



This is to certify that the
thesis entitled
HYBRID FORCE/POSITION CONTROL
IN THE ROBOT TOOL SPACE
presented by
Amit Yogesh Goradia
has been accepted towards fulfillment
of the requirements for
Master's degree in Electrical Engineering


Major professor

Date 8/22/01

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

HYBRID FORCE/POSITION CONTROL IN THE ROBOT TOOL SPACE

By

Amit Yogesh Goradia

A THESIS

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

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

2001

ABSTRACT

HYBRID FORCE/POSITION CONTROL IN THE ROBOT TOOL SPACE

By

Amit Yogesh Goradia

The capability of performing constrained motion tasks is key to many robotic applications that involve material handling, milling, deburring, polishing and surface tracking. Many strategies for performing constrained motion control have been proposed. The major drawback of these strategies is that they do not provide for the manipulator to interact with poorly structured or unknown environments. This is because using these strategies the interaction forces between the end-effector and the workpiece cannot be controlled in arbitrary directions.

This thesis presents the design of a new controller implemented in the robot tool space that can control the interaction forces in arbitrary directions. Using the proposed controller, the manipulator can interact with workpieces whose shapes are not pre-specified. Thus tasks like milling, deburring and polishing, that involve constrained motion control of the manipulator can be performed on a wide variety of workpieces without explicit knowledge of the shape of the workpiece. The design and proof of stability for this tool space hybrid force/position controller is presented. A comparison of this new tool space hybrid force/position controller with the existing task space hybrid force/position controller is also presented. Simulations studies for the task of tracking unknown surfaces are also presented.

copyright by

Amit Yogesh Goradia

2001

to my family

ACKNOWLEDGEMENTS

Foremost, I would like to express my deep respect and sincere gratitude to my advisor Dr. Ning Xi, without whom this work would not have been possible. This work is a result of his keen insight, knowledge, guidance, perseverance, encouragement and support. I would like to thank Jindong Tan from the Robotics and Automation Laboratory for his invaluable help. The discussions with him contributed substantially to my work and broadened my perspective. I would also like to thank my committee members Dr. H Khalil and Dr. R. L. Tummala for their time and valuable comments.

I would like to thank Aluel Go from the department of Agriculture Engineering, MSU for his continued support during the period of this research. I am grateful to my friends and colleagues from the Robotics and Automation Laboratory.

I dedicate this work to my family for their continuous emotional support: To my parents, for believing in me, to my sister-in-law, for everything she's done and finally to my brother, who set the standards for me, shared all my success and failures and stood by me through everything. Without their love and support this work would not have been possible.

Amit Yogesh Goradia

Michigan State University, East Lansing

August 2001

TABLE OF CONTENTS

List of Figures

Chapter 1

1.1 Introduction	1
1.2 Organization of this Thesis	4

Chapter 2

2.1 Understanding the Problem	7
2.2 Review of Related Works	9

Chapter 3

3.1 The Manipulator Dynamics Model	17
3.2 Computed Torque Control	18
3.3 Task Space Control	22
3.4 Tool Space Control	26
3.5 Relationship between the Task Space and the Tool Space	33
3.5.1 Position error in the tool frame	34
3.5.2 Velocity error in the tool frame	36

Chapter 4

4.1 Hybrid Force/Position Control	39
4.2 Hybrid Force/Position Control in the Task Space	42
4.3 Hybrid Force/Position Control in the Tool Space	46
4.4 Stability Analysis of the Tool Space Controller	52

Chapter 5	
5.1 Simulation Studies - Setup	59
5.2 Position Control Experiments	60
5.3 Hybrid Force/Position Control Experiments	67
5.4 Tracking a Curve in Two Dimensions	79
5.5 Tracking a Curve in Three Dimensions	89
Chapter 6	
6.1 Summary	95
6.2 Conclusion and Future Work	96
Appendix A	99
Appendix B	110
References	115

LIST OF FIGURES

Figure 1.1: A robot in contact with a surface of arbitrary shape	3
Figure 2.1: The robot in contact with the environment	8
Figure 3.1: The robot arm, controller and planner	18
Figure 3.2: Joint space computed torque controller	20
Figure 3.3: The task space computed torque controller for position control	24
Figure 3.4: The robot task frame and the tool frame	27
Figure 3.5: The tool space computed torque controller	31
Figure 3.6: The relationship between the task space and tool space frame of reference	34
Figure 4.1 Hybrid force/position controller	41
Figure 4.2 Task space hybrid force/position controller	45
Figure 4.3: The tool space hybrid force/position controller	51
Figure 5.1: The actual and desired position and orientation for task level position control	63
Figure 5.2: The actual and desired velocities for task level position control	63
Figure 5.3: The error in position for task level position control	64
Figure 5.4: The error in velocity for task level position control	64

Figure 5.5: The actual and desired positions for the tool space controller	
expressed in the task space reference frame	65
Figure 5.6: The actual and desired velocities for the tool space controller	
expressed in the task space reference frame	65
Figure 5.7: The errors in position for the tool space position controller	
expressed in the task coordinate frame	66
Figure 5.8: The errors in velocity for the tool space position controller	
expressed in the task coordinate frame.	66
Figure 5.9: Actual and desired positions for task space hybrid force/position	
control	69
Figure 5.10: Actual and desired velocities for task space hybrid force/position	
control.	69
Figure 5.11: Error in position for task space hybrid force/position control.	70
Figure 5.12: Error in velocity for task space hybrid force/position control.	70
Figure 5.13: The actual and desired force profiles for the task space hybrid	
force/position controller	71

Figure 5.14: The error in the force profiles for the task space hybrid	
force/position controller	71
Figure 5.15: The actual and desired positions, expressed in the task space,	
for the tool space hybrid force/position controller	74
Figure 5.16: The actual and desired velocities, expressed in the task space,	
for the tool space hybrid force/position controller	74
Figure 5.17: The position errors expressed in the task space for the tool space	
hybrid force/position controller	75
Figure 5.18: The errors in velocity, expressed in the task space, for the tool	
space hybrid force/position controller	75
Figure 5.19: Position errors expressed in the Cartesian tool space for the	
tool space hybrid force/position controller	76
Figure 5.20: Velocity errors expressed in the Cartesian tool space for the	
tool space hybrid force/position controller	76
Figure 5.21: The actual and desired forces expressed in the task space for the	
tool space hybrid force/position controller	77

Figure 5.22: The actual and desired forces expressed in the tool space frame of reference for the tool space hybrid force/position controller	77
Figure 5.23: The force error expressed in the tool space frame of reference for the tool space hybrid force/position controller	78
Figure 5.24: The cylindrical parabolic surface to be tracked	82
Figure 5.25: The actual positions and orientations of the manipulator	83
Figure 5.26: The interaction force expressed in the task space frame of reference	83
Figure 5.27: The interaction force expressed in the tool space frame of reference	84
Figure 5.28: The surface profile and the actual profile of the end-effector location in the x - z plane in the task space	84
Figure 5.29: The task space x , y and z components of the surface normal and the z -axis of the end-effector frame	85
Figure 5.30: The actual position of the end-effector expressed in the task space frame of reference	87
Figure 31: The interaction force expressed in the task space frame of reference	87
Figure 5.32: The interaction force expressed in the tool space frame of reference	88
Figure 5.33: The spherical surface that is to be tracked	91

Figure 5.34: The actual and desired positions and orientations expressed in the task space frame of reference	92
Figure 5.35: The actual interaction force expressed in the task space frame of reference	92
Figure 5.36: The actual and desired interaction forces expressed in the tool space frame of reference	93
Figure 5.37: The metric $m = x^2 + y^2 + z^2$.	93
Figure 5.38: The task space x , y and z components of the surface normal and the z -axis of the end-effector frame	94

Chapter 1

1.1 Introduction

Robotics is an exciting area that encourages many engineers to study the synthesis of some aspects of human function by the use of mechanisms, sensors, actuators and computers. There are many fictional representations of futuristic robots that we see in movies, on T.V. and in books. These robots are attributed excessive power, exceptional intelligence, and they even look like human beings. They are expected to work for human society in order to relieve humans from low level and heavy-duty jobs. After many years of investigation, people began to realize that human beings are so perfectly created that it is almost impossible to exactly simulate them.

The evolution of robotics is closely tied to the development of the technology of controls, sensors and computers. Robot control strategies have evolved from joint level control to task level control and from a non-model based approach to a model based approach. Recently there have been many advances in the field of robot force control and in the development of computation and sensing technology. Given these technological advances, robot force control may eventually replace classical joint level control in order to increase productivity by increasing operational speed and system flexibility.

The use of robots for performing industrial tasks has emerged as a major means for implementing industrial automation to achieve the ultimate goal of enhancing

productivity and product quality. The industrial robots of the early 1980s were mostly used in relatively simple operations like machine tending, material handling, painting and welding. These robots were basically position-controlled mechanisms utilizing error driven, simple independent joint control laws and did not use a model-based component at all. However, to utilize the robots for more labor-intensive applications *e.g.*, milling, deburring, polishing and surface tracking, more sophisticated control capabilities with a variety of sensors are required.

Recent applications of robots require the robot to come in contact with its environment. The environment can be either static (*i.e.*, fixed) or dynamic (*i.e.*, moving) or a combination of both. The capability of the robot to perform industrial tasks, such as milling, deburring and polishing, that involve contact with the workpiece depends on the ability of the controller to control the interaction forces between the robot and the environment. The ability to control the interaction forces could be achieved by the means of extremely precise position control in a pre-specified environment with very low tolerance. Such a scenario could only be true for either laboratory experiments or hard automation tasks where the robot is dedicated to a specific task and jigs with extremely high precision have been designed for the task. Also for such a control strategy the component part tolerances must be maintained at sufficiently low levels, which most likely adds to the cost of the final product.

For soft automation tasks performed in uncertain environments, position control alone cannot guarantee the ability to control the interaction forces. Also, the description of assembly tasks in terms of position-controlled motion is unnatural. Force control is a

natural control strategy for the assembly of a large class of products with low part tolerances.

The ability to control the force exerted by the robot in any arbitrary direction is very important when dealing with tasks that require contact of the robot with poorly structured or unknown workpieces (environment). Force Controllers implemented in literature do not provide the freedom to control the force applied by the robot in any arbitrary direction. Also for these controllers an accurate knowledge of the environment is needed for specifying the constraint frame. Thus, tasks like normal surface tracking, milling and deburring, which must be carried out on objects that are not pre-specified cannot be carried out. For example, Figure 1.1 shows a robot in contact with a surface whose exact shape is not known a priori.

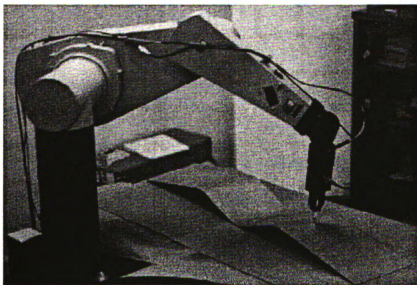


Figure 1.1: A robot in contact with a surface of arbitrary shape.
(Courtesy Dr. N. Xi, <http://www.egr.msu.edu/~xin>)

This thesis presents a new design of a controller that is implemented in a non-inertial frame (*i.e.*, moving frame). This controller provides for controlling the force

exerted by the robot in any direction while simultaneously controlling the position of the robot in the remaining directions. The design also accounts for the lack of knowledge about the exact shape of the object. Thus, the robot can interact with unknown environments and with workpieces whose shape is not exactly known.

A potential application of this controller would be in determining the exact shape of an unknown surface. It can also be used in milling, deburring, painting and welding of surfaces whose exact shape is not known a priori. This would save a lot of time in finding out the exact nature of the surface and in planning the robot path along the surface. Also the work piece can be changed instantly without much down time in order to plan for a new workpiece. This controller can also rule out the need for expensive jigs, which are specific to every object, needed for hard automation. Another advantage would be that larger part tolerances can be accepted which will reduce the cost of the final product.

1.2 Organization of this Thesis

The scope of this thesis is to investigate the implementation of a force control strategy by means of which the force applied by the robot can be controlled in any arbitrary direction. This work discusses the development of robot controller, implemented in the robot tool space, which can satisfy the above requirement. The discussion is divided into six chapters.

In Chapter 2 the mechanism of the problem is explained in detail and a review of related works is presented. Chapter 3 and Chapter 4 present the major contributions of

this research. In Chapter 3, the concept of pure position control in the robot tool space is discussed. First the robot dynamics model is given. Then a joint space and task space computed torque controllers are developed. Further a tool space controller is introduced and the relationships between the position, and velocity, in the task, and tool spaces, are explained in detail.

Chapter 4 introduces the concept of hybrid force position control in the task space. This controller is not robust and has limited applicability. Thus in order to overcome these drawbacks, the concept of hybrid force/position control in the robot tool space is developed. Stability analysis of a position controller in the tool space and the proposed tool space hybrid force/position controller are presented.

Simulation studies are presented in Chapter 5. First, position control experiments for the task space controller and the tool space controller are presented. For the task of moving along a horizontal constrained surface while applying a constant normal force to the surface, the task space hybrid force/position controller is compared with the tool space hybrid force/position controller. Further the tool space hybrid force/position controller is used to track an unknown two-dimensional trajectory while simultaneously regulating the force normal to the trajectory. Lastly, the tool space hybrid force/position controller is used to track an unknown three-dimensional trajectory. Chapter 6 first presents a summary of the work presented and then presents the conclusions and future work.

Appendix A presents the definition of Euler angles and the avoidance of singularities associated with the definition of the orientation representation. Appendix B

presents the derivation of the tool Jacobian and its derivative, which are used to decouple the tool space axes.

Chapter 2

2.1 Understanding the Problem

Before the manipulator comes in contact with the environment, the environment and the manipulator are two separate systems with no coupling between them. As soon as the manipulator comes in contact with the environment, the end-effector and the environment are pushed together and their surfaces deform. As a result, reaction forces are generated. The reaction forces, depending upon the control force and the dynamics of the system, now couple the originally separated systems. These reaction forces arise from the stiffness or admittance of the environment and the manipulator, which as in most cases is uncertain and nonlinear. Thus the coupled system, comprising of the manipulator and the environment, has uncertain and nonlinear dynamics. Figure 2.1 shows the robot in contact with the environment. It also identifies the task and the tool frame of reference.

