triangleleft

Proof The state variables in the error dynamics in joint space and task space are different, due to which the stability analysis of the hybrid motion controller is difficult. The errors in joint space and task space are defined as:

$$e_q = \begin{pmatrix} e_{q1} \\ e_{q2} \end{pmatrix} = \begin{pmatrix} q^d - q \\ \dot{q}^d - \dot{q} \end{pmatrix}, \quad e_x = \begin{pmatrix} e_{x1} \\ e_{x2} \end{pmatrix} = \begin{pmatrix} x^d - x \\ \dot{x}^d - \dot{x} \end{pmatrix}.$$

The error dynamics of the manipulator in joint space can be written by

$$\begin{aligned} \dot{e}_{q1} &= e_{q2} \\ \dot{e}_{q2} &= -K_{Pq}e_{q1} - K_{Vq}e_{q2}, \end{aligned} \tag{4.20}$$

or $\dot{e}_q = f_q(e_q)$. The error dynamics in the task space are

$$\begin{aligned}\dot{e}_{x1} &= e_{x2} \\ \dot{e}_{x2} &= (I - k)(\ddot{x}^d - \dot{J}\dot{q}) - k(K_{Px}e_{x1} + K_{Vx}e_{x2}),\end{aligned}\tag{4.21}$$

or $\dot{e}_x = f_x(e_x)$.

It is easy to prove that system (4.20) is stable. (4.21) has been proved to be ultimately bounded in $\Omega_0 \cup \Omega_1$. However, if the controller switches between u_1 (4.2) and u_3 (4.12), the system will switch between (4.20) and (4.21). Since the state variables in dynamics model (4.20) and (4.21) are different, the relation between the state variables in model (4.20) and (4.21) is derived first.

It is easy to prove that $x = h(q)$ and $\dot{x} = J\dot{q}$ are globally Lipschitz continuous in their defined domain, *i.e.*,

$$\begin{aligned}\|e_{x1}\| &= \|x^d - x\| = \|h(q^d) - h(q)\| \leq L_1\|q^d - q\| = L_1\|e_{q1}\| \\ \|e_{x2}\| &= \|\dot{x}^d - \dot{x}\| = \|J(q^d)\dot{q}^d - J(q)\dot{q}\| \leq L_2\|\dot{q}^d - \dot{q}\| = L_2\|e_{q2}\|\end{aligned}$$

where L_1 and L_2 are constants. In summary, the following relation between joint space error and task space error can be found:

$$\|e_x\| \leq L_b\|e_q\|$$

where L_b is a constant. Similar to the above inequality, the following relation can also be obtained based on the Lipschitz continuity of the function $q = h^{-1}(x)$, $\dot{q} = J^{-1}\dot{x}$ in $\Omega_0 \cup \Omega_1$.

$$\|e_q\| \leq L_a\|e_x\|$$

Therefore, the errors in joint space and task space satisfy the following inequality in

$\Omega_0 \cup \Omega_1$.

$$\|e_q\| \leq L_a \|e_x\|, \quad \|e_x\| \leq L_b \|e_q\| \quad (4.22)$$

Defining the Lyapunov function of system (4.20) and system (4.21)

$$V_1 = \frac{1}{2}(e_{q1}^T K_{Pq} e_{q1} + e_{q2}^T e_{q2})$$

$$V_2 = \frac{1}{2}[e_{x1}^T K e_{x1} + (\mu e_{x1} + e_{x2})^T (\mu e_{x1} + e_{x2})].$$

Since the system (4.20) has been proved asymptotically stable in the entire workspace, and (4.21) has been proved to be uniformly ultimate bounded, the Lyapunov functions satisfy the inequalities in region $\Omega_0 \cup \Omega_1$ [35]

$$a_1 \|e_q\| \leq V_1(e_q) \leq b_1 \|e_q\|$$

$$\frac{\partial V_1(e_q)}{\partial e_q} f_q(e_q) \leq -c_1 \|e_q\|$$

$$a_2 \|e_x\| \leq V_2(e_x) \leq b_2 \|e_x\|$$

$$\frac{\partial V_2(e_x)}{\partial e_x} f_x(e_x) \leq -c_2 \|e_x\|$$

where $a_1, b_1, c_1, a_2, b_2, c_2$ are positive scalars. For an initial time t_0 , the following inequalities can be obtained.

$$V_1(e_q(t_0 + \tau)) \leq e^{-\lambda_1 \tau_1} V_1(e_q(t_0))$$

$$V_2(e_x(t_0 + \tau)) \leq e^{-\lambda_2 \tau_2} V_2(e_x(t_0)) \quad (4.23)$$

where $\lambda_1 = c_1/b_1$ and $\lambda_2 = c_2/b_2$ are positive scalars. In order to prove the stability of the hybrid controller (4.19), it is essential to show that V_1 is monotonely decreasing at all odd number of switch instances, t_1, t_3, t_5, \dots , and V_2 is monotonely decreasing

at all even switching instances, t_0, t_2, t_4, \dots .

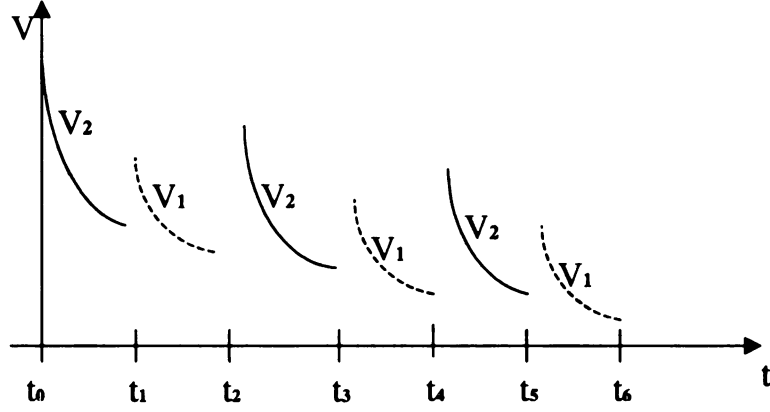


Figure 4.4: A typical switching sequence of the hybrid motion controller

Based on the switching sequence in Figure 4.4,

$$\begin{aligned}
V_1(e_q(t_1)) &\leq b_1 \|e_q(t_1)\| \leq b_1 L_a \|e_x(t_1)\| \\
&\leq \frac{b_1 L_a}{a_2} V_2(e_x(t_1)) \leq \frac{b_1 L_a}{a_2} e^{-\lambda_2(t_1-t_0)} V_2(e_x(t_0)) \\
V_2(e_x(t_2)) &\leq b_2 \|e_x(t_2)\| \leq b_2 L_b \|e_q(t_2)\| \\
&\leq \frac{b_2 L_b}{a_1} V_1(e_q(t_2)) \leq \frac{b_2 L_b}{a_1} e^{-\lambda_1(t_2-t_1)} V_1(e_q(t_1)) \\
&\leq \frac{b_2 L_b}{a_1} \frac{b_1 L_a}{a_2} e^{-\lambda_1(t_2-t_1)} e^{-\lambda_2(t_1-t_0)} V_2(e_x(t_0))
\end{aligned} \tag{4.24}$$

Form the above inequalities, it can be seen that

$$\begin{aligned}
V_2(e_x(t_2)) &\leq \frac{b_2 L_b}{a_1} \frac{b_1 L_a}{a_2} e^{-\lambda_1(t_2-t_1)} e^{-\lambda_2(t_1-t_0)} V_2(e_x(t_0)) \\
&= c e^{\mu(-\lambda_1(t_2-t_1)-\lambda_2(t_1-t_0))} e^{(1-\mu)(-\lambda_1(t_2-t_1)-\lambda_2(t_1-t_0))} V_2(e_x(t_0)) \\
&\leq e^{(1-\mu)(-\lambda_1(t_2-t_1)-\lambda_2(t_1-t_0))} V_2(e_x(t_0)), \quad \text{for } c e^{\mu(-\lambda_1(t_2-t_1)-\lambda_2(t_1-t_0))} \leq 1
\end{aligned} \tag{4.25}$$

Here

$$c = \frac{b_2 L_b}{a_1} \frac{b_1 L_a}{a_2}$$

is a constant and $0 < \mu < 1$. Based on Multiple Lyapunov Function method [8], [52],[69], the system can be proved to be stable as long as (4.25) holds. It means that $c e^{\mu(-\lambda_1(t_2-t_1)-\lambda_2(t_1-t_0))} \leq 1$ is a sufficient condition for the stability of the system during switching. This can be achieved by regulating two sets of parameters, λ_1 and λ_2 ; $t_2 - t_1$ and $t_1 - t_0$. The values of λ_1 and λ_2 are related to the gain matrices K_{px} , K_{vx} , K_{pq} and K_{vq} . In the implementation of the controller, high gains and high sampling frequency are chosen to ensure proper λ_1 and λ_2 . The values of $t_2 - t_1$ and $t_1 - t_0$ are also related to δ . The boundary for switching into and out of a continuous controller is not the same. This forms the dwell region Δ , and can effectively avoid chattering at switching. From the definition of (4.7), the value of m_{s2} cannot be changed immediately. Since $t_2 - t_1$ and $t_1 - t_0$ are time intervals when a continuous controller is enabled, they can be regulated by the value of δ . Therefore, the gain matrices and the size of the dwell region Δ are used to ensure that $c e^{\mu(-\lambda_1(t_2-t_1)-\lambda_2(t_1-t_0))} \leq 1$. The above analysis is valid for $\|e_x > \zeta_1\|$, the switching system is uniformly ultimate bounded.

In the design of the controllers, the size of the dwell region can be chosen based on two situations. The first one is the ultimate bounded. Considering controller (4.12) is ultimate bounded by ζ_1 , therefore $\|e_x\| \geq \|e_{x1}\| \geq \zeta_1$ should be ensured by δ . The other one is the the gain matrix and motion planning. It can be proved that the transition of e_x and e_q are bounded by their initial values during switching. And the bound is related to the gain matrices and the desired trajectory. It can be assumed that the initial value of errors are also bounded. Therefore, e_x and e_q are bounded. Thus, δ can be chosen that $e_q(t) \in \Omega_2$ for $t > T$ and T is a time instance after finite switches between joint level and task level control. By choosing an appropriate region, the switching sequence can also be completed in one cycle.

The same procedure can be applied to time instances t_4, t_6, \dots for V_2 . Therefore, V_2 is monotonely decreasing at all even switching instances. Similarly, V_1 is monotonely

decreasing at all odd switching instances t_1, t_3, t_5, \dots . Thus, the switching system is stable. \triangleleft

4.5 Experimental Implementation and Testing

The proposed hybrid robot motion controller was implemented on a PUMA 560 robot. Figure 4.5 shows the experimental setup. The controller ran under the real-time operating system QNX [1]. The parameters in the experiments are chosen as $\beta_i = \beta_w = 0.15, \beta_b = 0.25, \delta = 0.04, \alpha_i = 0.08, \alpha_b = 0.08, \alpha_w = 0.03, k_{i0} = k_{w0} = 0.2, k_{b0} = 1, w_1 = w_2 = 1, K_{P_x} = \text{diag}(4800.0, 6000.0, 4800.0, 200.0, 200.0, 280.0), K_{V_x} = \text{diag}(80.0, 88.1, 88.2, 22.0, 24.0, 25.6), K_{P_q} = \text{diag}(2400.0, 2400.0, 2400.0, 1100.0, 1100.0, 1100.0)$ and $K_{V_q} = \text{diag}(50.2, 50.5, 50.1, 36.0, 36.0, 36.0)$. All the parameters are determined experimentally. The sampling period is set as $1ms$. Given a path, the trajectory was computed based on the path planning approach in [26]. A trajectory is then formed by integrating temporal parameters with the given geometrical path considering the velocity and acceleration constraints. In the figures, the unit for position is meters. For angles, it is radians. The dash line denotes the desired trajectory and the solid line denotes the actual trajectory.

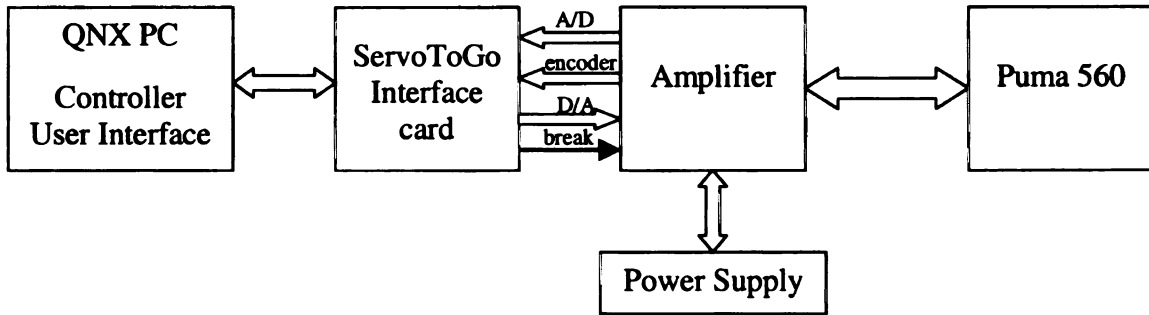


Figure 4.5: Experimental Setup

Figures 4.6 and 4.7 compares the DLS method with the hybrid motion controller when the robot approached its arm singular configurations at $x = 0.9, y = 0.15, z =$

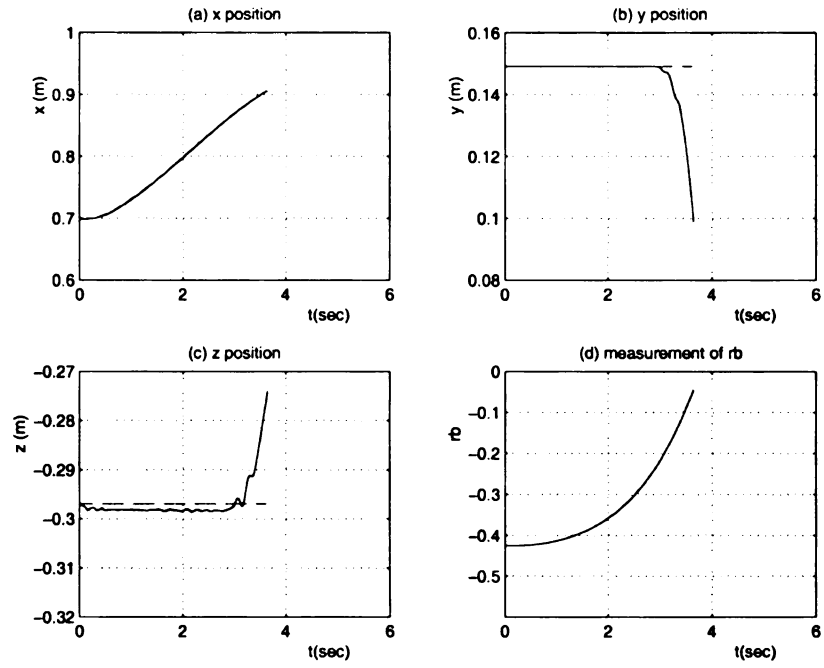


Figure 4.6: At boundary singular configuration, with DLS method only. Dash line: Desired; Solid line: Actual