Previous research studies on force control mostly deal with tasks where the shape of the environment is exactly known. Uncertainty in the shape of the environment is not taken into account. In these cases particular strategies for task formulation can be developed prior to the execution of the task. Thus, the direction in which force is to be controlled can be aligned to the task coordinate system. However, this sort of strategy formulation is precluded in the case where the shape of the environment is not exactly known.

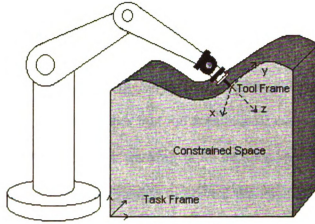


Figure 2.1: The robot in contact with the environment.

Many industrial tasks require the manipulator to handle various sizes and shapes of workpieces (objects), which, in many cases, are not known during the designing of the controller. For these reasons a generalized control algorithm must be developed which can deal with workpieces whose shape is not known a priori. One solution to the aforementioned problem is to develop a controller that can control the interaction force in any arbitrary direction.

An approach to constrained robot motion control is the hybrid force/position control strategy proposed by Raibert and Craig [4]. This strategy identifies the degrees of freedom in which the force and position are controlled individually. The directions in which the force and position are to be controlled must be orthogonal to each other. This method is conceptually much clearer than many other methods as it allows for design of the control law in the same reference frame used to describe the manipulator. Thus, its implementation is more practically feasible. However, the force can be controlled only

along the directions of the reference frame used to describe the manipulator. Therefore this strategy does not allow for the implementation of a general control algorithm by means of which the manipulator can interact with unknown objects.

The objective of this research work is to expand on the concept of hybrid force/position control in such a manner that the manipulator can interact with objects of unknown shape and perform contour tracking tasks on such objects. In order to attain this objective the manipulator must be able to control the applied force in any arbitrary direction that is not necessarily aligned with the reference frame used to describe the manipulator.

2.2 Review of Related Works

Traditional academic study of robot arm control deals with motion in space with no contact with the environment. As the application area of robot manipulators expanded to include such sophisticated tasks as assembly, polishing, deburring etc., the necessity for controlling the interaction forces as well as the position simultaneously was recognized more strongly. Historically, robot force control first began with remote manipulator and artificial arm control in the 1950s and 1960s. These systems were controlled in "natural" ways and lacked task understanding and strategy. With the advent of computer controls, various approaches to the creation of strategies for application to force feedback emerged. However all the approaches depended on people to formulate the details and the systems that were tested experimentally encountered the stability problem. The problem of stability was resolved by using sophisticated control algorithms, faster computation and flexible sensors.

In the 1950s, Goertz [2] developed mechanical master-slave manipulators with force reflectors for radioactive hot lab work. The operator guided the master with his hand and felt the contact forces experienced by the slave, which were reflected through the joints of both the devices. In the 1960s, Rothchild and Mann [3] led the development of a force feedback artificial arm for amputees.

In the 1960s and early 1970s, the human operator was replaced by a computer. However in implementing this transition two major problems emerged: (1) how to structure multi-axis arm systems that sense forces and use them to modify position commands, and (2) how to relate the requirements of a task to the motions required and the forces anticipated so that force motion strategies could be formulated. Since then the main approaches to force control have been [1]:

Logical branching feedback method (Earnst 1961): The essence of this method is a set of discrete moves terminated by a discrete event such as contact. IF-THEN logic is used in the control strategy.

Continuous feedback method (Groome 1972): It was applied to assembly and edge following tasks to maintain continuous contact. It is an early effort in continuous force feedback.

Damping method (Whitney 1977, Paul and Shimano 1976): It is an integrating controller in which sensed forces give rise to velocity modification.

Position methods:

Active compliance (Salisbury 1980): A desired force is calculated based on the difference between the desired and actual hand position.

Passive compliance (Watson 1976): The end-effector is equipped with a passive mechanical device composed of springs and dampers. It is capable of quick responses and is relatively inexpensive. But the application of such devices is limited to very specific tasks.

Impedance control (Hogan 1980) [5] [6]: An attempt to control the dynamic interaction between the manipulator and the environment. The dynamic relationship between force and position is controlled instead of pure force or position. The control structure is simple and the performance is robust. The drawback is limitation of the dynamic performance. Output force cannot be regulated unless an accurate environmental model can be obtained. A. Haze *et. al.* [7] developed a robust impedance controller design based on the chattering free continuous sliding mode.

Explicit force control (Nevins and Whitney 1973): This method employs a desired force input instead of a position or velocity input.

Implicit force control (Borrel 1979): There is no force sensor used for sensing the force. Instead the joint servo gains are adjusted to impart a particular stiffness matrix to the robot arm.

Hybrid force-position control (Raibert and Craig 1981 [4]; Mason 1981 [8]): It is one of the most popular force control schemes nowadays. It partitions the force control

problem using a set of position and force constraints depending on the mechanical and geometric characteristics of the performed task. Position constraints exist along the normal to the surface where the presence of a surface constrains the range of motion. Force constraints exist along surface tangents where it is impossible to apply arbitrary levels of force. A compliance selection matrix is used to determine the force-controlled direction and position-controlled direction. Various force control methods can be used in the force-controlled direction. The one most frequently used is explicit force control. Hybrid force/position control has given rise to many research works, which build on the same original concept of partitioning of the space into two orthogonal, force controlled and position controlled subspaces [9] [10] [11]. In [12] R. Anderson and M. Spong proposed a hybrid impedance controller, which combines the hybrid control approach with the impedance control approach. [13] presents a general first-order kinematic model of frictionless rigid-body contact for use in hybrid force/position control. These more general kinematics can be used to model tasks that cannot be described using the Raibert-Craig model such as maintaining contact at two points with skew, non-intersecting contact normals.

In recent years force control has been studied intensively. Various control methods have been proposed for constraint motion control. These include dynamic hybrid position/force control [14], operational space approach [15], compliant motion control [16] [17], object impedance control [18], hybrid control considering motor dynamics [19] and adaptive force control [20] [21] [22].

In the dynamic hybrid position/force control approach proposed by Tsuneo Yoshikawa [14] the dynamics of the manipulator are taken into consideration. The

constraints on the end-effector of the manipulator are described by a set of constraint hypersurfaces. It takes into account the end-effector coming into contact and moving along an arbitrary but known surface, though it does not deal with movement along an unknown surface.

In the operational space approach proposed by Khatib [15], generalized task specification matrices Ω and $\tilde{\Omega}$ are defined, which act on vectors described in the base frame of reference. The matrix Ω specifies, with respect to the base frame of reference, the directions of motion (displacement and rotations) of the end-effector. Forces and moments are to be applied in or about directions that are orthogonal to these motion directions. The matrix $\tilde{\Omega}$ specifies these directions. The end-effector equations of motion are developed in the operational space and control algorithm for free and constrained motion modes is given. This formulation proposed a unified approach for control of constrained manipulators. But it did not address the issue of the manipulator coming into contact with an unknown environment. In order to adopt this approach the exact nature of the task at hand must be known in advance and the task specification matrices must be defined a priori to the execution phase. In [29] Bona and Indri extended the operation space approach in order to achieve exact decoupling of the force and position control along the task subspaces.

[50] [56] and [51] point out the fallacy of the theoretical formulation of hybrid force/position control that is based on the concept of the position and force-controlled subspaces being orthogonal. Duffy [50] showed that the orthogonality was not invariant under the choice of coordinate frames and that the conventional formulation

therefore lacks physical meaning. [56] shows that the end-effector velocities and forces are not vectors but screws represented in axis and ray coordinates. Then an approach to the decomposition of the manipulator dynamics based on screw reciprocity is presented. In [51] Vukobratovic presents the concept of a new hybrid controller based on the environment dynamic nature and non-holonomic constraints.

Few research works have been focused on performing simultaneous position and force control on unknown environments, identifying at the same time their geometry [30]. In [31] a theoretical framework for the problem of exploring the environment and updating a nominal model is formulated. The problem of improving, by repeated experiments, the performance of a force control scheme over an unknown object is investigated in [32]. In [33] a hybrid velocity/force control scheme is described that employs local knowledge of the environment shape, obtained by sensing. [55] discusses the use of the physical concept of reciprocity and consistency to identify the motion constraints. A simple yet interesting approach to surface tracking tasks on unknown objects using C-surface concept is presented in [49]. It employs a task space hybrid force/position controller. In order to control the force along arbitrary directions, the task space axes definitions are changed rapidly in order to align them with the C-surface axes. The drawback of this method is that it cannot be used for surfaces with large curvature. Also stability of the manipulator system during the switching is not guaranteed.

Many researchers have addressed the inherent stability and other problems associated with force control. Kinematic and dynamic stability issues are discussed in [34] and [35]. [34] discussed the effect of the kinematic coordinate transformations on stability. It also showed that, hybrid force/position control method of Raibert and Craig

induced instabilities for revolute manipulators whereas other force control methods such as stiffness control, resolved acceleration control and operational space methods, do not induce such instabilities. In [36] it was shown that the $J^{-1}SJ$ term was the cause of the kinematic singularity pointed out in [34] and a sufficient stability condition was derived for the hybrid force/position controller. In [35] the dynamic stability issues related to force control are discussed. It is shown that the robot system tends to become unstable when in contact with stiff environments, due mainly to the high gain nature of the wrist force sensor feedback. In [16] a nonlinear approach for the stability analysis of compliant motion control was discussed and a bound for stable manipulation was derived. [37] provides an in depth analysis of position and force control of robot manipulators and a stability analysis for robot manipulators under the influence of external forces is presented.

The effects of friction and stiction on force control of manipulators are discussed in [38]. In [39] asymptotic stability issues associated with hybrid force/position control are addressed. [40] discusses the stability of impedance control and [41] addresses the problem of Cartesian compliance. Dynamic problems are discussed in [42]. [43] [54] and [44] discuss the important considerations in implementation and the bandwidth problems. Research work based on the aforementioned formulations were presented in [29], [45], [46], [47] and [48].

Many researchers have studied the control of phase transition (from free motion mode to constrained motion mode) and various schemes have been proposed. Approaches to impact control include generalized dynamical systems [23], discontinuous

control [25], [26], [11], [27], adaptive force control [24] and jump impact control [28] and positive acceleration feedback control [52] and [53].

The existing schemes of force control can be basically divided into two main categories. The first category is impedance control. It provides a stable uniform control structure for free as well as constrained motion of the manipulator. The problem with impedance control is that in order to regulate the output force an exact model of the environment must be known and integrated into the motion plan. The second category is hybrid control, which is a discontinuous scheme. In this scheme the output force can be controlled without an accurate knowledge of the environment but kinematic and dynamic stability issues must be addressed.

Chapter 3

3.1 The Manipulator Dynamics Model

The mathematical formulation of the equations of motion of an n degree of freedom robot arm, *i.e.*, the robot arm (manipulator) dynamics, can be expressed as a set of ordinary differential equations ([57], [58] and [59]). These ordinary differential equations are derived using the Lagrange-Euler formulation for the equations of motion of a rigid body. The manipulator dynamics, in general can be written as:

$$D(q)\ddot{q} + C(q, \dot{q}) + G(q) + J^T f = \tau, \quad (3.1.1)$$

$$y = H(q), \quad (3.1.2)$$

where τ is the $n \times 1$ vector of the torques applied to the individual joints, $q \in \mathcal{R}^n$ is the $n \times 1$ vector of the joint angles, $\dot{q}, \ddot{q} \in \mathcal{R}^n$, are the $n \times 1$ joint space velocity and acceleration vectors respectively. $D(q)$ is the $n \times n$ inertia matrix, $C(q, \dot{q})$ is the $n \times 1$ vector of the centripetal and coreiolis forces and $G(q)$ is the $n \times 1$ vector of gravity terms. y is the 6×1 vector of task space positions. $H(q)$ is the forward kinematics *i.e.*, it expresses the task space positions, y , in terms of the joint angles, q . $J(q)$ is the $n \times 6$ Jacobian relating the task space velocities to the joint space velocities, *i.e.*, $\dot{y} = J\dot{q}$. f is the 6×1 output force and moment vector in the task space.

3.2 Computed Torque Control

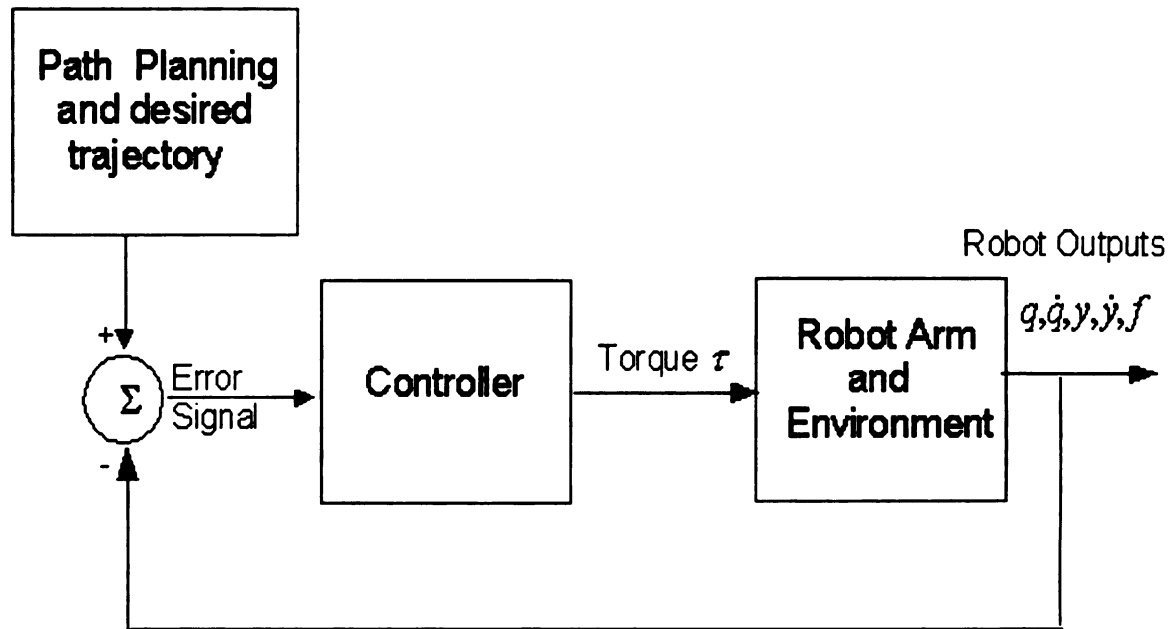


Figure 3.1: The robot arm, controller and planner.

Figure 3.1 shows the basic diagram of the robot and the controller. The input to the robot is the torque, τ , and the outputs are the joint position and velocity, Cartesian position and velocity and the output force. The controller maps the error signal input to the output torque. The planner generates the desired trajectory that is used to calculate the error signal, which in turn drives the controller.

The computed torque technique is basically a feedforward control scheme with feedback components. The feedforward control components compensate for the interaction forces between all the joints of the manipulator and the feedback component computes the correction torques to compensate for any deviations from the desired trajectory. Given the Lagrange-Euler formulation for the equations of motion of a

manipulator, the control problem can be formulated as: *“Find the appropriate motor torques to servo all the joints of the manipulator in real time in order to track a time-based trajectory as closely as possible”*. The individual motor torques are computed with the dynamic model of the manipulator using an estimate of the parameters in the model. Errors in position and velocity (time derivative of position) are feedback to the controller in order to compensate for modeling errors and parameter variations in the model. The structure of the control law has the form (as given in [53], [60]):

$$\tau = \hat{D}(q)v + \hat{C}(q, \dot{q}) + \hat{G}(q) + J^T f, \quad (3.2.1)$$

$$v = \ddot{q}_{des} + K_p(q_{des} - q) + K_v(\dot{q}_{des} - \dot{q}). \quad (3.2.2)$$

Here v is the vector of auxiliary inputs and q_{des} , $\dot{q}_{des}$, $\ddot{q}_{des}$ collectively represent the robot task expressed in the joint space *i.e.*, the desired trajectory in the joint space. K_p and K_v represent the feedback position and velocity gain matrices respectively.

Figure 3.2 shows the diagram of the joint space computed torque controller. The torque τ , which is output by the controller, is the input to the manipulator. The planner generates the desired trajectory, which is then used to calculate the error signal that drives the controller.

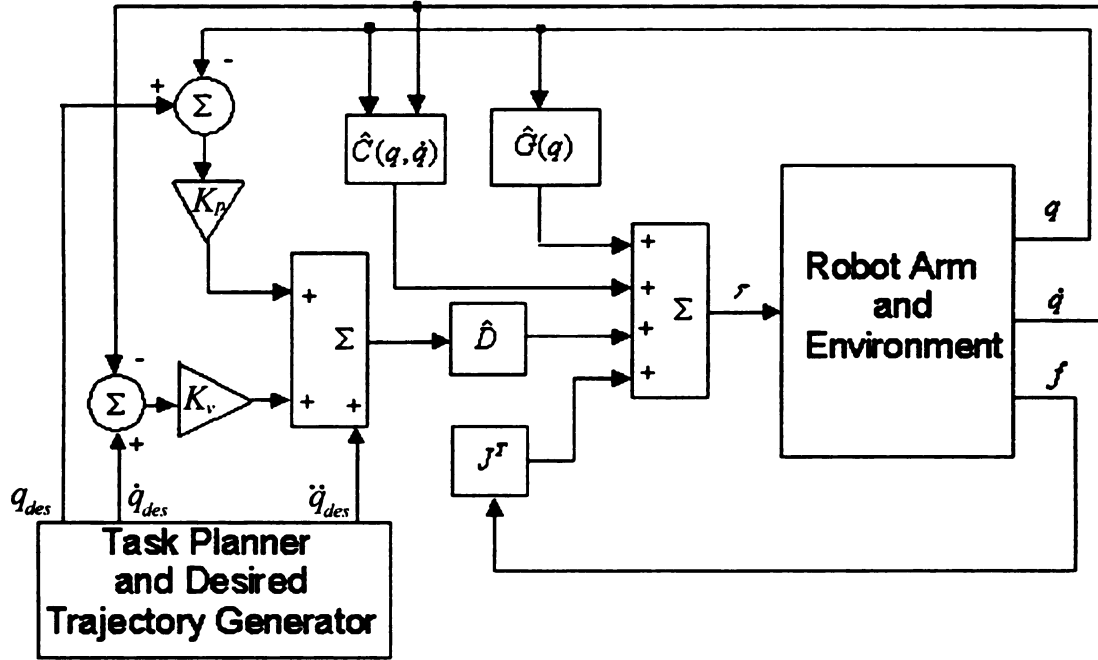


Figure 3.2: Joint space computed torque controller.

Substituting the computed torque, τ , in the dynamics equation of the manipulator (Equation 3.1.1)

$$D(q)\ddot{q} + C(q, \dot{q}) + G(q) = \hat{D}(q)(\ddot{q}_{des} + K_p(q_{des} - q) + K_v(\dot{q}_{des} - \dot{q})) + \hat{C}(q, \dot{q}) + \hat{G}(q).$$

Assuming that the parameters of the manipulator can be accurately estimated i.e., $D(q)$, $C(q, \dot{q})$ and $G(q)$, are equal to $\hat{D}(q)$, $\hat{C}(q, \dot{q})$ and $\hat{G}(q)$ respectively, the above equation simplifies to:

$$D(q)((\ddot{q}_{des} - \ddot{q}) + K_p(q_{des} - q) + K_v(\dot{q}_{des} - \dot{q})) = 0 \quad (3.2.3)$$

Now $D(q)$ is the inertia matrix and is by definition, nonsingular. The error dynamics of the combined system comprised of the manipulator and the controller can be formed as:

$$\ddot{e} + K_v \dot{e} + K_p e = 0, \quad (3.2.4)$$

where the joint space error e is expressed as $e = q_{des} - q$.

Expressing the above equation in state space representation and taking the state vector to be $x = [e \quad \dot{e}]^T$, Equation 3.2.4 can be written as:

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -K_p & -K_v \end{bmatrix} x, \quad (3.2.5)$$

The above Equation 3.2.5 can be written as:

$$\dot{x} = Ax$$

$$\text{where } A = \begin{bmatrix} 0 & 1 \\ -K_p & -K_v \end{bmatrix}.$$

By selecting appropriate values of position gain, K_p and the velocity gain K_v it can be ensured that the eigenvalues of the matrix A are located in the left half plane. This would in turn ensure that A is stable. Therefore the above error system is locally asymptotically stable for an appropriate choice of the gain matrices K_p and K_v *i.e.*, the error e asymptotically tends to zero. This means that, irrespective of the manipulator configuration, the controller system (Equation 3.2.1 and 3.2.2) can track any reasonable trajectories specified as q_{des} , $\dot{q}_{des}$ and $\ddot{q}_{des}$.

3.3 Task Space Control

In the computed torque joint space controller described in the previous section, the task to be performed (manipulator trajectories) have to be specified in the joint space (the generalized coordinate space). However, a robot task is generally represented by desired end-effector positions and orientations, and their time derivatives in the Cartesian coordinate frame. Thus, in order to use the joint space controller developed in the preceding section, inverse kinematics of the desired world space (Cartesian space) trajectories would have to be computed in order to get the desired trajectories in the joint space. This would be quite tedious. Instead, the manipulator dynamics and the desired trajectories could be expressed directly in the task space and thereby compute the error feedback terms for the control law in the world space (task space) coordinates (as is done in [61]). Let $y \in \mathbb{R}^6$ be a task space vector defined by $y = (x, y, z, O, A, T)^T$, where $(x, y, z)^T$ denotes the position of the end-effector in the task space and $(O, A, T)^T$ denotes an orientation representation (Orientation, Attitude, Tool angles) [62]. The relationship between the joint space position variable q , and the task space position variable y can be represented by:

$$y = H(q).$$

The velocity in the joint space can be related to the velocity in task space with the help of a Jacobian J_0 as:

$$\dot{y} = J_0(q)\dot{q}, \tag{3.3.1}$$

where J_0 is called the OAT Jacobian matrix. Using the above equation the relationship between the accelerations in the two spaces can be expressed as:

$$\ddot{y} = \dot{J}_0 \dot{q} + J_0 \ddot{q}. \quad (3.3.2)$$

The dynamics model of the manipulator expressed in terms of the task space variables can now be described as:

$$D(q)J_0^{-1}(q)(\ddot{y} - \dot{J}_0 \dot{q}) + C(q, \dot{q}) + G(q) + J^T f = \tau. \quad (3.3.3)$$

The nonlinear feedback control law can now be formulated in the task space as (from [53]):

$$\tau = D(q)J^*(v - \dot{J}_0 \dot{q}) + C(q, \dot{q}) + G(q) + J^T f, \quad (3.3.4)$$

where J^* is the pseudo-inverse of the OAT Jacobian matrix J_0 . If $n = 6$, J_0 is a square matrix, and J^* can be replaced by J_0^{-1} .

Applying this to the dynamics model of the manipulator, the manipulator system can be linearized and decoupled as:

$$\ddot{y} = v, \quad (3.3.5)$$

where v is the vector of auxiliary inputs.

A position and derivative (PD) control law for the controller can now be formulated as in [53]:

$$\mathbf{v} = \ddot{\mathbf{y}}_{des} + \mathbf{K}_v(\dot{\mathbf{y}}_{des} - \dot{\mathbf{y}}) + \mathbf{K}_p(\mathbf{y}_{des} - \mathbf{y}). \quad (3.3.6)$$

Figure 3.3 shows the diagram of the task space position controller. The trajectory planner generates a desired trajectory in the task space that is used to generate the task space position and velocity error. The controller uses this error to generate the joint torques τ . Note that the inverse of the OAT Jacobian matrix, J_0^{-1} , is used by the controller to decouple the task space directions.

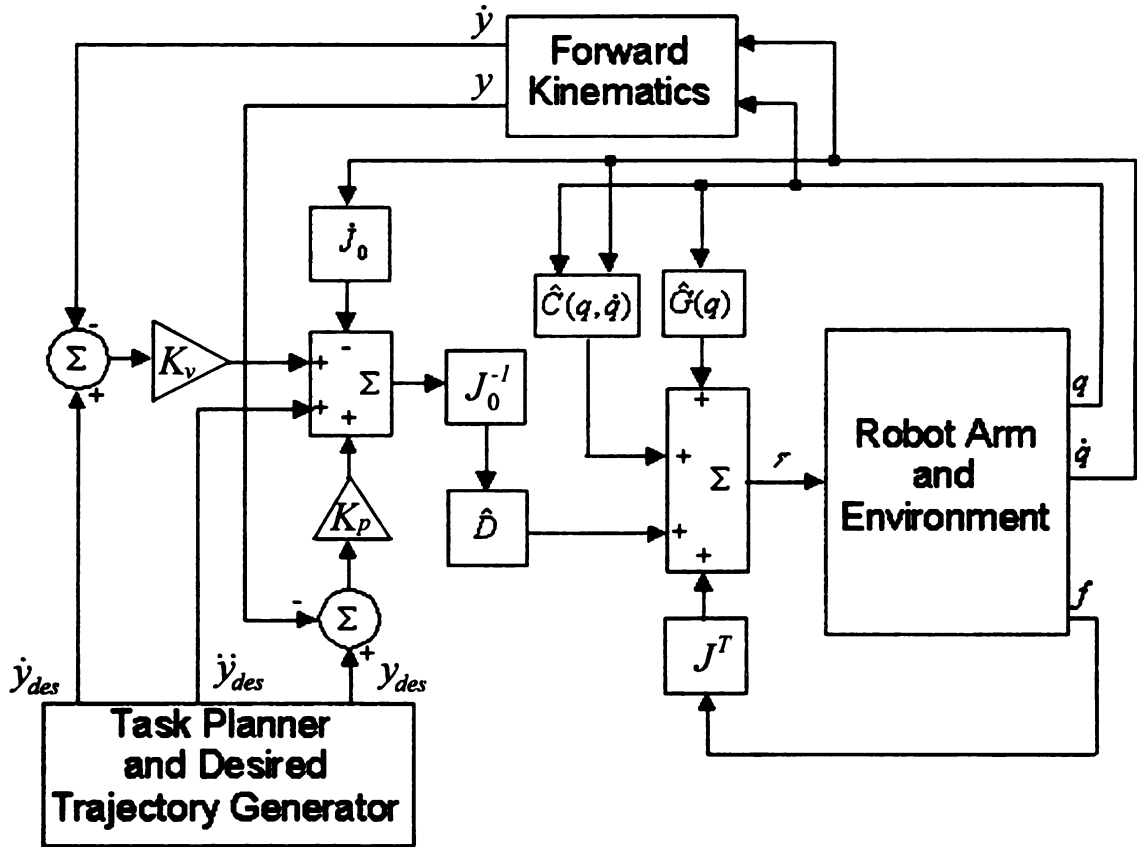


Figure 3.3: The task space computed torque controller for position control.

Using the dynamics equation of the manipulator (Equation 3.1.1), the combined linearized and decoupled system of the controller and manipulator can now be expressed as:

$$(\ddot{y}_{des} - \ddot{y}) + K_v(\dot{y}_{des} - \dot{y}) + K_p(y_{des} - y) = 0. \quad (3.3.7)$$

Now if the error in the task space is denoted as:

$$e_y = y_{des} - y, \quad (3.3.8)$$

the above Equation 3.3.8 can be expressed as:

$$\ddot{e}_y + K_v\dot{e}_y + K_p e_y = 0. \quad (3.3.9)$$

It is easy to show that the error system above is locally asymptotically stable for appropriate choices of the gain matrices K_p and K_v , as is done with the error dynamics of the joint level controller in the previous section. However, it should be noted that the existence of the task level controller depends on the existence of J_0^{-1} , while that of the joint level controller is independent of the inverse OAT Jacobian, J_0^{-1} . If the determinant of J_0 is very small, then, the determinant of J_0^{-1} could be very large, which would result in extremely high joint torques. Robot configurations for which $|J_0| = 0$ are called singular configurations and task level control cannot be successfully applied at and around such configurations. However, there are several methods using which singularity robustness can be achieved. One of them is singular robust inverse ([63], [64] [65]). In this method the inverse Jacobian is calculated using the damped least-squares technique

for calculating the matrix inverse. This method ensures that stable control can be achieved in the vicinity of the singularity although it cannot prevent the instability caused exactly at the singular configuration. Various other methods have been proposed which incorporate singularity avoidance in the path planning [66], [67].