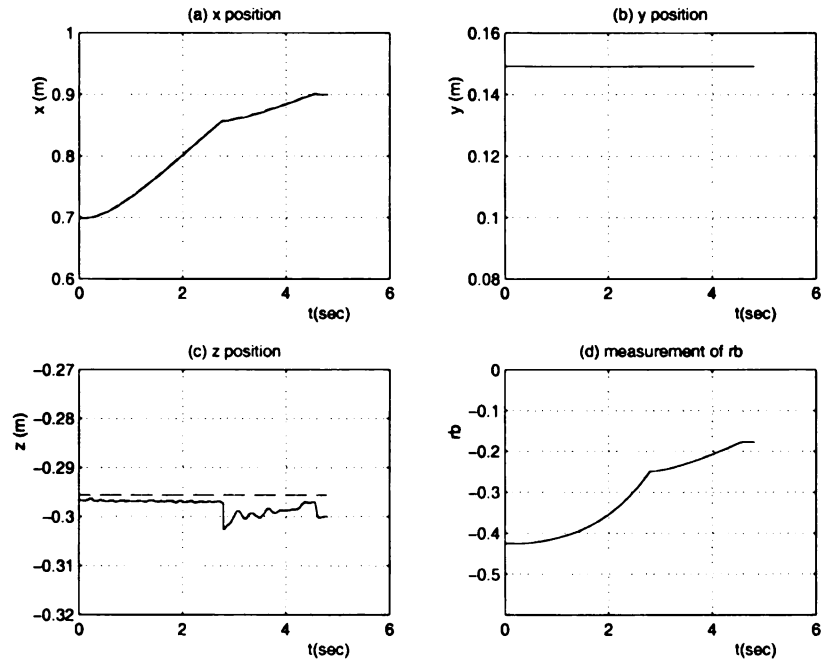


Figure 4.7: At boundary singular configuration, with hybrid motion controller. Dash line: Desired; Solid line: Actual

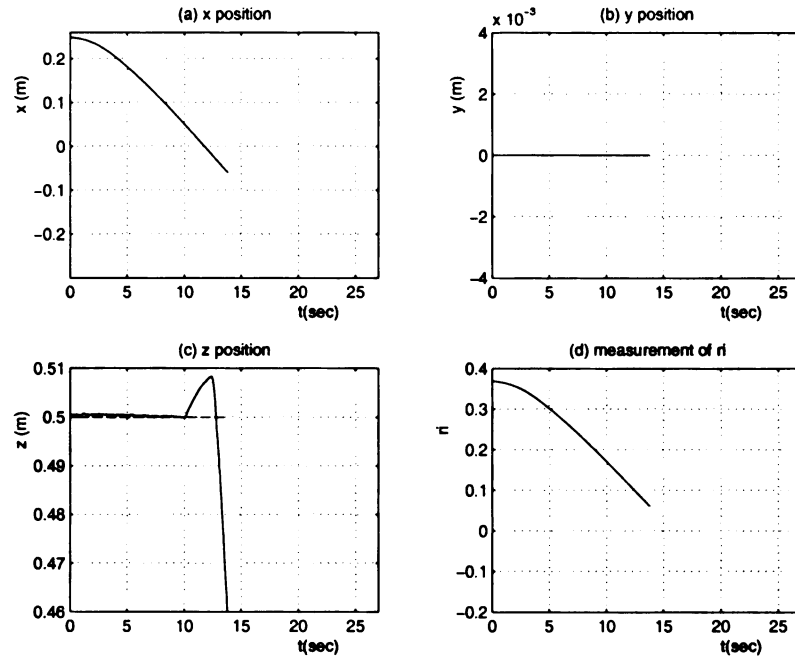


Figure 4.8: At interior singular configuration, with DLS only. Dash line: Desired; Solid line: Actual

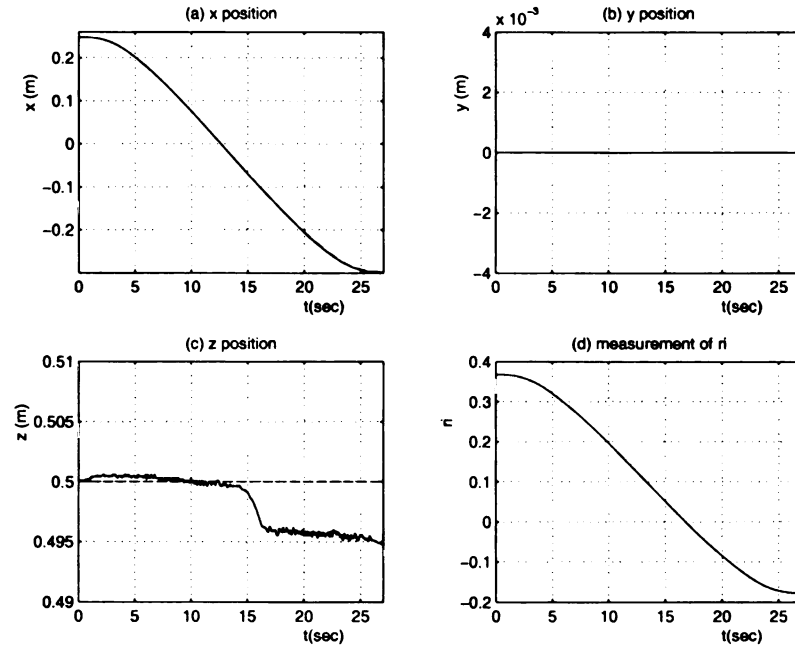


Figure 4.9: At interior singular configuration, with hybrid motion controller. Dash line: Desired; Solid line: Actual

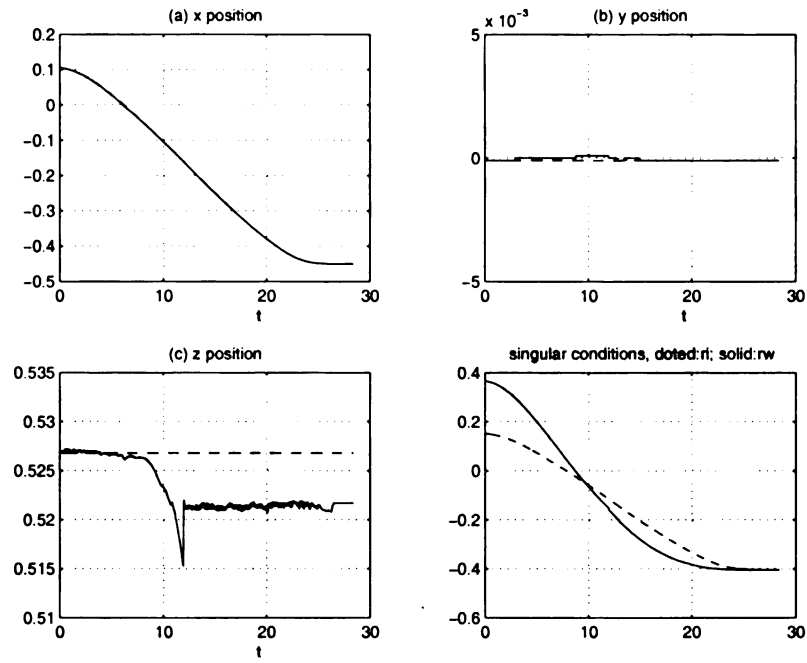


Figure 4.10: With both interior and wrist singular configurations. Dash line: Desired; Solid line: Actual

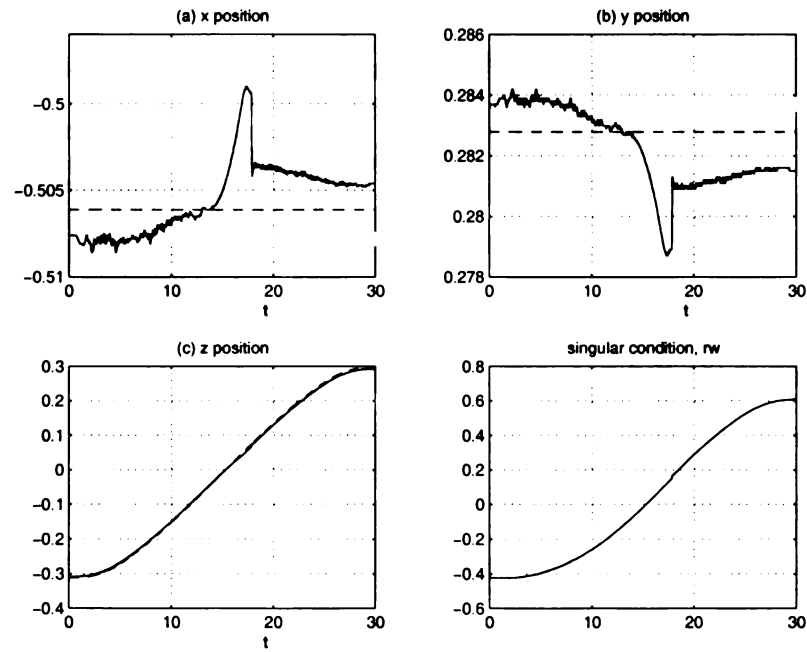


Figure 4.11: With wrist singularity. Dash line: Desired; Solid line: Actual

-0.297 . The task of the robot was to move along x direction from $0.7m$ to $0.9m$. In Figure 4.6, it can be seen that the error in y and z direction became bigger and bigger. In Figure 4.7, the hybrid robot motion controller was used. It can be seen that the tracking errors in y and z directions were much smaller comparing with Figure 4.6 and the task was completed. It is worth noting that the path was replanned in joint space after the controller switch into joint level control. The trajectory was planned in joint space such that the joint motion constraints were satisfied. The end effector is slowed down to avoid unreasonable joint velocities.

The desired trajectory for the next two experiments was $-0.3m$ to $0.25m$ in x direction so that interior singular configuration would be reached. Figure 4.8 shows the trajectory of the robot under task level controller u_3 . When the end-effector entered the interior singular configuration at $x = -0.06, y = 0, z = 0.45$, a very large deviation from the desired path was observed. Figure 4.9 shows the experimental results under the hybrid robot motion controller. Although the deviation from the desired path is still observable, it was smaller as compared to the results in Figure 4.8. The hybrid controller switches into joint level at $t = 14s$, where $\gamma_i = 0.08$; and it switches back to task level at $t = 19s$, where $\gamma_i = -0.08$.

Figure 4.10 shows the position tracking result when both interior and wrist singularities were reached in the motion. At time $t = 8sec$, both interior and wrist singular configurations were reached, the controller switches into joint level. Errors caused by switching at $t = 12sec$ can be seen from Figure 4.10 (a) (b) and (c). This also shows that the hybrid system was stable when double singular configurations were reached.

Figure 4.11 shows that the hybrid motion controller can stably pass the singular configurations with certain speed. In this experiment, x, y, O, A, T were fixed. The end effector position z changed from $-0.3m$ to $0.3m$. The average velocity is $0.2m/s$. The end-effector passed the wrist singular configuration at about $t = 15sec$ and in the middle of the path at certain speed. Small deviations from the desired path can be

seen from the results, but the system was stable. This experiment has demonstrated that the end-effector can pass the singular configurations using the hybrid motion controller.

CHAPTER 5

CONTROL OF MOBILE MANIPULATORS

5.1 Introduction

In Chapter 3 and Chapter 4, control of mobile platforms and singularities of robot arm are discussed. This chapter discusses the control of mobile manipulators, which consists of a mobile platform and a manipulator arm. To take the advantage of large space mobility of the mobile platform and dexterous manipulation of the manipulator, path planning and control of mobile manipulators are discussed. The kinematic and dynamic model of mobile manipulators are derived first. Tracking control and force control are then discussed for holonomic mobile manipulators.

A mobile manipulator usually consists of a mobile platform and a robot arm. Figure 5.1 shows the associated coordinate frames of both the platform and the manipulator, which is the top view and side view of figure 1.1.

- **World frame Σ :** Inertial frame where robot tasks are generally defined,
- **Moving frame Σ_b :** Frame attached on the mobile platform,
- **Virtual moving frame Σ'_b :** Frame attached on the mobile platform, parallel to Σ .

Since the tasks of a mobile manipulator are generally given in task space, *i.e.*, in the world frame, a unified model of the mobile manipulator in the world frame is preferable for control and path planning. For easy reference, all the definitions of the state variables for mobile manipulators, mobile platform, and the robot arm are listed as follows:

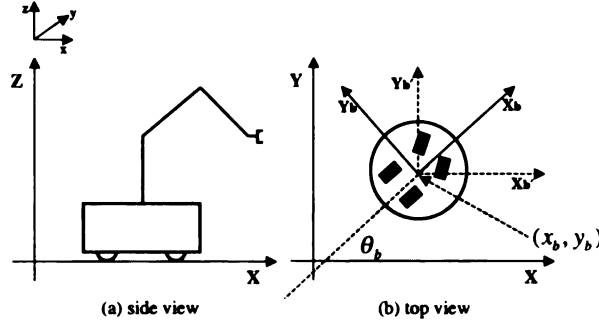


Figure 5.1: Mobile manipulators

- $q = \{q_1, q_2, q_3, q_4, q_5, q_6\}^T$ is the joint angles of the robot arm,
- $Y_b = \{x_b, y_b, \theta_b\}^T$ is the configuration of the mobile platform in the world frame,
- $q' = \{q_1 + \theta_b, q_2, q_3, q_4, q_5, q_6\}^T$ considers the orientation of mobile platform as an addition of the first joint angle of the robot arm,
- $\Theta_b = \{\theta_b, 0, 0, 0, 0, 0\}^T$ is the orientation vector of the mobile platform,
- $p = \{q_1, q_2, q_3, q_4, q_5, q_6, x_b, y_b, \theta_b\}^T$ is defined as the *joint* variables of the mobile manipulator,
- $y_b = \{x_b, y_b, h_b, 0, 0, 0\}^T$ is the origin of Σ_b and Σ'_b where h_b is the height of the mobile robot,
- $y_m = \{p_{xm}, p_{ym}, p_{zm}, O_m, A_m, T_m\}^T$ defines the end effector position and orientation in frame Σ_b ,
- $y'_m = \{p'_{xm}, p'_{ym}, p'_{zm}, O'_m, A'_m, T'_m\}^T$ defines the end effector position and orientation in the virtual moving frame Σ'_b ,
- $y = \{p_x, p_y, p_z, O, A, T\}^T$ defines the end-effector position and orientation in the world frame, which is also the output of the mobile manipulator,
- $f = \{f_x, f_y, f_z\}^T$ defines force applied on the end effector of the mobile manipulator.