3.4 Tool Space Control

In the previous section, using feedforward control, the manipulator system was linearized and decoupled as:

$$\ddot{y} = u ,$$

i.e., the error system applied as the feedback part of the control law was converted to the task space from the space of the generalized coordinate system by using the transform Equations 3.3.1 and 3.3.2.

It is important to note that both the above spaces are inertial spaces *i.e.*, stationary spaces. There are certain tasks like online surface estimation, surface tracking, milling and deburring that can be specified more easily and sometimes only in a space located at the end effector of the manipulator called the tool space. Consider $h \in \mathcal{R}^n$ being a tool space vector. It is defined as $h = (x, y, z, O, A, T)^T$, where $(x, y, z)^T$ denotes a position vector expressed in the tool space, $(O, A, T)^T$ denotes an orientation representation (Orientation, Attitude, Tool angles) [62] expressed in the tool space.

Figure 3.4 shows the Cartesian coordinate frame and the tool coordinate frame with the current and desired positions of the end effector. The actual and desired

positions can be expressed in both coordinate frames i.e., the task frame and the tool frame. Also coordinates of a particular point in space can be converted from one frame of reference to another using linear homogenous transformations.

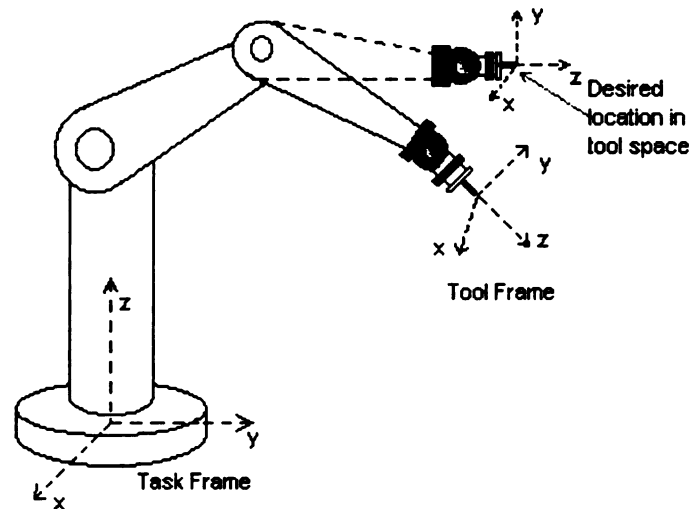


Figure 3.4: The robot task frame and the tool frame.

It is interesting to note that the actual position and orientation of the end effector always remains a constant when expressed in the tool frame of reference. This is because the tool space is a non-inertial space. Its frame of reference is attached to the end effector and so the frame of reference moves with the end effector, thus the coordinates of the end-effector expressed in this frame are always a constant as is shown by Figure 3.4.

In order to convert the joint space controller described in Section 3.2 to a tool space controller, the principle used in the previous section can be applied. There are some difficulties associated with this process. Due to the non-inertial properties of the tool space reference frame, it is not feasible to define reference trajectories in this space. The infeasibility arises from the fact that the reference frame is moving and its exact

position and orientation with respect to some inertial frame of reference is not known for every instant of time. Thus without a priori knowledge of the position and orientation of the tool frame during the planning of the trajectory, one cannot specify the reference trajectory in the tool space.

To overcome this a priori lack of knowledge of the position and orientation of the tool frame during the planning phase, the desired trajectories can be specified in an inertial reference frame like the task frame. When the robot arm is in motion, the exact position and orientation in the tool frame can be calculated by measuring the joint angles q . The velocity in the tool frame also can be determined by measuring the joint angle velocities $\dot{q}$. Using the joint angle and velocity measurements the exact location of the end-effector frame can be determined online. Hence, the homogenous transformation relating the task space and the tool space can now be calculated. This homogenous transformation can now be used to convert the desired trajectory already specified in the task space to a desired trajectory in the tool space. This conversion of desired trajectory from the task space to the tool space needs to be done online. By specifying the desired trajectories in the task space, the advantage of the robot task being represented as end-effector positions and orientations in the world frame can be retained as is done in the task space controller.

Let h_α , y_α be the homogenous coordinates (rotation matrices) of a point represented in the tool space and task space respectively. h_α , y_α are both 4×4 matrices. Let T be a 4×4 homogenous transformation that relates the homogenous task space coordinates to the homogenous tool space coordinates. Thus,

$$h_\alpha = Ty_\alpha. \quad (3.4.1)$$

Using the above equation the desired coordinates expressed in the task space can be converted to the tool space. Let $\mathfrak{S}_0$ be the tool OAT Jacobian relating the joint velocities to the velocity in the tool frame. Thus,

$$\dot{h} = \mathfrak{S}_0 \dot{q}. \quad (3.4.2)$$

The acceleration in the tool space can be derived by differentiating the above Equation 3.4.2 once with respect to time:

$$\ddot{h} = \dot{\mathfrak{S}}_0 \dot{q} + \mathfrak{S}_0 \ddot{q}. \quad (3.4.3)$$

Using the above equation the robot dynamics in tool space can be derived from the joint space dynamics as:

$$D(q)\mathfrak{S}_0^{-1}(\ddot{h} - \dot{\mathfrak{S}}_0 \dot{q}) + C(q, \dot{q}) + G(q) + J^T f = \tau. \quad (3.4.4)$$

From the desired task space trajectories, the desired position, velocity and acceleration in the task frame (y_{des} , $\dot{y}_{des}$ and $\ddot{y}_{des}$) are known for every time instant. Also the actual positions and velocities of the individual joints ($q, \dot{q}$) can be measured for every time instant. Note that the actual position and velocities of the joints are not known during the planning phase. Using the Equations (3.4.1, 3.4.2 and 3.4.3), the actual as well as the desired position, velocity and acceleration of the robot in the tool frame can be calculated online. With the knowledge of the actual and desired values of position, velocity and acceleration the control law in the tool space can be formulated as:

$$\tau = D(q)\mathfrak{S}_0^{-1}(\nu - \dot{\mathfrak{S}}_0\dot{q}) + C(q, \dot{q}) + G(q) + J^T f, \quad (3.4.5)$$

where ν is the vector of auxiliary inputs and it is defined as:

$$\nu = \ddot{h}_{des} + K_v(\dot{h}_{des} - \dot{h}) + K_p(h_{des} - h). \quad (3.4.6)$$

Figure 3.5 shows the block diagram of the tool space controller. Notice that the desired trajectories are specified in the task frame and then converted to the tool frame using coordinate transformations. The actual position and velocity of the robot is measured in the joint space and then converted to the tool space using the forward kinematics and coordinate transformations. The inverse of the tool space OAT Jacobian, $\mathfrak{S}_0^{-1}$, is used to decouple the tool space axes.

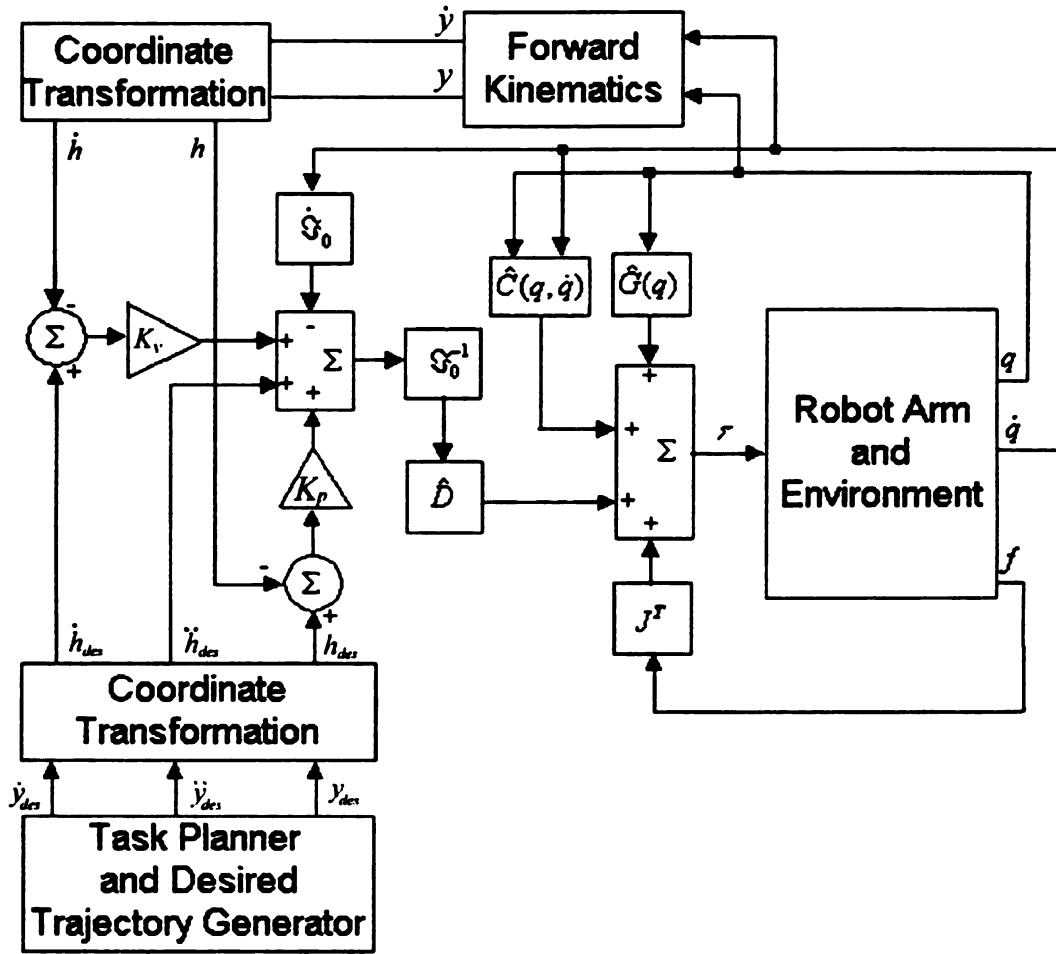


Figure 3.5: The tool space computed torque controller.

The combined system of the manipulator and the controller can now be expressed as:

$$(\ddot{h}_{des} - \ddot{h}) + K_v(\dot{h}_{des} - \dot{h}) + K_p(h_{des} - h) = 0. \quad (3.4.7)$$

Let the error in the tool space be expressed as:

$$e_h = h_{des} - h. \quad (3.4.8)$$

Thus, the error dynamics in the tool space given by Equation 3.4.7 can be formulated as:

$$\ddot{e}_h + K_v \dot{e}_h + K_p e_h = 0. \quad (3.4.9)$$

Using the state space representation with $x = [e_h \quad \dot{e}_h]^T$ as the state vector, the above Equation 3.4.9 can be written as:

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -K_v & -K_p \end{bmatrix} x \quad (3.4.10)$$

It is easy to prove that the error e_h in the tool space asymptotically tends to zero for appropriate choices of the gains K_p and K_v . Note that for the above error dynamics in Equation 3.4.10 to be stable the values of the gains K_p and K_v must be positive.

In this section, the control law for position control in the tool space has been formulated. In the next section the relationships between the joint space, task space and the tool space are explained. These relationships are essential for the implementation of the tool space controller as the actual and desired positions and velocities that are specified in the task space need to be converted to the tool space at every time step.

3.5 Relationship between the Task Space and Tool Space

For the tool space controller described in the previous section, there is a need to transform the desired positions and velocities specified in the task space to the tool space. Also the joint positions and velocities measured in the joint space need to be transformed

to the tool space. The need to transform these variables to the tool space arises from the fact that the error in position and the error in velocity in the tool space are required to drive the controller. Also the desired positions cannot be specified in the tool coordinate system, as its location is not known a priori during the planning phase. These transformations need to be carried out at the sampling frequency *i.e.*, they need to be calculated for each sample time. Figure 3.6 shows the positions of the task frame and the tool frame. It also depicts the actual and the desired positions of the end effector of the robot arm. The coordinates (position and orientation) of a point in the task space are converted to its tool space coordinates with the help of the homogenous transformation ${}^{task}T_{tool}$. This section describes the equations and procedures that are used to calculate the position and velocity error and the desired acceleration in the tool space.

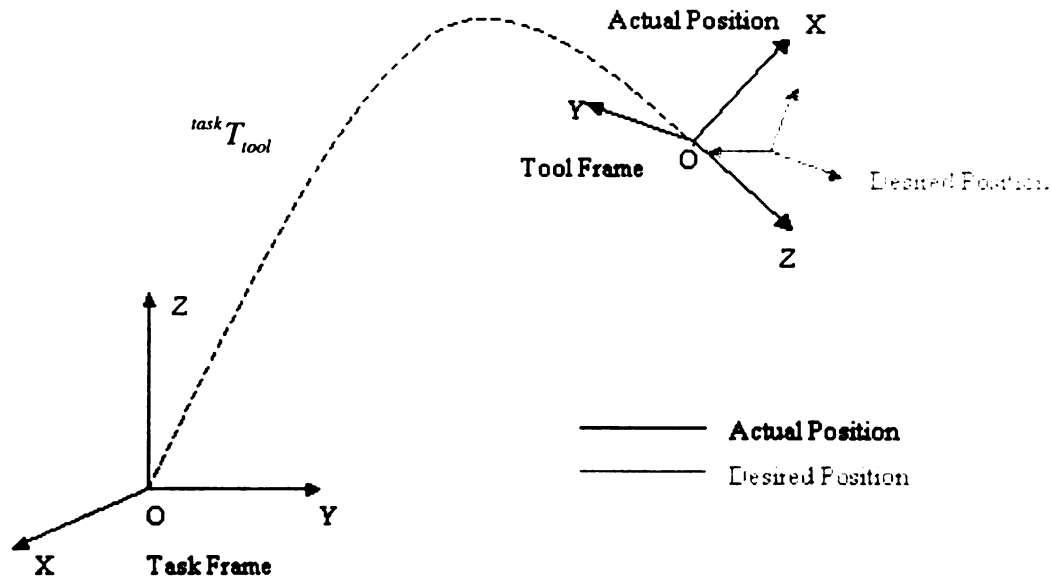


Figure 3.6: The relationship between the task space and tool space frame of reference.