5.2 Unified System Model

5.2.1 Kinematic model

From Figure 5.1, it can be seen that the translation that relates the virtual moving frame Σ'_b and the moving frame Σ_b can be defined by $T^b_{b'}$

$$T^b_{b'} = \begin{bmatrix} c_b & -s_b & 0 & 0 \\ s_b & c_b & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where c_b and s_b stand for $\cos \theta_b$ and $\sin \theta_b$ respectively. Since only translation is involved between moving from Σ and Σ'_b , it make it easier to derive the kinematics of the arm in frame Σ'_b and then transform it to frame Σ . The kinematics of robot arms in frame Σ_b and the kinematics of mobile platforms in the world frame Σ are used in building the kinematics of mobile manipulators in frame Σ . In moving frame Σ_b , the relation between the end effector and the joint angles of the arm can be defined as

$$y_m = h(q) \quad (5.1)$$

where h is a nonlinear transformation. The velocity and acceleration relationship between the task space and joint space can then be derived.

$$\begin{aligned} \dot{y}_m &= J\dot{q} \\ \ddot{y}_m &= J\ddot{q} + \dot{J}\dot{q} \end{aligned}$$

where Jacobian J denotes the relation between joint space and task space velocities.

The kinematics of the manipulator in frame Σ_b can be transformed into frame Σ'_b using the transformation $T^b_{b'}$. The end-effector in the moving frame Σ'_b can be proved

to be

$$y'_m = h(q')$$

where h is the same as the transformation in (5.1). In Σ'_b , the velocity and acceleration relationship between the task space and joint space can then be derived as:

$$\begin{aligned}\dot{y}'_m &= \bar{J}\dot{q}' \\ \ddot{y}'_m &= \bar{J}\ddot{q}' + \dot{\bar{J}}\dot{q}'\end{aligned}$$

Here \bar{J} denotes the velocity relation between task space and joint space in frame Σ'_b . Since only a translation is involved between the world frame Σ and frame Σ'_b , the end effector position and orientation in the world frame can be expressed as

$$y = y'_b + y'_m.$$

The velocity and acceleration relation in the world frame Σ can then be derived as:

$$\begin{aligned}\dot{y} &= \dot{y}'_b + \bar{J}\dot{q}' \\ \ddot{y} &= \ddot{y}'_b + \bar{J}\ddot{q}' + \dot{\bar{J}}\dot{q}' \\ &= \ddot{y}'_b + \bar{J}\ddot{q}' + \bar{J}\ddot{\Theta}_b + \dot{\bar{J}}\dot{q}'\end{aligned}\tag{5.2}$$

Actually, by relating the task space velocity and the joint space velocity of the

mobile manipulator in matrix form, the first equation in (5.2) can also be written as

$$\begin{aligned} \dot{y} &= \left[\begin{array}{c|ccc} & 1 & 0 & \bar{J}_{11} \\ & 0 & 1 & \bar{J}_{21} \\ & 0 & 0 & \bar{J}_{31} \\ & 0 & 0 & \bar{J}_{41} \\ & 0 & 0 & \bar{J}_{61} \\ & 0 & 0 & \bar{J}_{61} \end{array} \right] \dot{p} \\ &= J_h \dot{p} \end{aligned} \quad (5.3)$$

where J_h is defined as the Jacobian of the mobile manipulator. It can be seen from J_h that the manipulator Jacobian is augmented by a submatrix whose rank is 3 (\bar{J}_{61} equals 1). The Jacobian can be full rank even when \bar{J} is singular. Thus the mobile platform increases the capability of manipulation. It is worth noting that the virtual frame Σ'_b will ease the modelling of the mobile manipulator. The existing models of the robot arm and the mobile platform can be used.

5.2.2 Dynamics Model

The dynamics of the robot arm in frame Σ_b can generally be described by

$$M_1(q)\ddot{q} + c_1(q, \dot{q}) + g_1(q) = \tau_1 \quad (5.4)$$

where τ_1 is the 6×1 vector of applied torques, $M_1(q)$ is the 6×6 positive definite manipulator inertia matrix, $c_1(q, \dot{q})$ is the 6×1 centripetal and coriolis torques, and $g_1(q)$ is the 6×1 vector of gravity term. Substituting (5.2) into the dynamics (5.4), the robot dynamics in the form of world frame variables can be obtained:

$$M_1 \bar{J}^{-1}(\ddot{y} - \ddot{Y}_b - \bar{J}\ddot{\Theta}_b - \dot{\bar{J}}\dot{q}') + c_1 + g_1 = \tau_1. \quad (5.5)$$

Defining the output of the mobile manipulator as $x = \{x_1, x_2\}$, where $x_1 = y, x_2 = Y_b$, this equation can be rewritten in the form of

$$M_1 \bar{J}^{-1} \ddot{x}_1 + N_1 \ddot{x}_2 - M_1 \bar{J}^{-1} \dot{\bar{J}} \dot{q}' + c_1 + g_1 = \tau_1 \quad (5.6)$$

where $N_1 \ddot{x}_2$ combines terms $-M_1 \bar{J}^{-1} \bar{J} \ddot{\Theta}_b$ and $-M_1 \bar{J}^{-1} \ddot{y}_b$. Following the work of Liu and Lewis [43], Yamamoto and Yun [84], Morel and Dubowsky [27], the dynamics of a holonomic mobile platform can be represented by

$$N_2(p) \ddot{x}_1 + M_2(p) \ddot{x}_2 + c_2(p, \dot{p}) = \tau_2$$

where $N_2(p)$ stands for the interaction between the mobile platform and the robot arm, $M_2(p)$ is the inertia matrix of the mobile platform, and $\tau_2 = \{\tau_x, \tau_y, \tau_\theta\}$ is the input torques applied on the mobile platform. The dynamic model of mobile manipulators with holonomic mobile platform can then be described by

$$\begin{bmatrix} M_1 \bar{J}^{-1} & N_1 \\ N_2 & M_2 \end{bmatrix} \cdot \begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix} + \begin{bmatrix} -M_1 \bar{J}^{-1} \dot{\bar{J}} \dot{q}' + c_1 + g_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (5.7)$$

Let

$$\begin{aligned} M &= \begin{bmatrix} M_1 \bar{J}^{-1} & N_1 \\ N_2 & M_2 \end{bmatrix} \\ \tau &= \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \\ c &= \begin{bmatrix} -M_1 \bar{J}^{-1} \dot{\bar{J}} \dot{q}' + c_1 + g_1 \\ c_2 \end{bmatrix} \end{aligned}$$

(5.7) can be written in a concise form as

$$M\ddot{x} + c = \tau. \quad (5.8)$$

It is worth noting that both the kinematic model and the dynamic model is described in a unified coordinate frame. Since the end effector and mobile platform variables are described in unified frame, it will ease the resolution of kinematic redundancy and the force/position control.

5.3 Motion Control and Redundancy Resolution

5.3.1 Tracking Controller

Applying the following feedback control to (5.7)

$$\tau = \begin{bmatrix} M_1 \bar{J}^{-1} & N_1 \\ N_2 & M_2 \end{bmatrix} \cdot u + \begin{bmatrix} -M_1 \bar{J}^{-1} \dot{\bar{J}} \dot{q}' + c_1 + g_1 \\ c_2 \end{bmatrix} \quad (5.1)$$

the dynamics model of the mobile manipulator can be simplified and linearized as

$$\ddot{x} = u, \quad (5.2)$$

where $\tau = \{\tau_1, \tau_2\}^T$, the dynamic model of the mobile manipulator can be linearized and decoupled as

$$\ddot{x} = u, \quad (5.3)$$

where the vector $x = \{p_x, p_y, p_z, O, A, T, x_b, y_b, \theta_b\}^T$ is the mobile manipulator output and u is a linear control vector. Eq.(5.2) forms a unified model for the mobile manipulator. It is worth noting that vector x not only contains task level variables y , but

also part of the joint level variable Y_b . Since y is defined in task space, direct control of the end effector position and orientation has been implemented. As a result, the mobile manipulator controller has the advantage of the task level controller. At the same time, the joint level redundancy can also be resolved easily.

Given $x^d = \{p_x^d, p_y^d, p_z^d, O^d, A^d, T^d, x_b^d, y_b^d, \theta_b^d\}^T$ as the desired position and orientation of the mobile manipulator, the linear system feedback for model (5.2) can be designed as:

$$u = \ddot{x}^d + k_d(\dot{x}^d - \dot{x}) + k_p(x^d - x). \quad (5.4)$$

It is easy to prove that the above control algorithm (5.4) is asymptotically stable. However, the task of a mobile manipulator is generally given in the form of $y^d = \{p_x^d, p_y^d, p_z^d, O^d, A^d, T^d\}^T$. The desired mobile robot position and orientation $Y_b^d = \{x_b^d, y_b^d, \theta_b^d\}^T$ should be obtained by considering the mobile manipulator configuration and bandwidth of the mobile platform and the arm. By properly positioning the mobile platform, singular arm configurations can be avoided, *i.e.*, the coordinated motion of the mobile platform and the arm can maximize the capability of manipulation. In the following subsection, we give a coordination scheme considering the robot arm manipulation capability.

5.3.2 Singularity Analysis and Redundancy Resolution

Form (5.1), it can be seen that no matter how the linear controller u is designed, the feasibility of the mobile robot controller (5.1) depends on the existence of matrix \bar{J}^{-1} . From the singularity analysis of the manipulator, the following singular conditions can

be obtained [65]:

$$\begin{aligned}\gamma_b &= |d_4 c_3 - a_3 s_3| \\ \gamma_i &= |d_4 s_{23} + a_2 c_2 + a_3 c_{23}| \\ \gamma_w &= |-s_5|\end{aligned}\tag{5.5}$$

Here $\gamma_b = 0, \gamma_i = 0, \gamma_w = 0$ stand for certain arm singular configurations. Here d_4, a_2, a_3, a_4 are robot arm parameters and s_i, c_i stands for $\sin q_i$ and $\cos q_i$ respectively. When the robot arm approaching a singular configuration, the value of at least one of the three expressions will become smaller. If the manipulator is put on a fixed platform, singular configurations can not be avoided for some tasks. However, singular configurations of the arm can be avoided by appropriately positioning the mobile platform for a mobile manipulator. Given a path for the end effector of the mobile manipulator, a path for the mobile platform should be calculated online in such a way that manipulator is always in a dexterous configuration, *i.e.*, not near a singularity. The desired mobile platform path is therefore dependent on the system output. An event based planning approach [79], as shown in Figure 5.2, is used for online coordination control of the mobile platform and the manipulator.

The basic idea of event based planning is to introduce a new motion reference variable, s , which is different from time and directly related to the sensory measurement of system output. The motion information of both the arm and the manipulator is passed to the planner by the motion reference s . For coordinated control of multiple robots, the planner of each robot should take the common motion reference s as an input. Thus the robots are coordinated by a common motion reference.

The objective of coordinated control can be expressed by maximizing the ability of manipulation, which also means maximize the values of the expressions given by (5.5) and taking advantage of the high bandwidth of the manipulator. However, it is not necessary to keep the capability of manipulation at its maximum value. If the

conditions in (5.5) are always maximized, the manipulator will be fixed at a certain configuration and lose its dexterous manipulation capability. As long as the values of $\gamma_b, \gamma_i, \gamma_w$ are greater than certain prespecified minimum values, the manipulator will certainly have the ability to move. The desired position and orientation of the platform should be determined by the trajectory of the end effector and the conditions given by (5.5), which are also the system output. Define the motion of the end

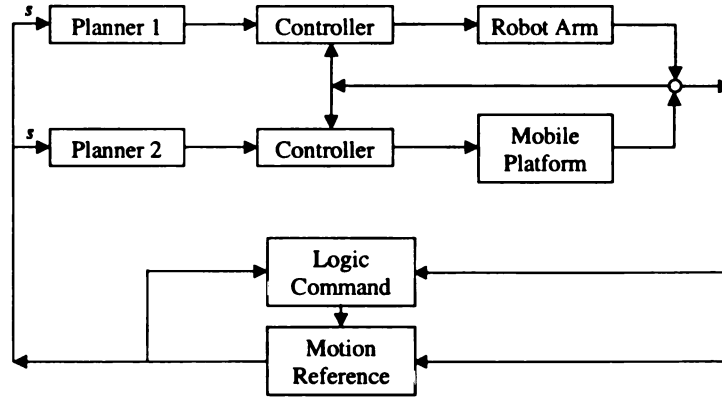


Figure 5.2: Event-based coordination between the arm and the mobile platform

effector as the motion reference, the desired trajectory of the mobile platform is a smooth function of s and $\gamma_b, \gamma_i, \gamma_w$.

$$\begin{aligned}
 x_b &= g_x(s, \gamma_b, \gamma_i, \gamma_w) \\
 y_b &= g_y(s, \gamma_b, \gamma_i, \gamma_w) \\
 \theta_b &= g_\theta(s, \gamma_b, \gamma_i, \gamma_w).
 \end{aligned} \tag{5.6}$$

Functions g_x, g_y, g_θ are continuous and piecewise differentiable. Here the motion reference s can represent the motion of both the platform and the manipulator, *i.e.*, a function of the augmented output x . Let g represent g_x, g_y , or g_θ and γ represent

γ_b, γ_i or γ_w , the requirement of the definition of coordination can be represented by

$$\begin{aligned} \frac{\partial g}{\partial s} &= 0 \quad \text{if } |\gamma| > \gamma_{min}, \text{ or } \frac{\partial \gamma}{\partial s} > 0 \\ \frac{\partial g}{\partial s} &> 0 \quad \text{if } |\gamma| \leq \gamma_{min} \text{ and } \frac{\partial \gamma}{\partial s} < 0 \end{aligned}$$

where γ_{min} stands for the allowable minimum value of γ . If the value of γ is greater than γ_{min} or it is increasing with respect to s , the platform can choose to keep the current position. The mobile platform should be commanded to a position such that the value of γ will increase. An example implementation of g considering only the arm singular configuration γ_b is shown here:

$$\begin{aligned} x_b^d &= x_b + (x_1^d(s) - x_1)r \\ y_b^d &= y_b + (x_2^d(s) - x_2)r \\ \theta_b^d &= \theta_b + \delta r, \end{aligned}$$

where $\delta = \text{atan2}(x_2 - y_b, x_1 - x_b)$ and

$$r^+ = \begin{cases} 0, & \text{if } (|\gamma| > \gamma_{min} \text{ or } \frac{\partial \gamma}{\partial s} > 0) \text{ and } r^- = 0 \\ \alpha_1, & \text{if } |\gamma| > \gamma_{min} \text{ and } r^- > 0 \\ \alpha_2, & \text{if } |\gamma| \leq \gamma_{min} \text{ and } \frac{\partial \gamma}{\partial s} < 0. \end{cases}$$

Here r is designed as a hybrid system [65] and it is piecewise continuous from the right, r^- and r^+ stand for the values of r before and after a time instant, $\alpha_2 > \alpha_1 > 0$ are constants. The values of α_2 and $\alpha_1 > 0$ are obtained adaptively in the experiment. r can take more discrete values in implementation. By this design, the end-effector will go into certain range of γ no matter what the initial value of γ is. Using this strategy, coordination can be achieved even when obstacles block one of them. When the end effector is blocked by obstacles but the mobile platform is free to move, s stops evolving and maintains a constant value. The desired trajectory of the mobile

platform will therefore not change and the whole system is stable. If the mobile platform is blocked but the arm is not, the arm can still continue performing its task without the help of the platform. If now a singular configuration is reached, the desired trajectory of both the arm and the platform will remain constant. Thus the system stability is not affected by obstacles or singularities.