3.5.1 Position error in the tool frame

The position of the robot in the joint space, task space and the tool space are denoted by $q = (q_1, q_2, q_3, q_4, q_5, q_6)$, $y = (x, y, z, O, A, T)^T$ and $h = (x, y, z, O, A, T)^T$ respectively. The homogenous position coordinates in the task space and the tool space are denoted by 4×4 matrices (y_α and h_α), which have form

$$\begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}_{4 \times 4},$$

where R is the 3×3 rotation matrix and p is the 3×1 position vector. Let $T(q)$ be the 4×4 homogenous transformation that transforms the positions in the task space to the tool space. Thus

$$h_\alpha = T(q)y_\alpha. \quad (3.5.1)$$

Given the rotation matrix, R , of the homogenous coordinates of a point in a particular frame of reference the O, A, T angles (coordinates) of that point in that frame of reference can be found using the equations described in [68]. Also the rotation matrix, R , can be derived given the O, A, T angles of a point.

To convert the desired positions specified in the task frame to the tool frame, first the homogenous coordinates of the point in the task frame *i.e.*, y_α , are derived from the task space coordinates y using the equations given in [68]. From the Equation 3.5.1, the homogenous coordinates in the tool frame *i.e.*, h_α can be found using the transformation

Tig

fou

coe

at

d

k

to

an,

$T(q)$, which is calculated online. The coordinates in the tool frame, h , can now be found by converting the homogenous tool frame coordinates, h_a , to the tool frame coordinates using the equations described in [68].

To find the actual position of the robot in the tool frame the same procedure used above can be applied. In this case first the homogenous coordinates in the task frame are derived from the joint angles using the forward kinematics equations. With the knowledge of the actual and desired position in the tool frame the error in position in the tool frame can be defined as:

$$e = h_{des} - h.$$

3.5.2 Velocity error in the tool frame.

A general 6×1 velocity vector *i.e.*, $\dot{y}$ or $\dot{h}$, in a three-dimensional Euclidean space such as the task space or the tool space is comprised of two parts. The first is a 3×1 linear component of the velocity denoted as v and it can be written as $v = (\dot{x}, \dot{y}, \dot{z})^T$. The second part is a 3×1 vector and it can be comprised of any general representation of the orientation velocity. The orientation representation chosen in this thesis is the OAT representation described in [68]. Thus the second part of the velocity vector is formed as $(\dot{O}, \dot{A}, \dot{T})^T$. Now the velocity vectors in the task and tool spaces can be expressed as $\dot{y} = (\dot{x}, \dot{y}, \dot{z}, \dot{O}, \dot{A}, \dot{T})^T$ and $\dot{h} = (\dot{x}, \dot{y}, \dot{z}, \dot{O}, \dot{A}, \dot{T})^T$ respectively.

Let $\omega = (\omega_x, \omega_y, \omega_z)^T$ be a 3×1 angular velocity vector which represents the angular velocity in the x , y and z directions in a three-dimensional Euclidean space.

Dep

wh

vec

ve

3:

a

T

h

las

spe

Depending on the choice of the orientation representation, a linear transformation, Φ , which is a function of O, A and T , can be defined which transforms the angular velocity vector, $\omega = [\omega_x \ \omega_y \ \omega_z]^T$, in a three-dimensional Euclidean space to the orientation velocity *i.e.*, $(\dot{O}, \dot{A}, \dot{T})^T$. The derivation of Φ is discussed in [68] and [58]. Thus Φ is a 3×3 matrix and it defines the relation between ω and $(\dot{O}, \dot{A}, \dot{T})$ as:

$$\begin{bmatrix} \dot{O} \\ \dot{A} \\ \dot{T} \end{bmatrix} = \Phi \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}, \quad (3.5.2)$$

and

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \Phi^{-1} \begin{bmatrix} \dot{O} \\ \dot{A} \\ \dot{T} \end{bmatrix}, \quad (3.5.3)$$

if Φ^{-1} exists.

The velocity error in the tool frame is defined as:

$$\dot{e} = \dot{h}_{des} - \dot{h}.$$

The desired velocity, $\dot{y}_{des}$, is specified in the task frame. To find the desired velocity, $\dot{h}_{des}$ in the tool frame, one needs to find the transformation that relates the velocity in the task frame to the velocity in the tool frame. The rotational velocity in the task frame is specified as a 3×1 velocity vector in the OAT orientation representation as $\dot{O}_{des}$, $\dot{A}_{des}$ and

$\dot{T}_{des}$. Let the transformation $T(q)$ which relates a position in the task frame to a position in the tool frame is defined as:

$$T = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Here $[n \ s \ a]$ represents the 3×3 rotation matrix and p represents the 3×1 position vector. Let Φ_{task} and Φ_{tool} be the linear transformations that relate the *OAT* velocities $(\dot{O}, \dot{A}, \dot{T})$ to the angular velocities ω , $(\omega_x, \omega_y, \omega_z)$, in their respective three-dimensional Euclidean spaces, *i.e.*, the task space and the tool space. The relation between the task space velocity and the tool space velocity vectors can now be formulated as (from [57]):

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{O} \\ \dot{A} \\ \dot{T} \end{bmatrix}_{tool} = \underbrace{\begin{bmatrix} I & 0 \\ 0 & \Phi_{tool} \end{bmatrix} \cdot \begin{bmatrix} n_x & n_y & n_z & (p \times n)_x & (p \times n)_y & (p \times n)_z \\ s_x & s_y & s_z & (p \times s)_x & (p \times s)_y & (p \times s)_z \\ a_x & a_y & a_z & (p \times a)_x & (p \times a)_y & (p \times a)_z \\ 0 & n_x & n_y & n_z \\ & s_x & s_y & s_z \\ & a_x & a_y & a_z \end{bmatrix} \cdot \begin{bmatrix} I & 0 \\ 0 & \Phi_{task}^{-1} \end{bmatrix}}_{\Lambda} \cdot \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{O} \\ \dot{A} \\ \dot{T} \end{bmatrix}_{task}.$$

Using the above equation the tool space desired velocity, $\dot{h}_{des}$, can be derived from $\dot{y}_{des}$, the desired velocity specified in the task space. Also the actual velocity in the tool space, $\dot{h}$, can be derived from $\dot{y}$, the actual velocity in the task space, similarly. The above equation can be written as:

$$\dot{h}_{des} = \Lambda \dot{y}_{des} \quad (3.5.4)$$

Differentiating the above equation once with respect to time the relation between the acceleration in the task space and the tool space can now be derived as:

$$\ddot{h}_{des} = \Lambda \ddot{y}_{des} + \dot{\Lambda} \dot{y}_{des} \quad (3.5.6)$$

Using the above equation the desired acceleration in the tool space can be calculated. The linear controller output, v , can now be calculated using $h, \dot{h}, \ddot{h}_{des}, \dot{h}_{des}, \ddot{h}_{des}, K_p$ and K_v using the Equation 3.4.6.

It should be noted that the transformation Λ is composed of 3 separate matrices as:

$$\Lambda = \begin{bmatrix} I & 0 \\ 0 & \Phi_{tool} \end{bmatrix} \Gamma \begin{bmatrix} I & 0 \\ 0 & \Phi_{task}^{-1} \end{bmatrix}$$

Form Equation 3.5.4, it can be seen that Γ is non-singular. This is because the 3×3 rotation matrix $R = [n \quad s \quad a]$ is non-singular by definition.

Chapter 4

4.1 Hybrid Force/Position Control

The contact forces generated between the robot and environment, when the robot comes in contact with the environment, need to be controlled. The idea of hybrid force/position control, proposed by Raibert and Craig [4], attempts at controlling the interaction forces between the manipulator and the environment in certain directions, while simultaneously controlling the position and orientation of the manipulator in other directions. It is based on Mason's concept of using a set of position and force constraints depending on the mechanical and geometric characteristics of the performed task [8].

The task to be performed is broken down into elemental components that are defined by a particular set of contacting surfaces. A set of 'Natural Constraints' is associated with each elemental component that results from the particular mechanical and geometric characteristics of the task configuration. Additional constraints, called 'Artificial Constraints', are introduced in accordance with these criteria to specify desired motions or force patterns in the task configuration. Thus for each desired trajectory in either force or position, an artificial constraint is defined.

For each task configuration a generalized surface can be defined in a constraint space having n degrees of freedom, with position constraints along the normal to this surface and force constraints along the tangents to this surface. These two types of

constraints, force and position, partition the degrees of freedom of possible hand motions into two orthogonal sets that must be controlled according to different criteria.

The natural constraints are used to partition the degrees of freedom into a position-controlled subset and a force-controlled subset. The artificial constraints are used to specify the desired position and force trajectories. A compliance selection vector, s , which is a binary N -tuple, specifies the degrees of freedom in the constraint frame that are under force control (indicated by $s_i = 1$). Setting $s_j = 0$ specifies the degrees of freedom under position control. Also a compliance selection matrix S is defined as $S = \text{diag}(s)$.

A general conceptual organization of the hybrid controller is given in Figure 4.1. In the figure there are two complementary sets of feedback loops, position and force, each with its own sensory system and control law that are controlling the same plant *i.e.*, the manipulator. The force and position controlled directions are selected with the help of the selection matrix S . In this control strategy the actuator drive signal for each joint corresponds to that particular joint's instantaneous contribution to satisfying each position and force constraint. Thus the actuator control signal for the i^{th} joint has N components – one for each position-controlled direction and one for each force controlled direction in the constraint frame. The control law is given as:

$$\tau_i = \sum_{j=1}^N \{ \Gamma_{ij} [s_j \Delta f_j] + \psi_{ij} [(1 - s_j) \Delta x_j] \},$$

where τ_i is the torque applied by the i^{th} actuator, Δf_j is the force error in the j^{th} degree of freedom of the constraint frame, Δx_j is the position error in the j^{th} degree of freedom of the constraint frame, Γ_{ij} and ψ_{ij} are the force and position compensation functions, respectively, for the j^{th} input and this i^{th} output and s_j is the j^{th} component of the compliance selection vector.

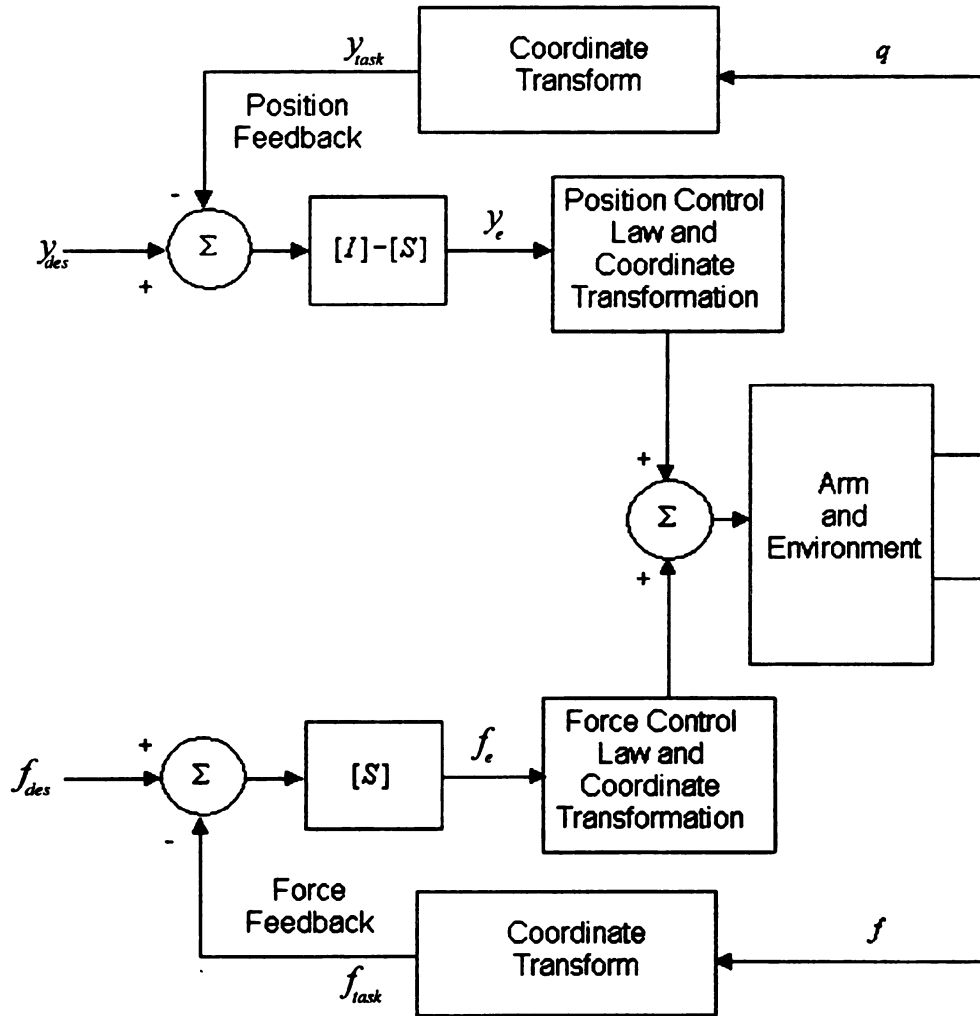


Figure 4.1 Hybrid force/position controller.

Using this approach the interaction forces between the manipulator and the environment can be controlled simultaneously with position of the manipulator in free space. However, it is extremely tedious and impractical to determine the force and position compensation functions, Γ_{ij} and ψ_{ij} , for all possible configurations of the manipulator and also a lot of inverse kinematics calculations will be required in this process. Further, as shown in [34], this sort of formulation for hybrid force/position control problem gives rise to inherent kinematic stability issues. The instability occurs due to the interaction of the Jacobian matrix J and the selection matrix S . A conceptually simpler control strategy for hybrid force/position control of the manipulator, implemented directly in the task space, is discussed in the following section.

4.2 Hybrid Force/Position Controller in the Task Space

Using the nonlinear feedback linearization control law from Equation 3.3.4 the dynamics of the robot system can be linearized and decoupled as:

$$\ddot{y} = v, \quad (4.2.1)$$

where v is the vector of auxiliary inputs (commanded acceleration).

The above system is decoupled with respect to the task space axes. Thus the commanded acceleration in any of the six directions of the task space can be controlled independently of the rest of the task space directions. This decoupling is very important for the realization of simultaneous control of forces in the constrained directions and position in the unconstrained directions after contact is established. Let the superscript u

denote a quantity in the unconstrained directions and the superscript c denotes a quantity in the constrained directions. The Equation 4.2.1 can be written as $\ddot{y}^u = v^u$ and $\ddot{y}^c = v^c$. [52]

Before impact, position control is required in both the unconstrained as well as constrained directions. Applying the PD position control law from Equation (3.3.6)

$$v^u = \ddot{y}_{des}^u + K_v(\dot{y}_{des}^u - \dot{y}^y) + K_p(y_{des}^u - y^u) \quad t \leq t_{sw}, \quad (4.2.2)$$

$$v^c = \ddot{y}_{des}^c + K_v(\dot{y}_{des}^c - \dot{y}^y) + K_p(y_{des}^c - y^c) \quad t \leq t_{sw}, \quad (4.2.3)$$

where t_{sw} is the instant of the detection of impact. By applying the above control schemes, the error dynamics in the unconstrained as well as constrained directions before impact can be expressed as:

$$\ddot{e}^u + K_v\dot{e}^u + K_p e^u = 0, \quad (4.2.4)$$

$$\ddot{e}^c + K_v\dot{e}^c + K_p e^c = 0, \quad (4.2.5)$$

where $e^u = y_{des}^u - y^u$ and $e^c = y_{des}^c - y^c$. Using conventional, state space pole placement methods, the desired transient and steady state responses can be achieved.

After detection of impact, position control is still needed in the unconstrained directions. Applying a PD position control law similar to Equation 4.2.2 the desired transient and steady state responses can be achieved.

$$v^u = \ddot{y}^u + K_{vu}(\dot{y}_{des}^u - \dot{y}^y) + K_{pu}(y_{des}^u - y^u) \quad t \geq t_{sw} \quad (4.2.6)$$

It is clear, from the above equations that the error dynamics of the position tracking of the closed-loop system are asymptotically stable.

For the constrained directions explicit force control is used. This makes the knowledge of the exact force model unnecessary for contact force control. The objective of the force control scheme is to track a desired force trajectory in the constrained directions. A force control scheme of the following form is considered:

$$\dot{v}^c = K_{vf}(\dot{f}_{des}^c - \dot{f}^c) + K_{pf}(f_{des}^c - f^c) + K_{if} \int_{t_{sw}}^t (f_{des}^c - f^c) dt \quad t \leq t_{sw}, \quad (4.2.7)$$

where K_{pf} , K_{vf} and K_{if} are the position derivative and integral gains for achieving the desired performance and f_{des}^c and f^c are the desired and the actual forces in the constrained directions expressed in the task space coordinates. Using this task space hybrid controller one can control the force applied by the robot in any of the task space directions independently of the rest of the directions.

Figure 4.2 shows the diagram of the task space hybrid force/position controller. The matrix $[S]$ is the selection matrix that is used to select the directions along which force is controlled. The matrix $[I - S]$ selects the directions along which position is to be controlled. The desired position trajectory generator generates the desired position trajectories and the desired force trajectory generator generates the desired force trajectories in the task space.

The sufficient conditions for stability for this controller are provided in [36]

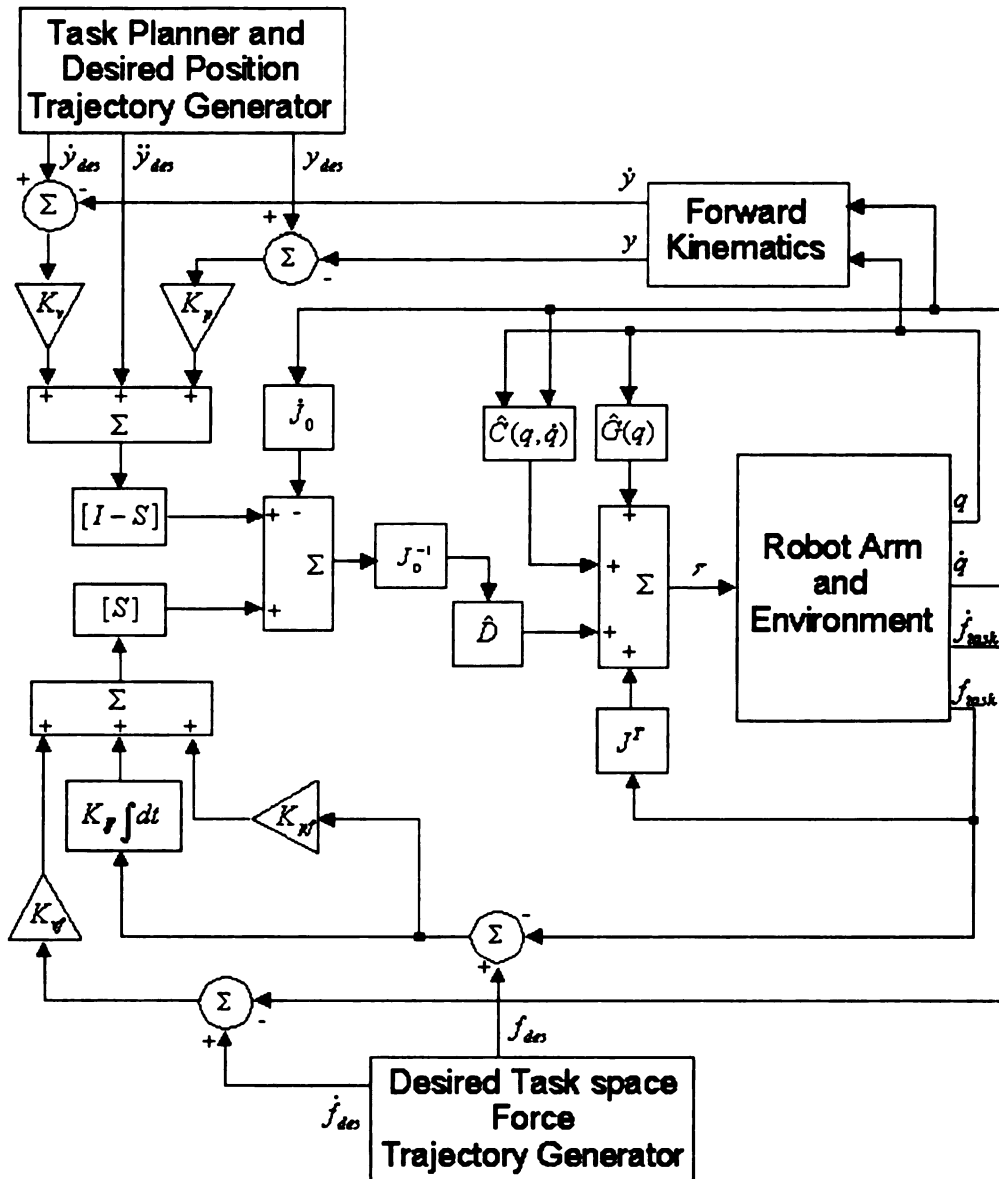


Figure 4.2 Task space hybrid force/position controller.

A major drawback of this strategy is the restrictions on the directions in which the force can be controlled. This means that for this strategy to work the constraint surface must be aligned along the task space directions in order to completely control the forces

of interaction. As pointed out by Duffy [50], this is a big restriction when dealing with objects that are not aligned with the task space axes. For a constraint surface that is not aligned along the task space axes the interaction force cannot be controlled unless the controller axes are aligned with the object surface. This means that the controller equation needs to be changed online to guarantee the alignment of the object surface and the task space axes, which may lead to instability of the system. Also this hybrid force position control scheme is not robust to the occurrence of impact in a direction where motion control has been planned [69]. In the case of such an impact, the end effector is not compliant along that direction. This will give rise to large interaction forces between the end effector and the environment, which cannot be controlled and will damage the manipulator or the environment. Further this scheme requires a detailed knowledge of the environment geometry, and therefore it is unsuitable for use in poorly structured environments and to the occurrence of unplanned objects.

For the reasons mentioned above it is not practical to implement hybrid force/position control that can interact with environments whose shape is not known a priori. The next section presents the design of a new controller that can overcome the abovementioned drawbacks of a task space controller.

4.3 Hybrid Force/Position Controller in the Tool Space

It is important to note that the above idea of hybrid force/position control is based on the notion of 'orthogonal complements'. This means that the force and the position are controlled in orthogonal (non-overlapping) subspaces of the task space. Continuing this idea of orthogonal complements further, hybrid force/position control can be

implemented in the tool space. The advantage of this method is that the tool space is not a fixed (inertial) space. It is constantly moving along with the movement of the end-effector. Thus, if the tool frame can be dynamically positioned in such a manner that its axes align with the direction of the force to be controlled, then the force can be controlled in any arbitrary direction.

However this method does pose some challenges in implementation as described in Section 3.4. This is because the frame in which the control is implemented is a non-inertial frame and the exact location of this frame is not known a priori during the planning phase. Thus specifying desired positions, orientations, forces and torques in the tool reference frame poses a challenge. But taking into account the fact that the exact location of the tool frame is known during the execution phase, the desired positions and orientations could be specified in an inertial frame, e.g. the task frame, and then converted to the tool frame online using coordinate transformations. This conversion is abetted by the knowledge of the exact location of the tool frame with respect to the task frame during the execution phase.

The desired force input is specified directly in the tool reference frame. This is because the force to be applied by the manipulator is constant in the 'z' direction of the tool reference frame, irrespective of the position and orientation of the tool reference frame. The interaction force is measured using a wrist mounted force torque sensor and can be transformed directly in the tool reference frame with the help of a linear transformation.

From Section 3.4, the dynamics of the robot manipulator expressed in the tool space can be written as:

$$\tau = D(q)\mathfrak{S}_0^{-1}(\ddot{h} - \dot{\mathfrak{S}}_0\dot{q}) + C(q, \dot{q}) + G(q) + J^T f .$$

Using the control law formulated in Equation 3.4.5,

$$\tau = D(q)\mathfrak{S}_0^{-1}(v - \dot{\mathfrak{S}}_0\dot{q}) + C(q, \dot{q}) + G(q) + J^T f ,$$

The manipulator system can now be linearized and decoupled as:

$$\ddot{h} = v , \tag{4.3.1}$$

where v is the vector of auxiliary inputs (commanded acceleration).

The above system is decoupled with respect to the tool frame axes. Thus the commanded acceleration in any of the tool space directions can be controlled independently of the rest of the tool space directions. Let the superscript u denote a quantity in the unconstrained directions of the tool space and the superscript c denotes a quantity in the constrained directions of the tool space. Thus the Equation 4.3.1 can be written as two separate equations:

$$\ddot{h}^u = v^u \text{ and}$$

$$\ddot{h}^c = v^c .$$

For the unconstrained directions, position control is required before and after impact. Applying the tool space PD position control law from Equation 3.4.6 in the unconstrained directions:

$$\ddot{v}^u = \ddot{h}_{des}^u + K_v^u(\dot{h}_{des}^u - \dot{h}^u) + K_p^u(h_{des}^u - h^u) \quad (4.3.2)$$

Using the above Equation 4.3.2, the position error dynamics in the unconstrained space can be written as:

$$\ddot{e}^u + K_v^u \dot{e}^u + K_p^u e^u = 0, \quad (4.3.3)$$

where $e^u = h_{des}^u - h^u$. Thus, desired transient and steady state responses can be achieved using conventional pole placement methods.

For the constrained space, before impact, position control needs to be applied. After detection of impact, the interaction force needs to be controlled. PD position control in the tool space is used in the constrained directions before the detection of impact. After the detection of impact, explicit force control is used. Thus the constrained space auxiliary input can be expressed as:

$$\ddot{v}^c = \ddot{h}_{des}^c + K_v^c(\dot{h}_{des}^c - \dot{h}^c) + K_p^c(h_{des}^c - h^c) \quad t \leq t_{sw}, \quad (4.3.4)$$

$$\ddot{v}^c = K_{vf}(\dot{f}_{des}^c - \dot{f}^c) + K_{pf}(f_{des}^c - f^c) + K_{if} \int_{t_{sw}}^t (f_{des}^c - f^c) dt \quad t \leq t_{sw}, \quad (4.3.5)$$

where K_{pf} , K_{vf} and K_{if} are the position derivative and integral gains for achieving the desired performance and f_{des}^c and f^c are the desired and the actual forces in the constrained directions expressed in the tool space coordinates.

In the tool space hybrid controller the desired position trajectories in the constrained as well as unconstrained space are initially specified in the task space frame of reference. Using the knowledge of the exact position and velocity of the joints of the manipulator, q and $\dot{q}$, the desired trajectories, y_{des} , $\dot{y}_{des}$ and $\ddot{y}_{des}$, are converted to the tool reference frame online. Thus, during the planning phase the desired position trajectories are specified as y_{des} , $\dot{y}_{des}$ and $\ddot{y}_{des}$, and the desired tool space trajectories h_{des} , $\dot{h}_{des}$ and $\ddot{h}_{des}$ are derived from them online using coordinate transformations given in Section 3.5. The force trajectory, f_{des} , is specified directly in the tool space. This is because the force to be controlled is along the tool space 'z' direction. So the force trajectory can be directly specified in the tool frame of reference along the 'z' direction.

Figure 4.3 shows the implementation of the tool space hybrid force/position controller. Note that the desired positions and orientations are generated in the task space by the task space trajectory generator. These are then converted to the tool space using coordinate transformations and the Jacobian matrix relating the task space and tool space velocities. The desired force trajectories are specified in the tool space. The matrices $[S]$ and $[I - S]$ are used to partition the tool space into two orthogonal position controlled and force controlled subspaces.