5.4 Experimental Implementation and Results

5.4.1 *Hardware and Software Implementation*

The motion planning and control approaches for a mobile manipulator have been implemented on a mobile manipulator consisting of a Nomadic XR4000 mobile robot and a Puma560 robot arm, as shown in Figure 1.3. For the mobile manipulator, there are two PCs in the mobile platform, one uses Linux as the operating system and runs the mobile robot control software and the other uses a real time operating system QNX and runs the Puma 560 control software. The two computers are connected via an Ethernet connection and communicate at a frequency of 300-500Hz. The sampling period for the Puma 560 control software is 1ms.

The hardware structure is shown in Figure 5.3. The control softwares was developed for the experiments in this chapter and chapter 6. The software components for the experiments are listed as follows and a software structure is shown in Figure 5.4:

- Non-time based mobile robot motion control software
- Singularity-free robot arm motion control software
- Mobile manipulator motion control and force control software

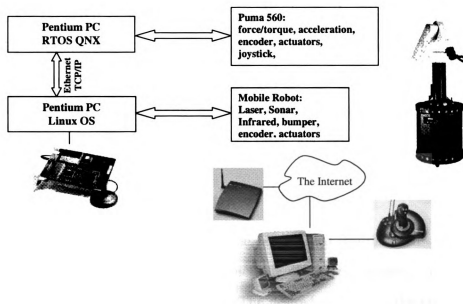


Figure 5.3: Hardware structure of the mobile manipulator

5.4.2 Path Tracking and Tele-operation

In Figure 5.5 and 5.6, the results of path tracking are presented. A sequence of tasks are given in the form of desired end effector position. In Figure 5.7 and 5.8, the results of tele-operation are recorded. The desired end-effector position is generated by Microsoft Joystick connected to a remote PC and sent to robot controller via wireless Internet.

Plots (a) and (b) of figures 5.5-5.8 show the end-effector position in the world frame. The solid curve is the end-effector position in world frame and the dash line is the mobile platform position in the world frame Σ . The end-effector position in the moving frame Σ_b is shown in plots (c) of figures 5.5-5.8. The solid curve is x_m and the dash line is y_m . The measurement of the capability of manipulation is shown in plots (d) of figures 5.5-5.8. Its maximum value is 4.4. For a sequence of tasks, as shown in Figure 5.5 and 5.6, the capability of manipulation is kept at certain value, which is larger than 0.4. The trajectory of the mobile platform is given on line according to the measurement of the capability of manipulation and the end-effector

output. For tasks not known *a priori*, as shown in Figure 5.7 and 5.8 (a)(b)(c), the capability of manipulation can adapt to a higher value, as shown in Figure 5.8(d). This phenomena demonstrates that the robot arm is kept in a dexterous workspace during the operation and this kinematic redundancy resolution scheme is suitable for a point-to-point task, a sequence of multiple tasks, and task unknown *a priori*.

5.4.3 Path Tracking with Unexpected Obstacles

To test the path tracking controller and the event based coordination scheme when tasks are not known *a priori* and unpredictable obstacle is present, the results of teleoperation are presented in Figure 5.9. The desired position of the end effector is generated by a joystick. Figures 5.9 (a) (b) depict the end effector position in x and y direction of the world frame respectively. Figure 5.9 (c) shows the position of the platform x_b and y_b . The operation can be divided into four parts by $t_1 = 15sec$, $t_2 = 44.4sec$ and $t_3 = 54.9sec$, as shown in the figure. Before t_1 , the mobile manipulator operates normally. At time t_1 , the platform meets a unexpected obstacle and can not move any more, as shown in Figure 5.9 (c). The tasks assigned to the mobile manipulator can still be performed by the robot arm without the help of platform. It can be seen that the value of manipulation capability measurement becomes very small around time instance 38sec. At t_2 , the obstacle is removed, the platform begins to cooperate and the manipulation capability adapts to a higher value. At time instance t_3 , the platform is blocked again, the robot arm perform the task alone and the mobile manipulator stops evolving after the manipulation capability drops to a very low value. By the event-based coordination scheme and path tracking controller based on the unified model, the mobile manipulator system is stable even when unexpected obstacles occurred. capability is especially useful when the mobile manipulator works in a highly unstructured environment. This experiment also shows that the kinematic redundancy is resolved online and the manipulation capability is

optimized during the operation.