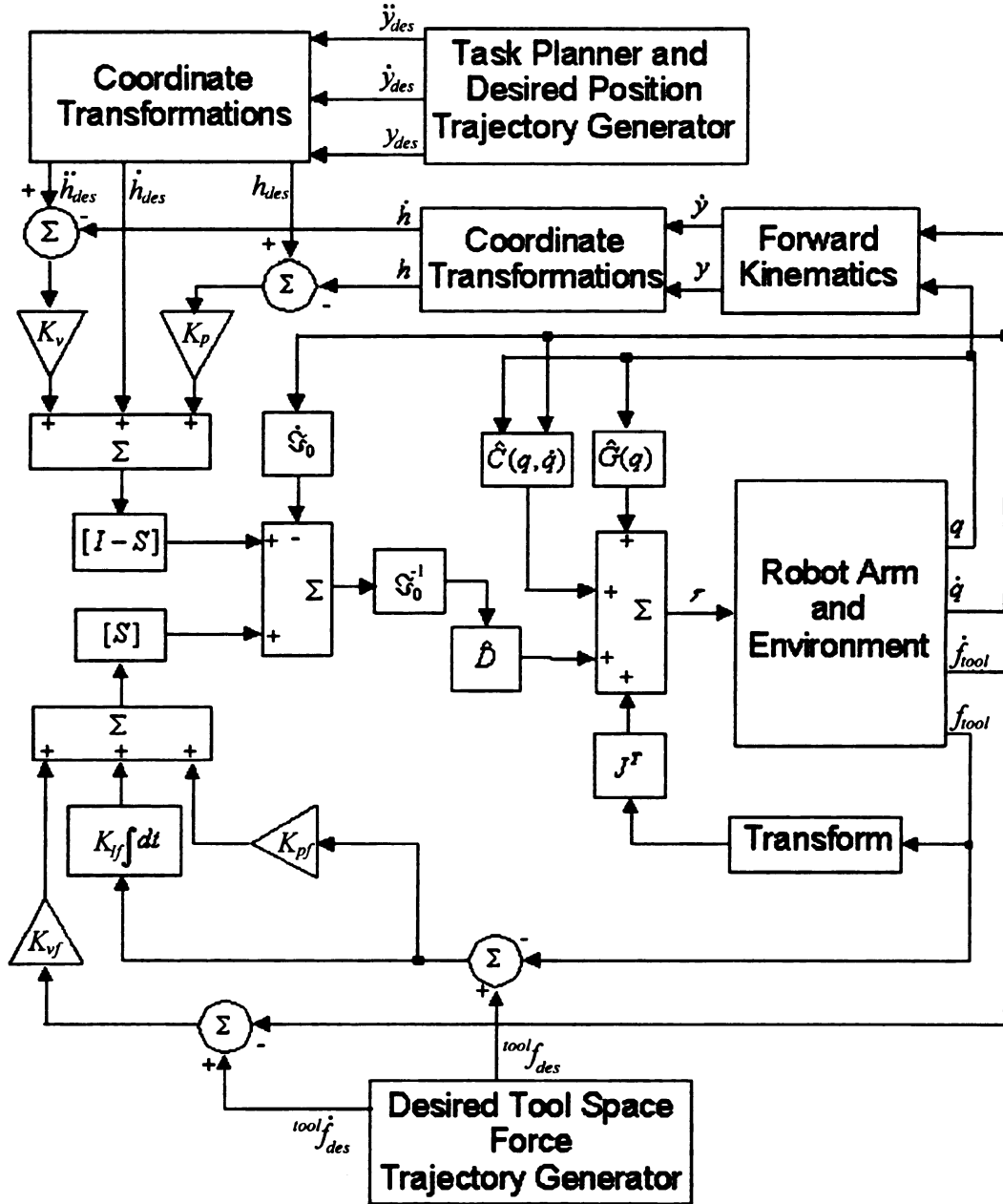


Figure 4.3: The tool space hybrid force/position controller.

In order to implement 'robust' hybrid force/position control in the tool space, the direction of the tool space frame of reference that is under force control, *i.e.*, the tool z -axis must be aligned with the direction of the interaction force. If this is not the case, all

the components of the interaction force cannot be controlled and that could either lead to loss of contact or to excessively large interaction forces that would damage the manipulator or the environment. Thus the alignment of the constrained direction of the tool frame and the direction of the interaction force must be ensured. For that reason it is imperative to implement the orientation control of the manipulator in such a manner that the alignment is achieved.

The wrist mounted force sensor senses the direction as well as the magnitude of the interaction force at the tip of the manipulator. Using linear coordinate transformations, the sensed force direction, which is in the wrist coordinate system, is converted to the tool coordinate frame with the help of the joint angles and joint velocity. The tool space controller uses this (tool space) direction to orient the end-effector along the direction of the sensed force. Thus, the constraint frame (end-effector frame) will be aligned along the direction of the interaction force. This will in turn ensure that the interaction forces can be fully regulated. The gains for the orientation control loop are set at moderately high values for a fast response. The following section analyses the stability of this controller.

4.4 Stability Analysis of the Tool Space Controller

In Section 3.3, it has been shown that the task level controller is asymptotically stable in the entire workspaces except around singularities where J_0^{-1} is not defined. In Section 3.4 a similar stability analysis was presented for the tool level position controller. The desired trajectories for the tool space controller are specified in the task space, which

are then converted to the tool space online. Thus, if the transformation between the task space and the tool space is not defined, then one cannot even begin to talk about stability of the tool space control law. But, here we note that, the tool and task spaces are 3-dimensional Cartesian spaces that are related to each other by a linear, bounded, non-singular homogenous transformation. Thus, for every point in the task space there exists a corresponding representation in the tool space. Further, as shown in Section 3.5, the velocities in the task space and tool space are related to each other by a linear, bounded, non-singular transformation. In this section it will be proved that a tool space position controller is stable and further stability analysis for the tool space hybrid force/position controller is presented.

The tool space position controller is given as (from Equation 3.4.5):

$$\tau = D(q)\mathfrak{S}_0^{-1}(\nu - \mathfrak{S}\dot{q}) + C(q, \dot{q}) + G(q) + J^T f, \quad (4.4.1)$$

where ν is the vector of auxiliary inputs and it is defined as:

$$\nu = \ddot{h}_{des} + K_v(\dot{h}_{des} - \dot{h}) + K_p(h_{des} - h). \quad (4.4.2)$$

Assuming the exact model of the manipulator is known and $\mathfrak{S}_0^{-1}$ exists, the error dynamics of the combined system of the manipulator and the controller can be written as (from Equation 3.4.9):

$$\ddot{e}_h + K_v\dot{e}_h + K_p e_h = 0. \quad (4.4.3)$$

Forming the state vector as $x = [e_h \quad \dot{e}_h]^T$, the above system can be written as:

$$\dot{x} = Ax \quad (4.4.4)$$

$$\text{where } A = \begin{bmatrix} 0 & 1 \\ -K_p & -K_v \end{bmatrix}$$

For the above system to be asymptotically stable about the origin, all the eigenvalues of the matrix A must have negative real parts. Thus if K_p and K_v are positive, the above system is asymptotically stable and $x \rightarrow 0$. Since $x = [e_h \quad \dot{e}_h]^T$, e_h and $\dot{e}_h$ tend to zero asymptotically and in turn the end-effector tracks the desired tool space trajectory.

Here it is noted that the above error dynamics were formulated in the robot tool space, while the desired trajectory was specified in the task space. In order to show that the robot tracks the trajectory in task space, we take help of the transformations that relate the task space positions and velocities to the tool space positions and velocities. The position and velocity in the task space and tool space are specified as 6×1 vectors as:

$$y = \begin{bmatrix} x \\ y \\ z \\ O \\ A \\ T \end{bmatrix}_{task}, \quad \dot{y} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{O} \\ \dot{A} \\ \dot{T} \end{bmatrix}_{task}, \quad h = \begin{bmatrix} x \\ y \\ z \\ O \\ A \\ T \end{bmatrix}_{tool} \quad \text{and} \quad \dot{h} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{O} \\ \dot{A} \\ \dot{T} \end{bmatrix}_{tool}$$

where y and h represent vectors in the task and tool spaces respectively.

Rotation matrices can represent the position and orientation of a rigid body in 3-dimensional Cartesian space. Let $[y]$ and $[h]$ denote the rotation matrix representations

of position and orientation of a point in the task space and tool space respectively. For every position vector y (h) there exists a unique rotation matrix $[y]$ ($[h]$) and vice versa. The rotation matrices for a point in the task space and the same point in the tool space are related by a non-singular homogenous transformation as:

$$[h]_{4 \times 4} = \begin{bmatrix} R_{3 \times 3} & p_{3 \times 1} \\ 0 & 0 & 0 & 1 \end{bmatrix}_{4 \times 4} [y]_{4 \times 4} . \quad (4.4.5)$$

Hence, the position in the task space is related to the position in the tool space by a non-singular, invertible (by definition) homogenous transformation. Let us assume that the transformation is bounded. This is a valid assumption because by definition $R_{3 \times 3}$ is always bounded and $p_{3 \times 1}$ is the position vector from the origin of the task space to the origin of the tool space. So we can assume that $p_{3 \times 1}$ is bounded if the region of interest is limited to the vicinity of the task space origin. Hence the transformation itself is bounded. Thus we can conclude that if the error in position in the tool space tends to zero, the position error in the task space also tends to zero.

Further, from Section 3.5, the velocities in the tool space and the task space are related by the following equation:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{O} \\ \dot{A} \\ \dot{T} \end{bmatrix}_{tool} = \begin{bmatrix} I & 0 \\ 0 & \Phi_{tool} \end{bmatrix} \cdot \underbrace{\begin{bmatrix} n_x & n_y & n_z & (p \times n)_x & (p \times n)_y & (p \times n)_z \\ s_x & s_y & s_z & (p \times s)_x & (p \times s)_y & (p \times s)_z \\ a_x & a_y & a_z & (p \times a)_x & (p \times a)_y & (p \times a)_z \\ 0 & n_x & n_y & n_z \\ s_x & s_y & s_z \\ a_x & a_y & a_z \end{bmatrix}}_{\Lambda} \cdot \begin{bmatrix} I & 0 \\ 0 & \Phi_{task}^{-1} \end{bmatrix} \cdot \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{O} \\ \dot{A} \\ \dot{T} \end{bmatrix}_{task}.$$

From the above equation, if we assume that Φ_{tool} and Φ_{task}^{-1} are non-singular and bounded then it can be proved that the transformation Λ is invertible and bounded. Hence if the velocity error in the tool space tends to zero, the velocity error in the task space also tends to zero. The assumption that Φ_{tool} and Φ_{task}^{-1} are non-singular and bounded is valid for most of the points in the Cartesian spaces except when $A = \pm \frac{\pi}{2}$. The above result is valid only for the points in the task space and tool space where $A \neq \pm \frac{\pi}{2}$.

Further in this section we will prove the stability for the tool space hybrid force/position controller. The equations of motion of the robot expressed in the tool space are given as:

$$\tau = D(q)\mathfrak{S}_0^{-1}(\ddot{h} - \dot{\mathfrak{S}}\dot{q}) + C(q, \dot{q}) + G(q) + J^T f \quad (4.4.6)$$

The tool space hybrid force/position controller is given by:

$$\tau = D(q)\mathfrak{S}_0^{-1}(v - \dot{\mathfrak{S}}\dot{q}) + C(q, \dot{q}) + G(q) + J^T f \quad (4.4.7)$$

where the auxiliary input v is given by:

$$v = [I - S](\ddot{h}_{des}^u + K_v^c(\dot{h}_{des}^u - \dot{h}^u) + K_p^c(h_{des}^u - h^u)) + [S](K_{vf}(\dot{f}_{des}^c - \dot{f}^c) + K_{pf}(f_{des}^c - f^c) + K_{lf} \int_{t_{rw}}^t (f_{des}^c - f^c) dt) \quad (4.4.8)$$

Here $[S]$ is the selection matrix that decides which tool space axes are under position control and which are under force control.

Combining the equations of the manipulator and the controller the linearized and decoupled equations of the complete system are given by:

$$\ddot{h} = v$$

The above equation can be expanded as:

$$[I - S]\ddot{h}^u + [S]\ddot{h}^c = [I - S](\ddot{h}_{des}^u + K_v^c(\dot{h}_{des}^u - \dot{h}^u) + K_p^c(h_{des}^u - h^u)) + [S](K_{vf}(\dot{f}_{des}^c - \dot{f}^c) + K_{pf}(f_{des}^c - f^c) + K_{lf} \int_{t_{rw}}^t (f_{des}^c - f^c) dt) \quad (4.4.9)$$

Equating the quantities in $[S]$ and $[I - S]$, we get two separate equations for the constrained and unconstrained directions respectively as:

$$\ddot{h}^c = K_{vf}(\dot{f}_{des}^c - \dot{f}^c) + K_{pf}(f_{des}^c - f^c) + K_{lf} \int_{t_{rw}}^t (f_{des}^c - f^c) dt \quad (4.4.10)$$

for the constrained directions and,

$$\ddot{h}^u = \ddot{h}_{des}^u + K_v^u(\dot{h}_{des}^u - \dot{h}^u) + K_p^u(h_{des}^u - h^u) \quad (4.4.11)$$

for the unconstrained directions.

For the unconstrained directions, the error dynamics can be expressed as:

$$\ddot{e}^u - K_v^u \dot{e}^u - K_p^u e^u = 0. \quad (4.4.12)$$

For appropriate choices of K_p^u and K_v^u the above error dynamics can be proved to be stable. For the constrained directions, let us assume that the actual acceleration, $\ddot{h}^c$, in these directions is zero. Thus the Equation 4.4.10 can be written as:

$$K_{vf}(\dot{f}_{des}^c - \dot{f}^c) + K_{pf}(f_{des}^c - f^c) + K_{if} \int_{t_{nv}}^t (f_{des}^c - f^c) dt = 0 \quad (4.4.13)$$

We can easily prove that the above Equation 4.4.13 is asymptotically stable for proper selection of K_{pf} , K_{vf} and K_{if} .

Thus we can prove that the combined system of the manipulator and the controller is asymptotically stable.

Chapter 5

5.1 Simulation Studies - Setup

Intensive simulations were conducted at the Robotics and Automation laboratory at Michigan State University in order to verify the theoretical results. A 6 DOF Puma 560 robot arm manipulator [68] was used for the simulation studies. The end effector was mounted with a jr3 force/torque sensor, which senses the forces and torques directly in the hand coordinate frame. The sampling period for the control software was set at 1ms.