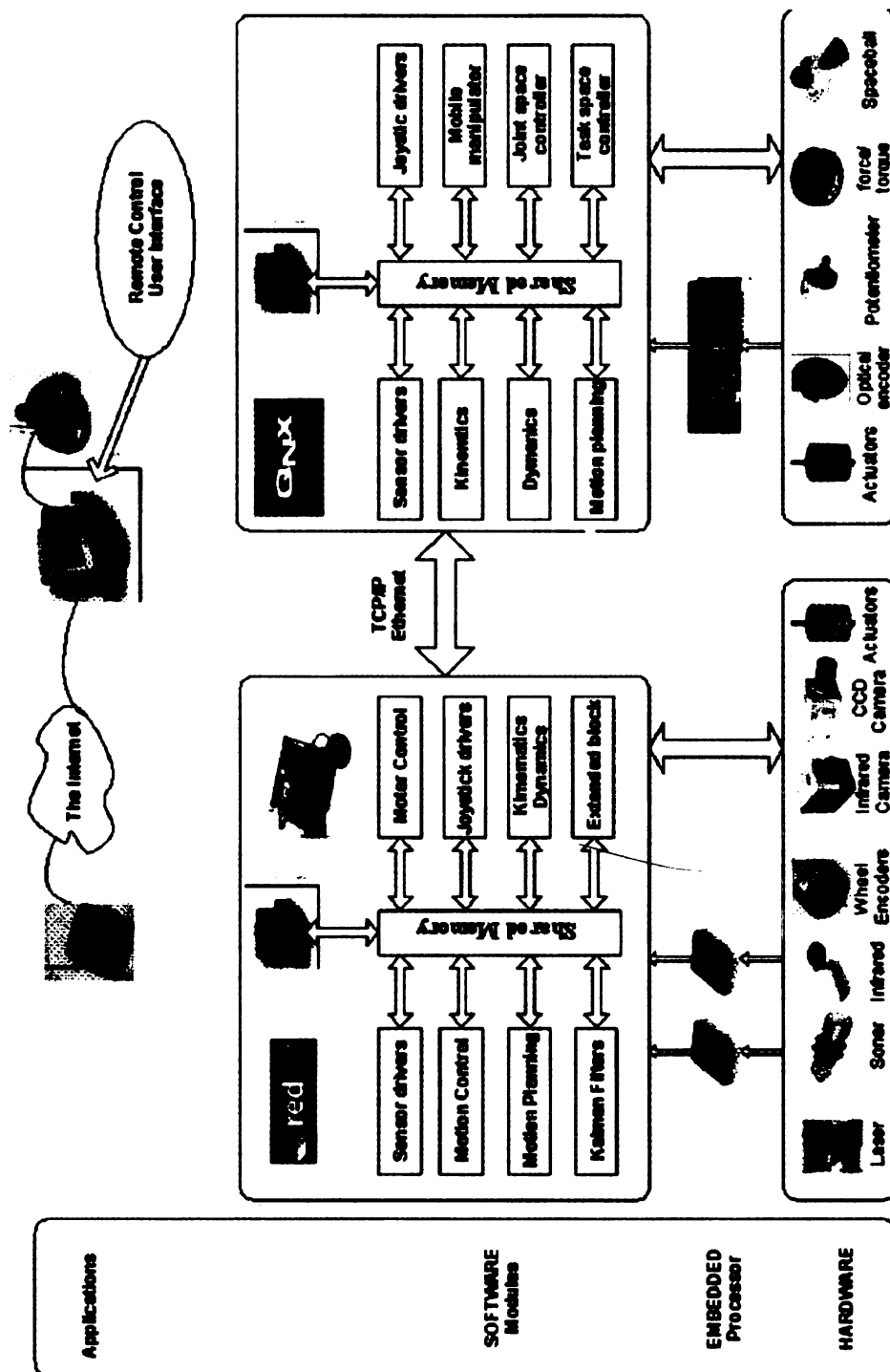


Figure 5.4: Software structure of the mobile manipulator

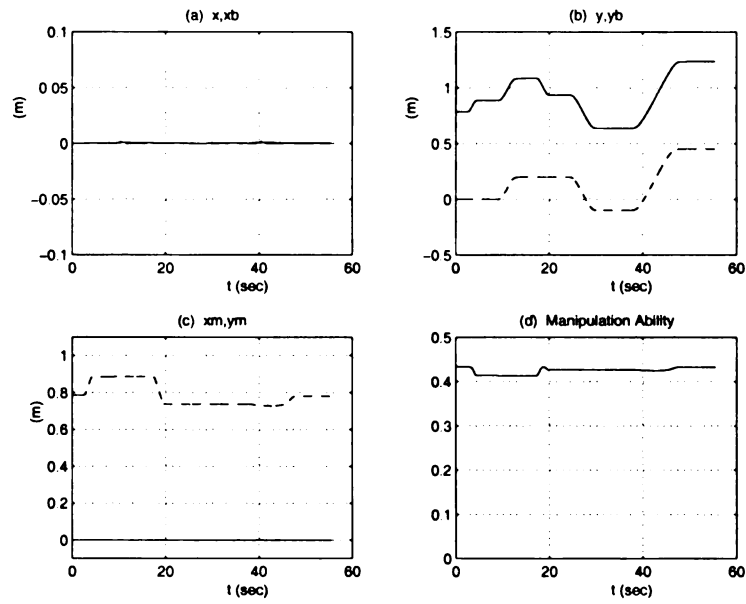


Figure 5.5: Multiple tasks path tracking

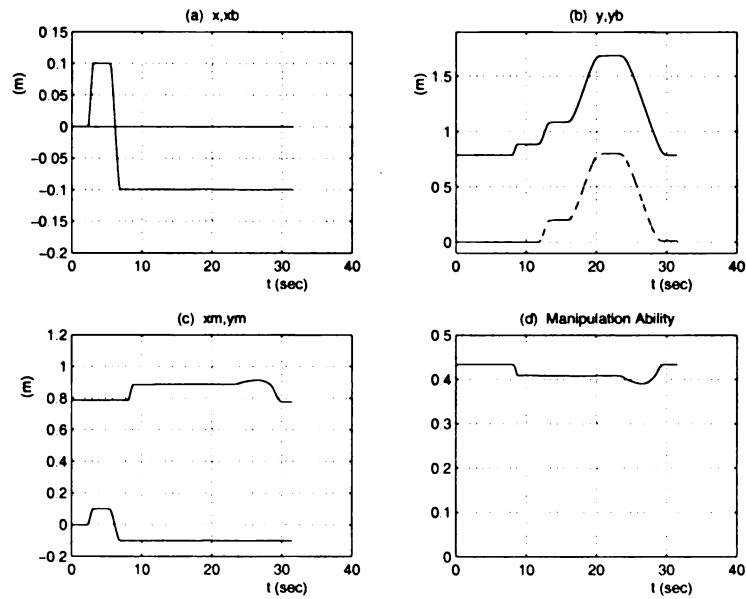


Figure 5.6: Multiple tasks path tracking

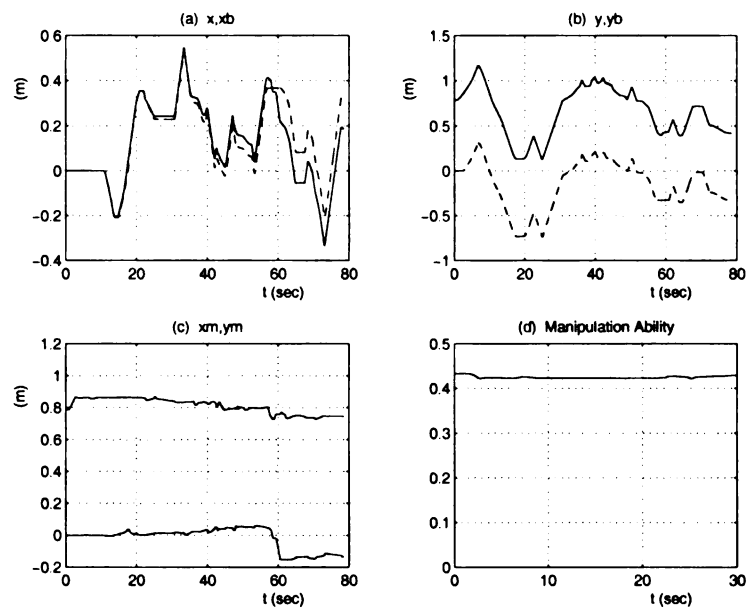


Figure 5.7: Tele-operation

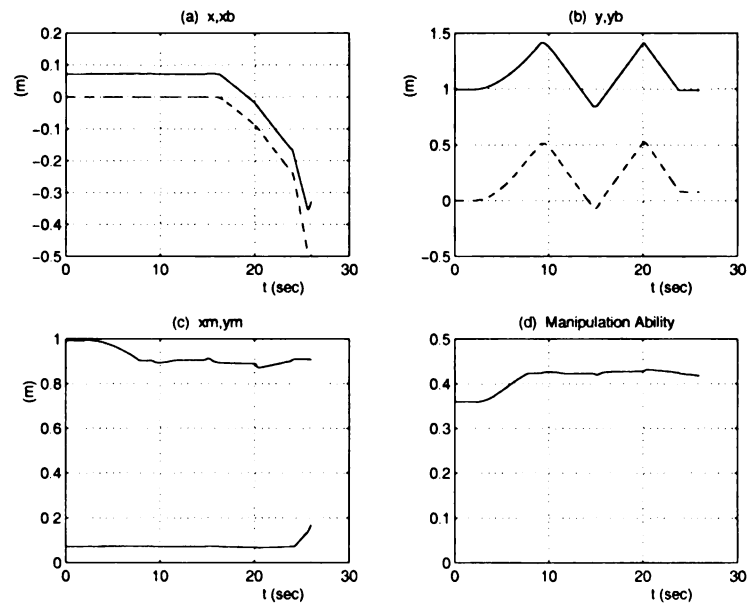


Figure 5.8: Tele-operation

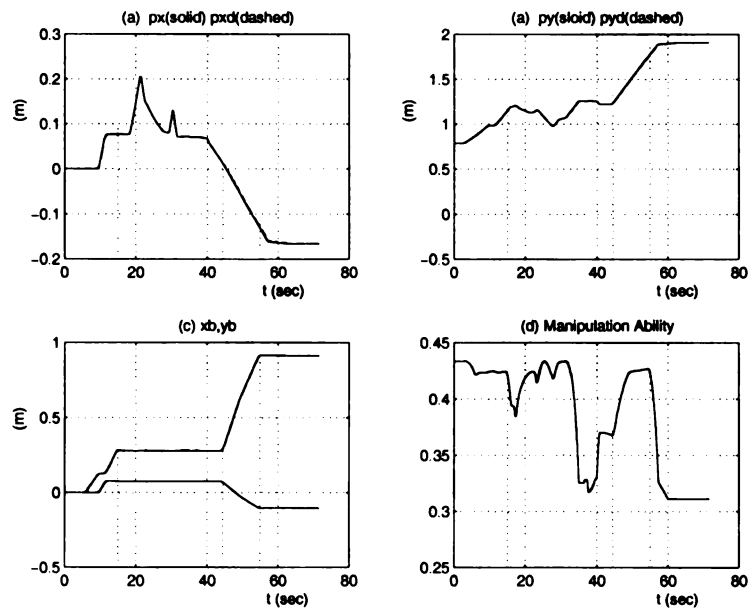


Figure 5.9: Teleoperation with unexpected obstacles

CHAPTER 6

INTEGRATED SENSING AND CONTROL OF MOBILE MANIPULATOR

6.1 Introduction and Force Control Schemes Revisit

Besides trajectory tracking, object manipulation is another kind of task for a mobile manipulator. The mobile manipulator needs to interact with the environment, and output force control is necessary in this case. There are generally two ways for the end effector to interact with the environment. One is interacting with static environments or objects, such as force tracking along hard surface. Force control schemes such as hybrid force/position control, impedance force control, explicit force control and many others have been proposed in the context of fixed base manipulator. The force control schemes are highly dependent on the environments of the robot. In many cases, the environment is assumed as static and the directions for force control and position control are separated. The end effector can also interact with dynamic environments, such as interacting with moving objects. For some objects that are too heavy or large for the mobile manipulator to carry, pushing or pulling is an alternate choice to manipulate objects. Since the mobile manipulator can provide both motion in large workspace and dexterous manipulation capability, we discuss the force control of the mobile manipulator while pushing an object. This requires the mobile manipulator to provide both output force and motion along the same task direction. In this dissertation, the kinematic redundancy of the mobile manipulator is utilized to decouple the force control loop and motion control loop in the same task direction. To compare with, the force control schemes based on a decoupled nonredundant robot model is revisited first.

Hybrid force/position control was first proposed by Raibert and Craig [55]. The workspace is divided into two orthogonal subspaces as shown in Figure 6.1. A selection

matrix S determines the subspaces for which force or position are to be controlled. The control laws for position and force control can be designed independently to satisfy the different control requirements of force and position. Generally, the force control law is designed to interact with static environment. However, the motion of the environment should also be considered when the robot interacts with a moving object. To improve the performance of the force control law, Schutter [61] proposed an approach to feed forward the object motion parameters such as object velocity \dot{x}_o and acceleration \ddot{x}_o , as shown by the dotted lines in Figure 6.1. The desired output force f^d along the motion can be tracked. However, it can be seen that the motion control along the same task direction is open loop.

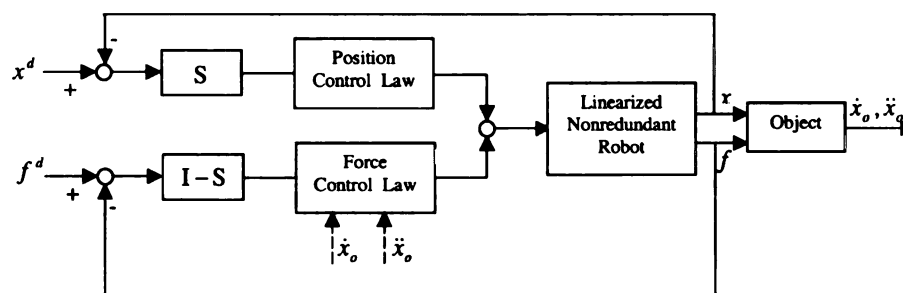


Figure 6.1: Hybrid Force/Position Control

For an object or environment, it is assumed that the end effector position and the contact force with the environment along one DOF can not be controlled independently. The force can then be regulated by controlling the impedance, or compliance of the robot, as shown in Figure 6.2 [29]. The basic idea of this approach is to design a control law which will function in accordance with $f = M\ddot{x} + B\dot{x} + Kx$, where the constant matrices M, B, K represent inertia, damping and stiffness matrices of the interactive system respectively.

Combining the hybrid control and impedance control, hybrid impedance control was proposed [2]. The environment dynamics was also represented in an impedance form and the interaction between the environment and the manipulator was studied.

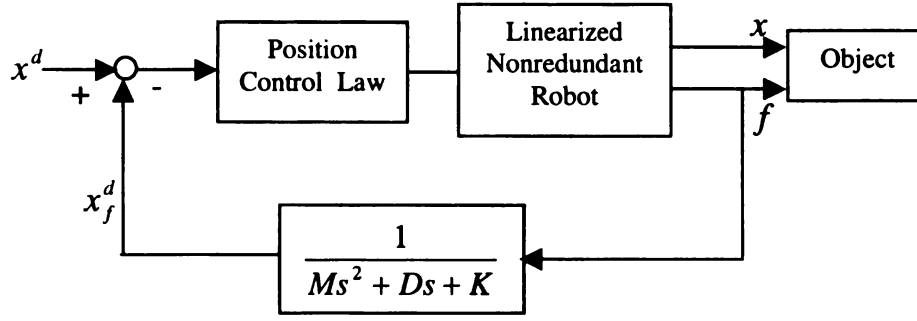


Figure 6.2: Impedance Control

Since the robot may encounter different environments for various applications, control gains of the robot should be tuned in accordance with the environmental characteristics. This scheme also has a slow response to force perturbations and the performance of the implicit force control is restricted by the bandwidth of the position controller [74].

6.2 Output Force/Position Control of Redundant Robots

For a nonredundant robot arm, the directions for force control and position control have to be orthogonal[36], as shown in Figures 6.1. Therefore the force and the position can not be controlled along the same task direction. It is caused by the equal number of control input and desired system outputs. For the hybrid force/position control scheme, the force and position control directions are generally separated by a selection matrix S . For many cases of the impedance control scheme, the control law is essentially a modified position controller. And the performance of the implicit force control is restricted by the bandwidth of the position controller. However, the force control loop and position control loop of a redundant robot can be decoupled along the same task direction. Since the force control is immensely environmental related, the hybrid force/position control of a redundant robot is discussed from two aspects, the capability of the robot and the constraints the environment.

Due to the kinematic redundancy, there are more control inputs than the desired

outputs. For the decoupled unified system model (5.3) of the mobile manipulator, the linear control input is a 9×1 vector, while the desired end effector position and orientation, $\{p_x^d, p_y^d, p_z^d, O^d, A^d, T^d\}^T$, is only a 6×1 vector. The redundant degrees of freedom, which correspond the extra linear control inputs, can be utilized to accomplish secondary tasks. For a path tracking task, the redundant degree of freedom can be used to position the mobile platform such that singular configurations of the arm are avoided. For a task to interact with the environment, output force of the end effector can be chosen as secondary tasks. For instance, the desired output position and force of the mobile manipulator can be chosen as $\{f_x^d, f_y^d, f_z^d, O^d, A^d, T^d, p_x^d, p_y^d, \theta_b^d\}^T$, where f_x^d, f_y^d and f_z^d are the desired output forces. As shown in Figure 6.3, a selection matrix S is not necessary for the redundant robot. It is worth noting that along x direction of the world frame Σ , both desired position and force, p_x^d and f_x^d , are chosen. p_y^d and f_y^d are chosen simultaneously in y direction.

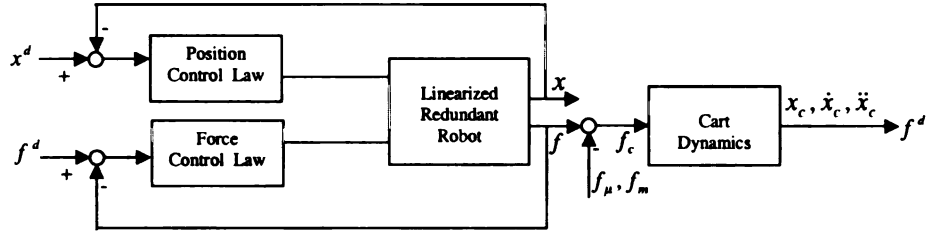


Figure 6.3: Hybrid Force/Position Control for Redundant Robots

Since system (5.3) is decoupled, it can be divided into two subsystems, position control subsystem and force control subsystem. The state variable space of the position control subsystem, x_p , is a subspace of the state space x of system (5.3). Let $x_p = \{p_x, p_y, O, A, T, \theta_b\}^T$ and denote its corresponding linear control input by u_p . The force control subspace x_f is chosen as $x_f = \{x_b, y_b, p_z\}^T$ and the corresponding linear control input is denoted by u_f . System (5.3) can therefore be rewritten into

two subsystems:

$$\begin{aligned}\ddot{x}_p &= u_p \\ \ddot{x}_f &= u_f\end{aligned}\tag{6.1}$$

The linear feedbacks for the two subsystems can be designed as

$$\begin{aligned}u_p &= k_{pp}(x_p^d - x_p) + k_{pd}(\dot{x}_p^d - \dot{x}_p) + \ddot{x}_p^d \\ u_f &= \ddot{x}_f + k_{fp}(f^d - f) + k_{fi} \int_0^t (f^d(\sigma) - f(\sigma)) d\sigma\end{aligned}\tag{6.2}$$

In the controller (6.2), the force control loop and position control loop along the same task direction are decoupled due to the redundancy of the control inputs. And explicit force/position control of the end effector can be designed.

From equation (6.2), it seems that the desired position and force along the same task direction can be tracked simultaneously. This is only mathematically true. Actually, the environment plays an important role in the force control schemes. The desired output forces f_x^d and f_y^d in (6.2) are directly related to the environment dynamics. For instance, zero force output should be commanded in the free space. For an object, the force and position is related by object dynamics and the friction. The dynamics of the object and friction determine the desired output force and the output position of the end effector. However, force and position do not have to be related to the mass of the object only. For some tasks, the force and position can be planned separately in the same task directions[66]. For instance, while a robot cooperates with humans, the output force of a human, f_m , is independent of the robot. This kind of environment can be considered as non-passive environments. In multi-robot coordination tasks, more than one robot can share a task. The robots can be considered working in a non-passive environment. The output force for individual robots can be considered independently from its motion, as long as the composed force satisfies the constraints of the environment dynamics. In brief, the environments will determine the desired

output force and position, and force and position can be planned independently for certain applications, specially if the environment is non-passive or dynamic[66]. The cart pushing task is a typical example of the non-passive or dynamic environments since the payload and friction of the cart are unknown. In this case, the advantage of decoupled force and position control lies in that the interacting force with the dynamic environment can then be regulated explicitly by considering the environmental dynamics, as shown in Figure 6.3. The force does not have to be regulated implicitly as it is done in the implicit force control scheme. This ensures the bandwidth of the force control loop, and ensures that the force planning for manipulating the cart can be tracked.

6.3 Integrated Task Planning and Control

Due to the large workspace and dexterous manipulation capability, mobile manipulators can be used in a variety of applications. For some tasks like painting and soldering, only the motion planning and control of the mobile manipulator are involved. For some complicated tasks, such as pushing a nonholonomic cart, the mobile manipulator maneuvers the object by the output forces. Therefore, the task planning for manipulating a cart involves three issues: the motion and force planning of the cart, the trajectory planning and control of the mobile manipulator, synchronization of the motion planners for the cart and the mobile manipulator.

Figure 6.4 shows a flow chart for the integrated task planning and control for pushing a nonholonomic cart. First, for a given task, the geometrical path needs to be determined considering the nonholonomic constraint of the cart. Secondly, a trajectory should be planned based on the geometrical path, and the input force for the cart to track the trajectory should be planned. Thirdly, the trajectory planning and coordination of the mobile manipulator [66] should be considered. Finally, the output force control of the mobile manipulator and task planning should be integrated

based on the decoupled system model (5.3).

In the cart pushing task, the mobile manipulator and the cart compose a system. Therefore, the trajectory planning of the mobile manipulator and the cart should be synchronized. The event-based approach for the path planning and control of multirobot coordination [82] can be used in integrating the planning and control of the pushing task. A common motion reference s is chosen for the trajectory planning of both the mobile manipulator and the cart. In other words, the evolution of the cart trajectory and the mobile manipulator trajectory is based on the same motion reference. It is worth noting that the common motion reference s is computed based on sensory information of the mobile manipulator.

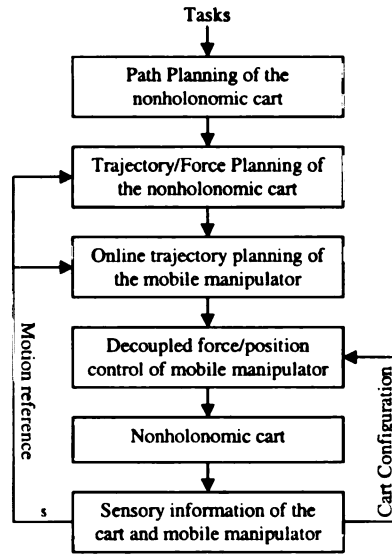


Figure 6.4: Integrated Task Planning and Control for the Mobile Manipulation of a Nonholonomic Cart

6.3.1 Motion Planning of the Cart

The cart shown in Figure 6.5(b) is a nonholonomic system. The path planning for a nonholonomic system is to find a path connecting specified configurations that also satisfy nonholonomic constraints. The cart model is similar to a nonholonomic mobile robot except that the cart is a passive object. Therefore, many path planning

algorithms for a nonholonomic system such as steering using sinusoids [45] and goursat normal form approach [70] can be used for the motion planning of the cart. In this dissertation, the kinematic model is transformed into the chained form and the motion planning is considered as the solution of an optimal control problem. Considering the kinematic model of the cart as

$$\dot{x}_c = v_1 \cos \theta_c \quad (6.3)$$

$$\dot{y}_c = v_1 \sin \theta_c \quad (6.4)$$

$$\dot{\theta}_c = v_2 \quad (6.5)$$

$$\dot{x}_c \sin \theta_c - \dot{y}_c \cos \theta_c = 0, \quad (6.6)$$

where v_1 and v_2 are the forward velocity and the angular velocity of the cart respectively. An optimal trajectory connecting an initial configuration and a final configuration can be obtained by minimizing the control input energy $J = \int_{t_0}^{t_f} \frac{1}{2} \|v(t)\|^2 dt$. Given a geometric path, an event-based approach can be used to determine the trajectory [82].

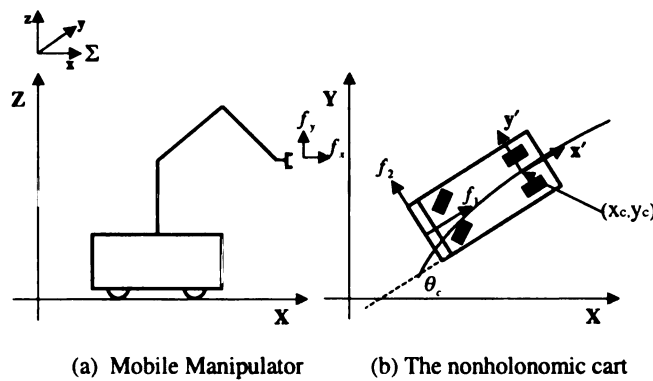


Figure 6.5: A mobile manipulator and a nonholonomic cart

6.3.2 Force Planning of the Cart

The nonholonomic cart shown in figure 6.5(b) is a passive object. Assuming that the force applied on the cart can be decomposed into f_1 and f_2 , the dynamic model of the nonholonomic cart in world frame Σ can be represented by

$$\begin{aligned}\ddot{x}_c &= \frac{\lambda}{m_c} \sin \theta_c + \frac{f_1}{m_c} \cos \theta_c \\ \ddot{y}_c &= -\frac{\lambda}{m_c} \cos \theta_c + \frac{f_1}{m_c} \sin \theta_c \\ \ddot{\theta}_c &= \frac{f_2}{I_c}\end{aligned}\tag{6.7}$$

where x_c, y_c and θ_c are the configuration of the cart, m_c and I_c are the mass and inertia of the cart. λ is a Lagrange multiplier and θ_c is the cart orientation.

As shown in figure 6.5, The input force applied onto the cart, f_1 and f_2 , corresponds the desired output force of the end effector, f_x^d, f_y^d . In other words, the mobile manipulator controls the cart to achieve certain tasks by its output force. Therefore, manipulating the cart requires the motion planning and control of the mobile manipulator based on the decoupled model (5.3), and the motion and force planning of the cart based on its dynamic model (6.7).

To track a given trajectory, the input forces applied onto the cart, f_1 and f_2 , need also be planned. The input force planning of the cart is equivalent to the output force planning of the mobile manipulator. The control input of the nonholonomic cart is determined by its dynamic model (6.7), the nonholonomic constraint (6.6) and its desired trajectory $\{x_c^d, y_c^d, \theta_c^d\}$.

Due to the lack of a continuous time-invariant stabilizing feedback[9], output stabilization is considered in this dissertation. Choosing a manifold rather than a particular configuration as the desired system output, the system can be input-output linearized. By choosing x_c, y_c as the system output, the system can be linearized with respect to the control input f_1 and v_2 . This can be explained by the following

derivations. From equation (6.3) and (6.4), it is easy to see that $v_1 = \dot{x}_c \cos \theta_c + \dot{y}_c \sin \theta_c$. Here v_1 is actually the forward velocity along x' direction. Considering the velocity along y' direction is $\dot{x}_c \sin \theta_c - \dot{y}_c \cos \theta_c = 0$, the following relation can be obtained:

$$\begin{aligned}\dot{v}_1 &= \ddot{x}_c \cos \theta_c + \ddot{y}_c \sin \theta_c \\ \dot{v}_2 &= \ddot{\theta}_c\end{aligned}$$

Suppose the desired output of interest is $\{x_c, y_c\}$, the following input-output relation can be obtained by the derivative of the equations (6.3,6.4):

$$\begin{aligned}\ddot{x}_c &= \frac{1}{m_c} \cos \theta_c \cdot f_1 - v_1 \sin \theta_c \cdot v_2 \\ \ddot{y}_c &= \frac{1}{m_c} \sin \theta_c \cdot f_1 + v_1 \cos \theta_c \cdot v_2\end{aligned}$$

Considering f_1 and v_2 as the control inputs of the system, the input output can be formulated in a matrix form:

$$\begin{pmatrix} \ddot{x}_c \\ \ddot{y}_c \end{pmatrix} = G \begin{pmatrix} f_1 \\ v_2 \end{pmatrix}$$

where

$$G = \begin{pmatrix} \frac{\cos \theta_c}{m_c} & -v_1 \sin \theta_c \\ \frac{\sin \theta_c}{m_c} & v_1 \cos \theta_c \end{pmatrix}$$

The nonholonomic cart can then be linearized and decoupled as

$$\begin{pmatrix} \ddot{x}_c \\ \ddot{y}_c \end{pmatrix} = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$$

where $\{w_1, w_2\}^T = G\{f_1, v_2\}^T$. Given a desired path of the cart, x_c^d, y_c^d , which satisfies

the nonholonomic constraint, the controller can be designed as

$$\begin{aligned} w_1 &= \ddot{x}_c^d + k_{px}(x_c^d - x_c) + k_{dx}(\dot{x}_c^d - \dot{x}_c) \\ w_2 &= \ddot{y}_c^d + k_{py}(y_c^d - y_c) + k_{dy}(\dot{y}_c^d - \dot{y}_c) \end{aligned}$$

The angular velocity v_2 can then be obtained by $\{f_1, v_2\}^T = G^{-1}\{w_1, w_2\}^T$. The physical meaning of this approach is that the control input f_1 generates the forward motion and v_2 controls the cart orientation such that x_c^d and y_c^d are tracked. However, control input v_2 is an internal state controlled by f_2 , the tangent force applied onto the cart. The control input f_2 can then be computed by the backstepping approach [35] based on the design of v_2 . Defining $v_2 = \phi(x_c, y_c, \theta_c)$ and $z = \dot{\theta}_c - \dot{\phi}$, then equation (6.5) can be transformed into

$$\dot{z} = -\dot{\phi} + \frac{L}{I_c} f_2 \quad (6.8)$$

The control input f_2 can then simply be designed as

$$f_2 = \frac{I_c}{L} (\dot{\phi} - k_\theta(\dot{\theta}_c - \dot{\phi})) \quad (6.9)$$

It is worth noting that the desired cart configuration $\{x_c^d, y_c^d\}$ is the result of trajectory planning. As shown in figures 6.5 (a) and (b), the output force of the mobile manipulator corresponds to the control input force of the cart. Given the force planning f_1 and f_2 , the desired output force of the mobile manipulator would be: $f_x^d = f_1 * \sin \theta_c - f_2 * \cos \theta_c$, $f_y^d = f_1 * \cos \theta_c + f_2 * \sin \theta_c$. To integrate the task planning and control, the decoupled force/position controller 6.2 can be used to track certain trajectories.

6.3.3 Integration of Sensing, Task Planning and Control

To implement the force control of the cart, the actual configuration $\{x_c, y_c, \theta_c\}$ and corresponding velocities need to be estimated. Since the cart has no sensors equipped on it, sensors on the mobile manipulators, such as laser range finder, encoder and force/torque sensor are used to estimate the configuration of the cart. To estimate the orientation of the cart in the moving frame attached on the mobile platform, laser range finder is used. Figure 6.6 shows a configuration of the cart in the moving frame of the mobile platform. The laser sensor provides a polar range map of its environment in the form of (α, r) , where α is a angle from $[-\pi/2, \pi/2]$ which is discretized into 360 units. The range sensor provides a ranging resolution of 50mm. Figure 6.7 (a) shows the raw data from the laser sensor. Obviously the measurement of the cart is mixed with the environment surround it. Only a chunk of the data is useful and the rest should be filtered out. A sliding window, $w = \{\alpha_1, \alpha_2\}$ is defined. The data for $\alpha \notin w$ are discarded. To obtain the size of w , the encoder information of the mobile manipulator should be fused with the laser data. Since the cart is hold by the end effector, the approximate position of x_r, y_r , as shown in Figure 6.6, can be obtained. Based on x_r, y_r and the covariance of the measurement r , which is known, the sliding window size can be estimated. The filtered data by the sliding window is shown in Figure 6.7 (b).

In this applications, the cart position and orientation (x_c, y_c, θ_c) are the parameters to be estimated. Assuming the relative motion between the cart the robot is slow, the cart position and orientation can be estimated by the standard Extended Kalman Filter(EKF) technique[31]. The cart dynamics are described by vector function:

$$\dot{X}_c = F(X_c, f) + \xi \quad (6.10)$$

where f is the control input of the cart, which can be measured by the force torque sen-

sensor equipped on the end effector of the mobile manipulator. $X_c = \{x_c, \dot{x}_c, y_c, \dot{y}_c, \theta_c, \dot{\theta}_c\}$. ξ is random vector with zero mean and covariance matrix σ_x . Considering the output vector as $Z(t) = \{\alpha, r\}^T$, as shown in Figure 6.6, the output function is described as:

$$Z(t) = G(X_c) + \zeta \quad (6.11)$$

where ζ is a random vector with zero mean and covariance matrix σ_z . It is worth noting that nonholonomic constraint should be considered in (6.10) and the measurement (α, r) should be transformed into the moving frame. By applying the standard EKF technique to (6.10) and (6.11), an estimate of the cart state variables \hat{X}_c for X_c can be obtained. The dotted line in Figure 6.7 (b) shows an estimation of the cart configuration in the mobile frame Σ_b . It is worthy noting that both the force/torque sensor information and the laser range finder information are used in the estimation of the cart configuration.

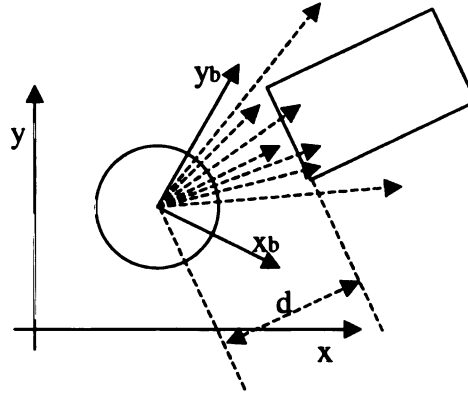


Figure 6.6: Configuration of the cart in the moving frame

To interact with a cart with different weight and length, the cart parameters, m_c, I_c and cart length L can also be estimated based on the force/torque sensor and accelerometer. Details please see [64].

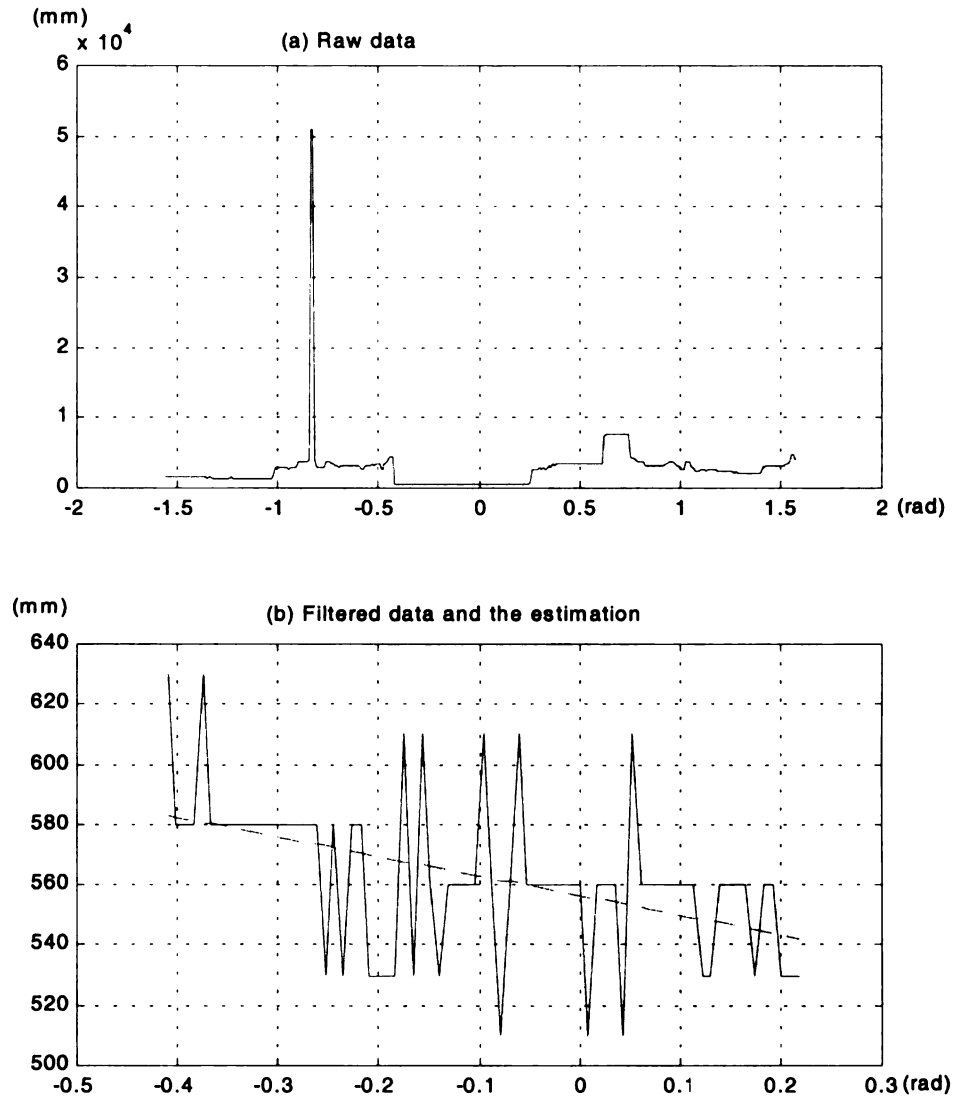


Figure 6.7: Laser range data

6.4 Experimental Implementation and Results

The path planning and control approaches have been implemented on a mobile manipulator consisting of a Nomadic XR4000 mobile robot and a Puma560 robot arm, as shown in Figure 1.3. For the nonholonomic cart, the mass is $45kg$ and the cart length is $0.89m$. The gripper of the mobile manipulator held the handle of the cart during the manipulation.

6.4.1 Pushing Along a Straight Line

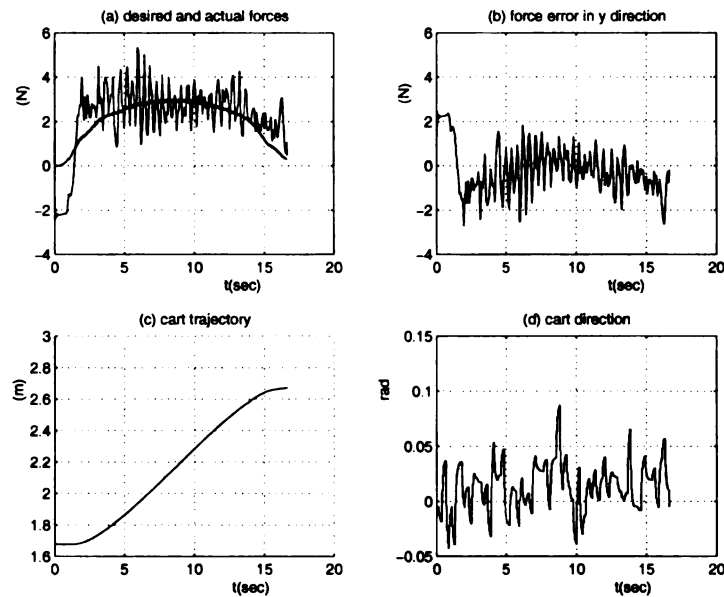


Figure 6.8: Pushing the cart along a straight line

Figure 6.8 shows the results of pushing a cart along a straight line using the mobile manipulator. In this experiment, the task was to push the cart along y direction in frame Σ from $1.675m$ to $2.675m$. Figure 6.8 (a) shows the desired output force and actual output force in y direction. Figure 6.8 (b) recorded the force tracking error. It is worth noting that the static friction force was also overcome in the first two seconds of the task. Figures 6.8 (c)(d) show the trajectory of the cart and the cart orientation respectively. The position and the orientation of the cart were estimated by fusing the sensor information on the mobile manipulator. The laser range sensor, encoders of the mobile platform and the robot arm are used in the estimation. In this experiments, the motion in y direction was chosen as the motion reference to synchronize the motion planning of the cart and the mobile manipulator. This experiments demonstrated that the desired force, the desired trajectory of the cart has been tracked with satisfactory precision.

For the same task, figure 6.9 shows the experimental results when an unexpected

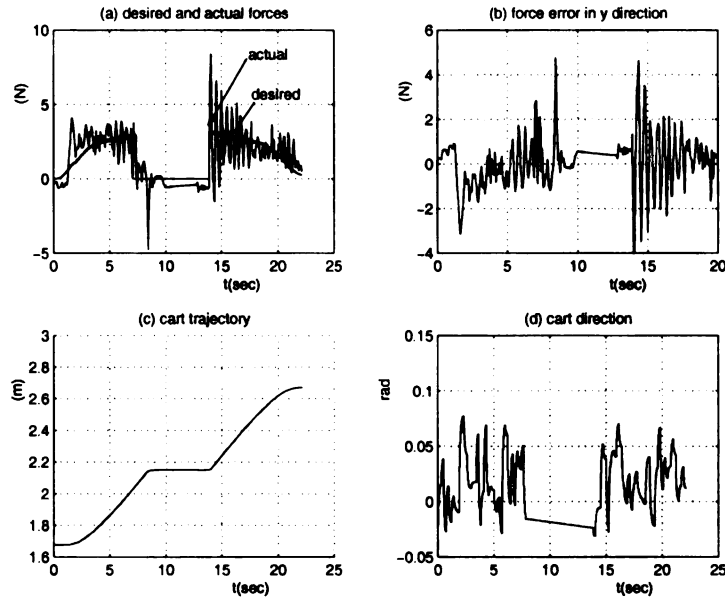


Figure 6.9: Pushing with unexpected obstacle

obstacle blocked the mobile manipulator along the straight line. At about 7.5s, the system is blocked by an obstacle, the motion reference stops evolving. As a result, the desired output force was set to zero and the cart stopped moving. When the obstacle was removed at about 14s, the motion reference started evolving again, the desired output force was computed based on the desired motion of the cart and the pushing task is resumed. It can be seen that the motion of the cart and mobile manipulation system was synchronized by a common motion reference.

6.4.2 Turning the Cart at a Corner

Pushing the cart along a straight line is relatively easy since the desired end effector position and output force is easy to plan. The second experiment considered a complex task. The mobile manipulator first pushed the cart forward 0.4m along y in about 20 seconds, made a turn in about 35 seconds, and then pushed the cart forward again for 0.4m along x direction. Figure 6.10(a) shows the trajectories of the cart, x_c and y_c , and the end effector trajectory, p_x and p_y . Figure 6.10(b) is the cart

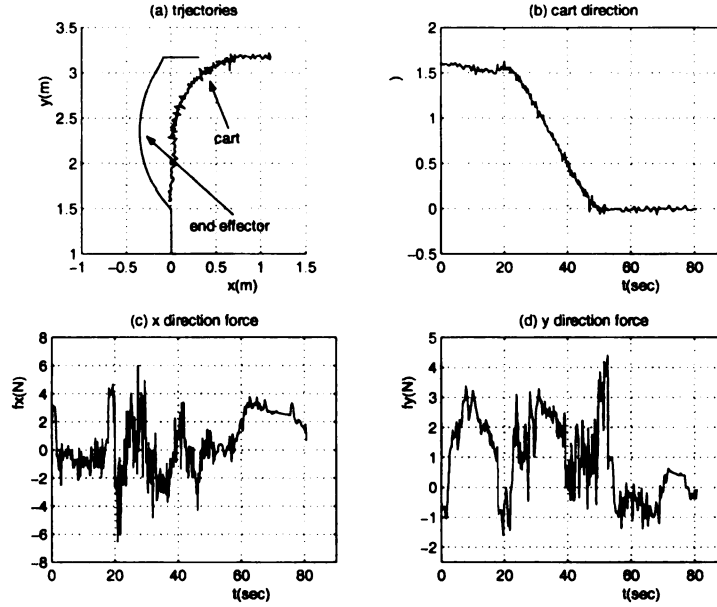


Figure 6.10: Turning the cart at an corner

orientation θ_c with respect to time. The cart started from a configuration parallel to y direction, and turned to a configuration parallel to x direction. The output force is planned based on the desired trajectory of the cart. Figure 6.10 (c)(d) are the force applied onto the cart, f_x and f_y . It is worth noting that the force are recorded in world frame Σ . It is seen that the force f_x pushed the cart along x direction in the last 20 seconds, and f_y pushed the cart along y direction in the first 20 seconds. This experiment has demonstrated that a complex task can be completed by the integrated task planning and control approach.

6.4.3 Pushing Along a Sine Wave

The task in this experiment was to push the cart forward along a sine wave denoted by $x_c = 0.2 \sin(1.8y_c)m$. In this experiment, the force f_1 generates the motion along the sine wave, the force f_2 regulates the cart orientation. Figure 6.11 (a) shows the desired and actual trajectory of the cart, and the desired and actual of the end effector. The noise of the actual trajectory was caused by the noisy estimation of cart

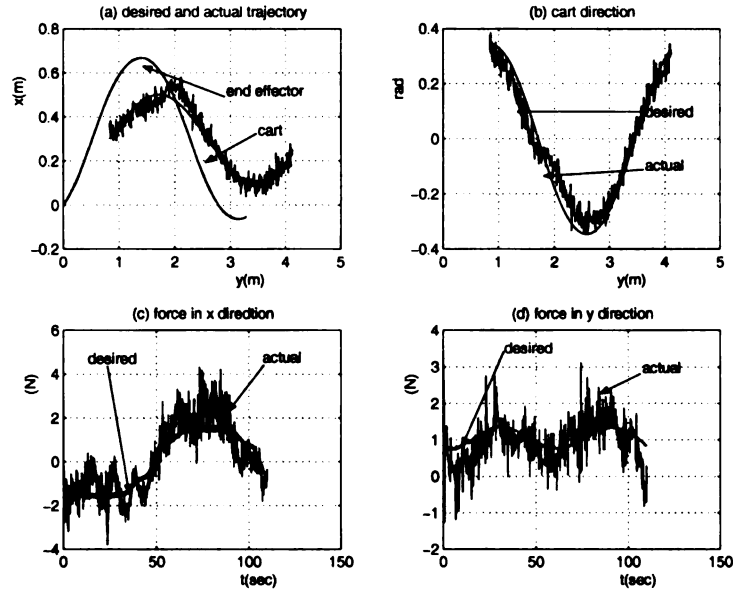


Figure 6.11: Pushing the task along a sine wave

orientation, which is shown in Figures 6.11 (b). Figure 6.11 (c)(d) show the desired and actual force signal in x and y direction respectively. f_1 and f_2 were transformed to frame Σ and compared with the actual force signal in the same frame. It is seen the desired force in both directions were tracked. At $t = 25s$ or $y_c = 1.9m$, the cart slipped on the floor and the cart orientation was changed, the mobile manipulator still can manipulate the cart such that the desired trajectory was tracked. In this experiments, the motion of the cart in y direction, y_c , was chosen as the motion reference. The experiments results has shown that the integrated task planning and control approach enables the mobile manipulator to perform complicated tasks with satisfactory performance. In all the experiments, both the desired cart trajectory and desired output force were tracked.

CHAPTER 7

CONCLUSIONS

The major results of this work can be concluded in four parts, the motion reference project method and its application in mobile robots, robot arm singularity analysis and its application to singularityless tracking control and mobile manipulator coordination, the unified model of mobile manipulators and an online redundancy resolution scheme, force/position control along the same task direction and its application to manipulation of moving objects.

First, a new design method for non-time based tracking controller has been developed. It converts a controller designed using traditional time-based approach to a new non-time based controller. This approach significantly simplifies the design procedure to achieve a non-time based tracking controller. The results provides an efficient and systematic approach to design non-time based tracking controller for general nonlinear dynamical systems. It is an important step in the development of a theoretical foundation for sensor-referenced intelligent control. The non-time based approach is also a key step in design the coordination of the mobile platform and the manipulator arm. In Chapter 3, this method is applied to tracking controllers of nonholonomic mobile platforms. The experimental results have clearly demonstrated the advantage of the non-time based tracking controller in the presence of unexpected obstacles.

The singularity of manipulators and mobile manipulators are studied. For the manipulator arm, a singularity-free robot motion controller which can work at or near singular configurations is presented. Furthermore, the stability of the hybrid controller has been theoretically proven. The primary contribution of the hybrid system approach is that the motion controller can satisfy different constraints in different workspace of the robot. The continuous dynamics have different state variable and the supervisory control is represented in Max-Plus algebra. The stability of such a

switching control system is analyzed by Multiple Lyapunov functions. For the mobile manipulator, the kinematic redundancy is used to avoid singular arm configurations, the redundancy resolution is therefore solved by the coordination scheme of the mobile manipulator.

A unified dynamic model for the mobile manipulator is derived and nonlinear feedback is applied to linearize and decouple this model. The kinematic redundancy and dynamic properties of the platform and the robot arm have been considered in the unified model. Based on this model, mobile manipulator tasks such as path tracking control, force and position control can be easily designed and kinematic redundancy can be used to accomplished secondary tasks, such as controlling force and position along the same direction. An event-based coordination scheme is proposed for the cooperation of the platform and the robot arm. This online approach is suitable for a point-to-point task, tele-operation and a sequence of tasks. The mobile manipulator system is stable even in the case of appearance of unexpected obstacles and the manipulation capability can adapt to a best possible configuration during the operation.

Based on the decoupled and linearized model, this dissertation further discusses the force/position control along the same task direction utilizing the kinematic redundancy of mobile manipulators. For tasks that require both force and position control along the same task direction, such as pushing a nonholonomic cart, an integrated task planning and control for mobile manipulators is proposed. The approach for manipulating the nonholonomic cart integrates the planning and control of the mobile manipulator, the motion planning and force planning of the nonholonomic cart, and the estimation of the cart configuration based on the sensors of the mobile manipulators. The integrated task planning and control enables the robot to interact with moving objects and complete complicated tasks by the output force. The approaches discussed in this dissertation can be used for the motion planning and control of mo-

bile manipulators which can find applications in manufacturing, hospitals, homes and offices. Experimental results have demonstrated the advantage of the integrated task planning and control approach.

BIBLIOGRAPHY

- [1] *Manuals of QNX Operating System*. QNX Software Systems Ltd.
- [2] R. J. Anderson and M. W. Spong. Hybrid impedance control of robotic manipulators. *IEEE Journal on Robotics and Automation*, 4(5):549–556, 1988.
- [3] G. Antonelli, N. Sarkar, and S. Chiaverini. External force control for underwater vehicle manipulator systems. In *Proceedings of IEEE Conference on Decision and Control*, pages 2975 –2980, Phoenix, Arizona, Dec 1999.
- [4] F. L. Baccelli, G. Cohen, G. J. Olsder, and J. Quadrat. *Synchronization and Linearity, an Algebra for Discrete Event Systems*. John Wiley & Sons, 1992.
- [5] N. S. Bedrossian. Classification of singular configurations for redundant manipulators. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 818 –823, 1990.
- [6] A. K. Bejczy and S. Lee. Robot arm dynamic model reduction for control. In *Proceedings of IEEE Conference on Decision and Control*, pages 1466–1476, San Antonio, Texas, 1983.
- [7] B. E. Bishop and M. W. Spong. Control of redundant manipulators using logic-based switching. In *Proceedings of IEEE Conference on Decision and Control*, pages 1488–1493, Tampa, Florida, 1998.
- [8] M. S. Branicky. Multiple lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):475–482, 1998.
- [9] R. W. Brockett. Asymptotic stability and feedback stabilization. *Differential Geometric Control Theory by Brockett, R. W., Millman, R. S., and Sussmann, J. J.*, 27:181–208, 1983.

- [10] L. G. Bushnell, D. M. Tibury, and S. S. Sastry. Steering three-input nonholonomic systems: The fire truck example. *The International Journal of Robotics Research*, 14(4):366–381, 1995.
- [11] F. Caccavale, S. Chiaverini, and B. Siciliano. Second-order kinematic control of robot manipulators with jacobian damped least-squares inverse: Theory and experiments. 2(3):188–194, 1997.
- [12] P. H. Chang, K. C. Park, and S. Lee. An extension to operational space for kinematically redundant manipulators: Kinematics and dynamics. *IEEE Transactions on Robotics and Automation*, 16(5):592–596, 2000.
- [13] M. W. Chen and A. M. S. Zalaza. Dynamic modeling and genetic based trajectory generator for non-holonomic mobile manipulators. *Control Engineering Practice*, 5(1):39–48, 1997.
- [14] F. T. Cheng, J. S. Chen, and F. C. Kung. Study and resolution of singularities for a 7 dof redundant manipulators. *IEEE Transactions on Industrial Electronics*, 45(3):469–480, 1998.
- [15] F. T. Cheng, T. L. Hour, Y. Y. Sun, and T. H. Chen. Study and resolution of singularities for a 6 dof puma manipulators. *IEEE Transactions on Systems, Man and Cybernetics B: Cybernetics*, 27(2):332–343, 1997.
- [16] E. Cheung and V. Lumelsky. Motion planning for a whole-sensitivity robot arm manipulator. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 344–348, Cincinnati, Ohio, May 1990.
- [17] S. Chiaverini, B. Siciliano, and O. Egeland. Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator. *IEEE Trans. Control System Technology*, 2(2):123–134, 1994.

- [18] J. H. Chung and S. A. Velinsky. Interaction control of a redundant mobile manipulator. *The International Journal of Robotics Research*, 17(12):1302–1309, 1998.
- [19] J. H. Chung and S. A. Velinsky. Robust interaction control of a mobile manipulator - dynamic model based coordination. *Journal of Intelligent and Robotic Systems*, 26, 1999.
- [20] R. Colbaugh, M. Trabatti, and K. Glass. Redundant nonholonomic mechanical systems: Characterization and control. *Robotica*, 17:203–217, 1999.
- [21] O. G. DeLuca, A. and C. Samson. Feedback control of a nonholonomic car-like robot.
- [22] R. M. Desantis. Path-tracking for a tracker-trailer-like robot. *The International Journal of Robotics Research*, 13(6):533–543, 1994.
- [23] I. Duleba. Channel algorithm of transversal passing through singularities for non-redundant robot manipulators. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1302–1307, San Francisco, CA, April 2000.
- [24] M. Egerstedt and X. Hu. Coordinated trajectory following for mobile manipulation. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 3479–3484, San Francisco, CA, April 2000.
- [25] R. E. Fenton, G. C. Melocik, and K. W. Olson. On the steering of automated vehicles: Theory and experiments. *IEEE Transactions on Automatic Control*, 21(3):306–314, 1976.
- [26] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee. *Robotics: Control, Sensing Vision, and Intelligence*. McGraw-Hill Book Company, 1987.

- [27] S. D. G. Morel. The precise control of manipulators with joint friction: A base force/torque sensor method. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 360–365, Minneapolis, MN, April 1996.
- [28] A. Hemami, M. G. Mehrabi, and R. M. H. Cheng. A synthesis of an optimal control law for path tracking in mobile robots. 8(2):383–387, 1992.
- [29] N. Hogan. Impedance control - an approach to manipulation 1. theory, 2. implementation, 3. application. *Transactions of the ASME: Journal of Dynamic Systems, Measurement and Control*, 107(1):1–7, 1985.
- [30] A. Isidori. *Nonlinear control systems*. Springer, Berlin, New York, 1995.
- [31] M. Kam, X. Zhu, and P. Kalata. Sensor fusion for mobile robot navigation. *Proceedings of the IEEE*, 85(1):108–119, 1997.
- [32] I. Kaminer, A. Pascoal, E. Hallberg, and C. Silvestre. Trajectory tracking for autonomous vehicles: an integrated approach to guidance and control.
- [33] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi. A stable tracking control method for an autonomous mobile robot. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 384–389, Cincinnati, OH, May 1990.
- [34] K. Kang, A. Sparks, and S. Banda. Coordinated control of multisatellite systems. *AIAA J. Guidance, Control, and Dynamics*, 24(2):360–368, 2001.
- [35] H. K. Khalil. *Nonlinear Systems*. Prentice Hall, Inc, 1996.
- [36] O. Khatib. A unified approach to motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.

- [37] O. Khatib. Mobile manipulation: The robotic assistant. 26, 1999.
- [38] J. Kieffer. Manipulator inverse kinematics for untimed end-effector trajectories with ordinary singularities. *The International Journal of Robotics Research*, 11(3):225–237, 1992.
- [39] J. Kieffer. Differential analysis of bifurcations and isolated singularities for robots and mechanisms. *IEEE Transactions on Robotics and Automation*, 10(1):1–10, 1994.
- [40] D. H. Kim and J. H. Oh. Experiments of backward tracking control for trailer system. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 19–22, Detroit, MI, May 1998.
- [41] M. Kirćanski, N. Kirćanski, D. Leković, and M. Vukobratović. An experimental study of resolved acceleration control of robots at singularities: Damped least-squares approach. *Transactions of the ASME: Journal of Dynamic Systems, Measurement and Control*, 119:97–101, 1997.
- [42] J. K. Lee and H. S. Cho. Mobile manipulator motion planning for multiple tasks using global optimization approach. *Journal of Intelligent and Robotic Systems*, 18:169–190, 1997.
- [43] K. Liu and F. L. Lewis. Decentralized continuous robust controller for mobile robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1822–1827, Atlanta, GA, May 1990.
- [44] M. Mason. Compliance and force control for computer controlled manipulators. 11(6):418–432, 1981.
- [45] R. M. Murray and S. S. Sastry. Nonholonomic motion planning: Steering using sinusoids. *IEEE Transactions on Automatic Control*, 38(5):700–716, 1993.

- [46] K. T. R. Muszynski. Singular inverse kinematic problem for robotic manipulators: A normal form approach. *IEEE Transactions on Robotics and Automation*, 14(1):93–104, 1998.
- [47] Y. Nakamura and H. Hanafusa. Inverse kinematic solutions with singularity robustness for robot manipulator control. *Transactions of the ASME: Journal of Dynamic Systems, Measurement and Control*, 108:163–171, 1986.
- [48] D. N. Nenchev. Tracking manipulator trajectories with ordinary singularities: a null space-based approach. *The International Journal of Robotics Research*, 14(4):399–404, 1995.
- [49] D. N. Nenchev, Y. Tsumaki, and M. Uchiyama. Singularity-consistent parameterization of robot motion and control. *The International Journal of Robotics Research*, 19(2):159–182, 2000.
- [50] K. O’Neil and Y.-C. Chen. Instability of pseudoinverse acceleration control of redundant mechanisms. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2575–2582, San Francisco, CA, April 2000.
- [51] F. Papoulias. Stability considerations of guidance and control laws for autonomous underwater vehicles in the horizontal plane. In *Proceedings of 7th International Symposium on Unmanned Untethered Vehicle Technology*, pages 140–158, Durham, NH, 1991.
- [52] P. Peleties and R. Decarlo. Asymptotic stability of m-switched systems using lyapunov-like functions. In *Proceeding of American Control Conference*, pages 1679–1683, 1991.
- [53] C. Perrier, L. Cellier, and P. Dauchez. Experimental comparison of hybrid and external control structures for a mobile manipulator. In *Proceedings of IEEE*

- International Conference on Robotics and Automation*, pages 3577–3582, Albuquerque, New Mexico, April 1997.
- [54] F. G. Pin, J.-C. Culioli, and D. B. Reister. Using minimax approaches to plan optimal task commutation configuration for combined mobile platform-manipulator systems. *IEEE Transactions on Robotics and Automation*, 10(1):44–54, 1994.
 - [55] M. H. Raibert and J. J. Craig. Hybrid position/force control of manipulators. *Journal of Dynamic Systems and Control*, 102, 1981.
 - [56] M. Sampei and K. Furuata. Robot control in the neighbourhood of singular points. *IEEE Journal on Robotics and Automation*, 4(3):303–309, 1988.
 - [57] M. Sampei, T. Tamura, T. Itoh, and M. Nakamichi. Path tracking control of trailer-like mobile robot. In *IEEE/RSJ International Conference On Intelligent Robots And Systems*, pages 193–198, Osaka, Japan, 1991.
 - [58] M. Sampei, T. Tamura, T. Kobayashi, and N. Shibui. Arbitrary path tracking control of articulated vehicles using nonlinear control theory. *IEEE Trans. Control System Technology*, 3(1):125–131, 1995.
 - [59] C. Samson. Path following and time-varying feedback stabilization of a wheeled mobile robot. In *Proceedings International Conference ICARCV'92*, Singapore, 1992.
 - [60] C. Samson. Control of chained systems application to path following and time-varying point-stabilization of mobile robots. *IEEE Transactions on Automatic Control*, 40(1):64–77, 1995.
 - [61] J. D. Schutter. Improved force control laws for advanced tacking application. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1497 –1502, 1988.

- [62] H. Seraji. A unified approach to motion control of mobile manipulators. *The International Journal of Robotics Research*, 17(2):107–118, 1998.
- [63] D. H. Shin, S. Singh, and J. J. Lee. Explicit path tracking by autonomous vehicles.
- [64] Y. Sun, J. Tan, and N. Xi. Interactive model identification for nonholonomic cart pushing by a mobile manipulator. In *submitted to IEEE ICRA2002*, Washington, D.C.
- [65] J. Tan and N. Xi. Hybrid system design for singularityless task level robot controllers. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 3007–3012, San Francisco, CA, April 2000.
- [66] J. Tan and N. Xi. Unified model approach for planning and control of mobile manipulators. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 3145–3152, Seoul, Korea, May 2001.
- [67] T. J. Tarn, A. K. Bejczy, G. T. Marth, and A. K. Ramadorai. *Kinematic Characterization of the PUMA 560 Manipulators*. Washington University, 1991.
- [68] T. J. Tarn, A. K. Bejczy, G. T. Marth, and A. K. Ramadorai. Performance comparison of four manipulator servo schemes. *IEEE Control Systems Magazine*, 13(1):22–29, 1993.
- [69] T. J. Tarn, Y. Wu, N. Xi, and A. Isidori. Force regulation and contact transition control. *IEEE Control Systems Magazine*, 16(1):32–40, 1996.
- [70] D. Tibury, R. M. Murray, and S. S. Sastry. Trajectory generation for the n-trailer problem using goursat normal form. *IEEE Transactions on Automatic Control*, 40(5):802–819, 1995.

- [71] J. Tierno. Control of low earth orbit satellite clusters. In *Proceedings of IEEE Conference on Decision and Control*, Sydney, Australia, 2000.
- [72] Y. Tsumaki, S. Kotera, D. N. Nenchev, and M. Uchiyama. Singularity-consistent inverse kinematics of a 6 d.o.f. manipulator with a non-spherical wrist. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2980– 2985, Albuquerque, New Mexico, April 1997.
- [73] Y. Umeda and D. Nakamura. Hybrid position/force control of a mobile manipulator based on cooperative task sharing,. In *Proceedings of the IEEE International Symposium on Industrial Electronics*, pages 139–144, Bled, Slovenia, 1999.
- [74] M. Vukobratović. How to control robots interacting with dynamic environment. *Journal of Intelligent and Robotic Systems*, 19(2):119–152, 1997.
- [75] K. Wedeward, R. Colbaugh, and A. Engelmann. Singularity robustness: Methods for joint-space and task-space control. In *Proceedings of IEEE International Conference on Control Applications*, pages 22–27, Hartford,CT, October 1997.
- [76] Y. Wu. *Force Regulation and Contact Transition Control*. Dissertation of Washington University, 1996.
- [77] N. Xi. *Event-Based Planning and Control for Robotic Systems*. Dissertation of Washington University, 1993.
- [78] N. Xi and T. Tarn. Integrated task scheduling and action planning/control for robotic systems based on a max-plus algebra model. In *IEEE/RSJ International Conference On Intelligent Robots And Systems*, Grenoble, France, September 1997.
- [79] N. Xi and T. J. Tarn. Integration of heterogeneity for human-friendly robotic operations. *Journal of Intelligent and Robotic Systems*, 25:281–293, 1999.

- [80] N. Xi, T. J. Tarn, and A. K. Bejczy. Event-based planning and control for multi-robot coordination. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 251–258, Atlanta, Georgia, May 1993.
- [81] N. Xi, T. J. Tarn, and A. K. Bejczy. Intelligent planning and control for robot arms. pages 18–23, Sydney, Australia, July 1993.
- [82] N. Xi, T. J. Tarn, and A. K. Bejczy. Intelligent planning and control for multi-robot coordination: An event-based approach. *IEEE Transactions on Robotics and Automation*, 12(3):439–452, 1996.
- [83] N. Xi, T. J. Tarn, C. Guo, and A. K. Bejczy. Fusion of human and machine intelligence for telerobotic systems. In *Proceedings of IEEE International Conference on Robotics and Automation*, Nagoya, Japan, May 1995.
- [84] Y. Yamamoto and X. Yun. Coordinating locomotion and manipulation of a mobile manipulator. *IEEE Transactions on Automatic Control*, 39(6):1326–1332, 1994.
- [85] M. Zhao, N. Ansari, and E. S. H. Hou. Mobile manipulator path planning by a genetic algorithm. In *IEEE/RSJ International Conference On Intelligent Robots And Systems*, pages 681–688, Raleigh, NC, 1992.

MICHIGAN STATE UNIVERSITY LIBRARIES



3 1293 02372 0497