Experimental comparisons between the existing Task Space Controller and the proposed Tool Space Controller for pure position control were conducted. Simulation studies for the proposed Tool Space Hybrid Force/Position Controller in contact with a compliant surface aligned along the manipulator reference frame were carried out. The results were compared with the existing Task Space Hybrid Force/Position Controller performing a similar task. Finally simulations were carried out on the proposed Tool Space Hybrid Force/Position Controller in contact with a constraint surface of unknown shape.

The constraint surfaces were simulated as linear spring dampers with large stiffness coefficients [70]. Thus the environment is modeled as:

$$F(t) = -K_e(x(t) - x_0),$$

where $F(t)$ is the force generated by the environment, $x(t)$ is the actual position of the robot, x_0 is the position of the wall and K_e is the stiffness coefficient. For the experiments K_e was taken to be 1000 N/m. Also the moments about the x , y and z axes were assumed to be zero for simplification purposes.

A frictionless contact was assumed between the end-effector and the environment. This scenario, though unrealistic, can be simulated in actual practice using a roller ball contact assembly.

5.2 Position Control Experiments

Simulations were carried out to compare the task space controller and the tool space controller for pure position control. To make the simulation more realistic dynamically, instead of some straight lines or movement on a horizontal plane, the desired path of the end effector is chosen as a circle with a constant angular velocity ω on a plane that is at an angle to the XOY plane. The desired orientations were taken to be continuous sinusoids. The desired paths were:

$$x_{des} = 0.707R \cos(\omega t) + 0.4$$

$$y_{des} = R \sin(\omega t) + 0.3$$

$$z_{des} = 0.707R \cos(\omega t) + 0.1$$

$$O_{des} = -0.5 \sin(\omega t) + 0.5$$

$$A_{des} = -0.5 \sin(\omega t) + 0.5$$

$$T_{des} = -0.5 \sin(\omega t) + 0.5$$

where $R = 0.25\text{ m}$ and $\omega = \frac{\pi}{8}$.

The same task was simulated for both the task level and the tool level controller. The desired and actual output position/velocity and error in position/velocity of the robot in the Cartesian task frame were plotted against time for both the controllers. The results of these experiments are shown in Figures 5.1 to Figure 5.4

Figure 5.1 - 5.4 show the plots of the experiment carried out for task level position control. It can be noted that the error in position for these experiments is of the order 10^{-3} to 10^{-4} meters. Better results can be achieved by proper tuning of the controller gains K_p and K_v for achieving desired dynamic performance. Figure 5.1 (a) - (f) show the actual and desired positions and orientations for the task level position controller. Figures 5.2 (a) – (f) depict the plots of the actual and desired velocity in the six independent directions of task space. From the above two figures we see that the desired and actual position and velocity are nearly the same, which indicates that the system has very good tracking properties.

Figure 5.3 (a) – (f) plots the position error for the task level position error. The errors in velocity for the task level position control are shown in Figure 5.4 (a) – (f). The errors in position and velocity are very small thereby indicating good dynamic response of the controller.

The same experiment was carried out for the tool space position controller. For comparison with the task space controller, the Cartesian task space positions and velocities for the tool space controller are plotted in Figures 5.5 and 5.6 respectively. The

desired positions and velocities for the tool space controller are specified in the task space frame of reference, which is an inertial frame. Hence, the error plots for position and velocity of the tool space controller are plotted in the Cartesian task space in the Figures 5.7 and 5.8 respectively.

Figure 5.5 (a) – (f) shows the actual and desired positions in the task space frame of reference. The actual and desired velocities are plotted in the task frame in Figure 5.6 (a) – (f) for the tool space position controller. The errors in position and velocity expressed in the task frame for this tool space controller are also plotted in Figures 5.7 and 5.8. Figure 5.7 (a) – (f) depict the position errors for this tool level controller expressed in the task frame. The velocity errors for the tool level controller expressed in the task frame of reference are shown in Figure 5.8 (a) – (f).

The gains for orientation control are kept moderately high to achieve a faster response for changes in the orientation. This would prevent excessively large interaction forces in the directions that are not under force control by aligning the tool space z axis with the direction of the sensed force.

The errors plots in the task space and the tool space are nearly of the same magnitude and also have a similar profile. This is because the task space and the tool space are related by a linear transformation. Numerous such experiments for position tracking were conducted, which gave similar results. Thus from these experiments it can be concluded that position control in the tool space can be implemented.

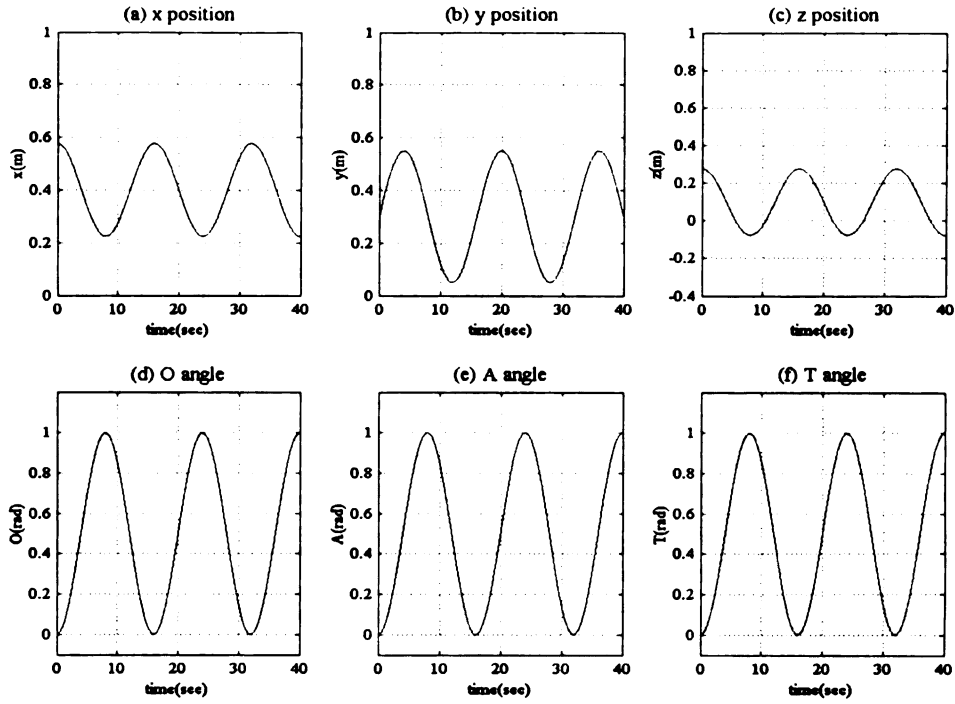


Figure 5.1: The actual and desired position and orientation for task level position control.

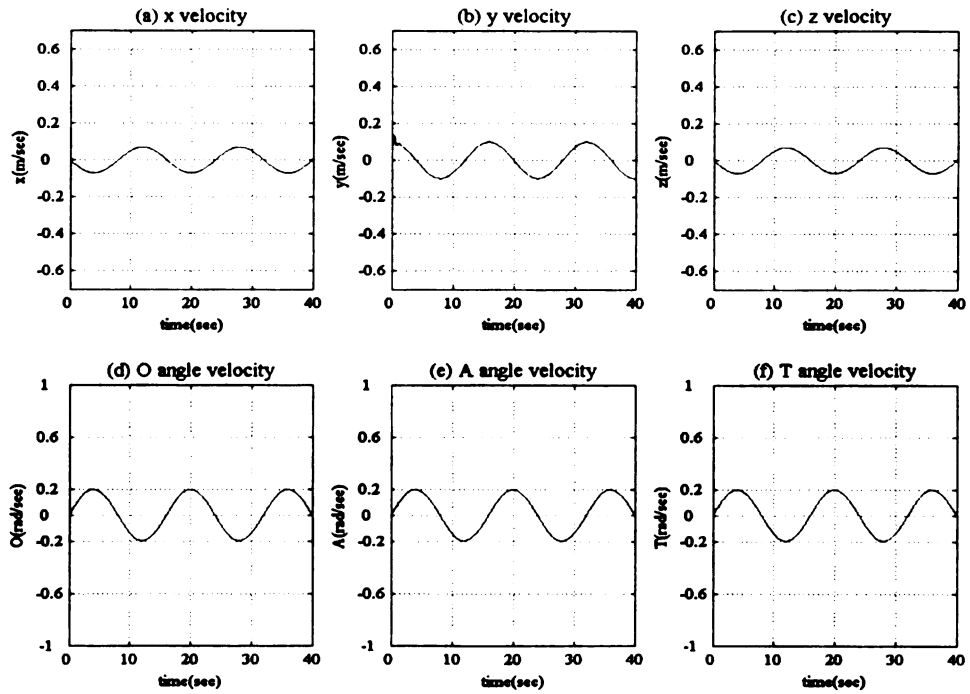


Figure 5.2: The actual and desired velocities for task level position control.

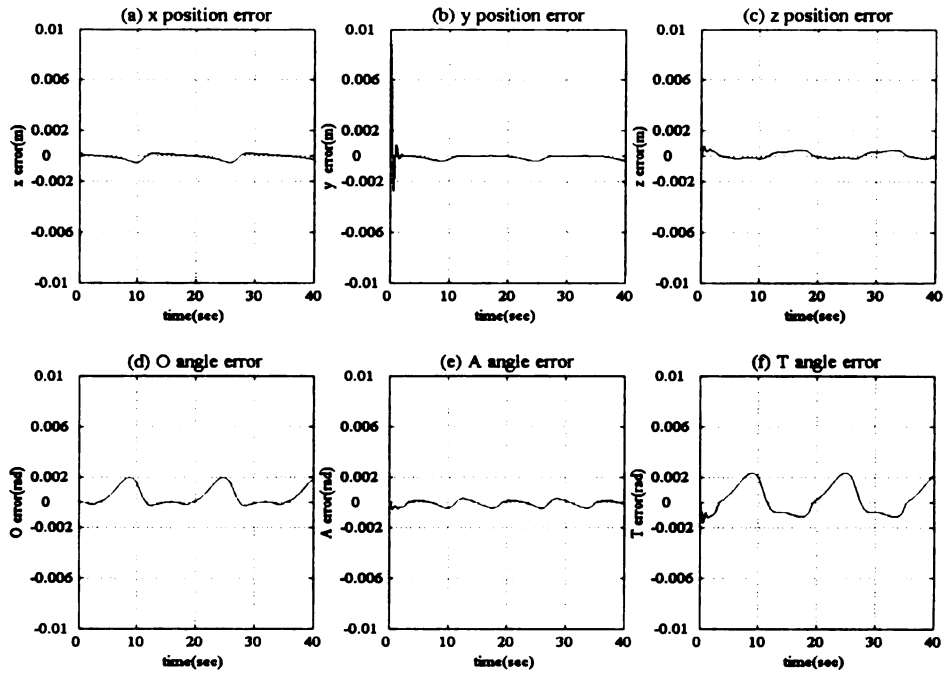


Figure 5.3: The error in position for task level position control.

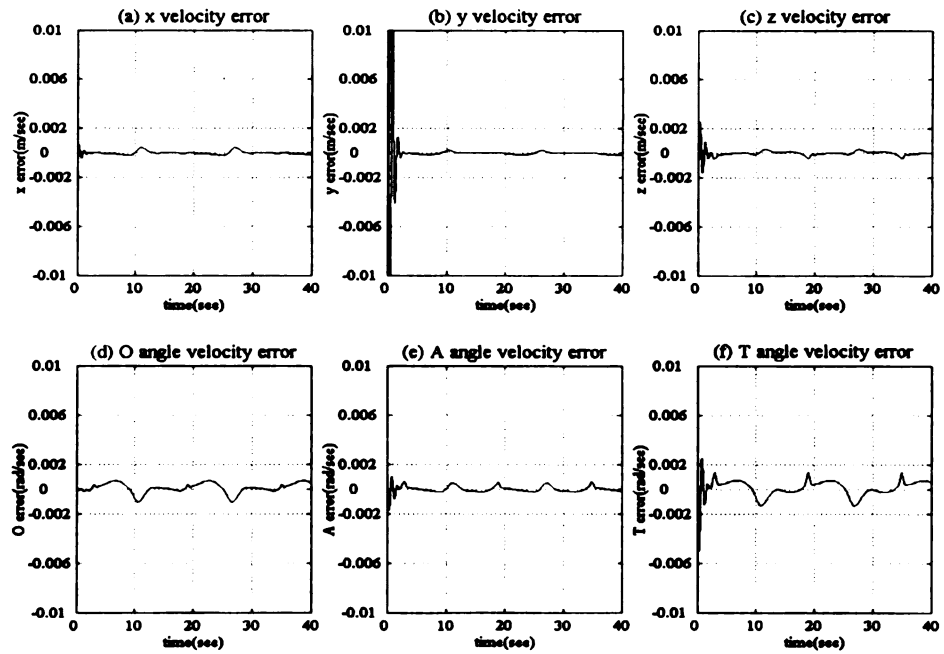


Figure 5.4: The error in velocity for task level position control

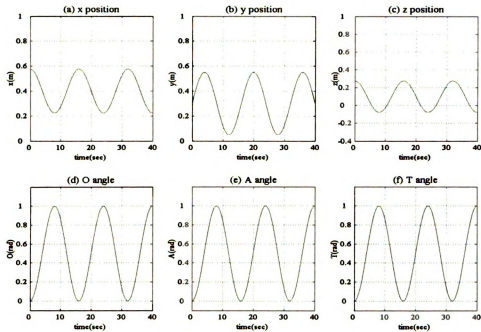


Figure 5.5: The actual and desired positions for the tool space controller expressed in the task space reference frame.

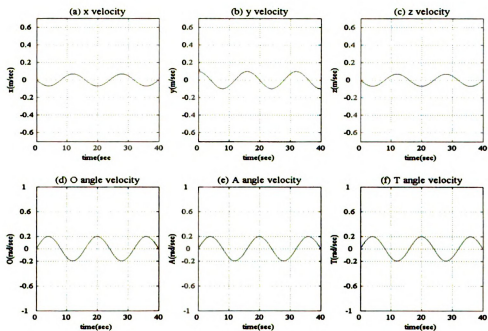


Figure 5.6: The actual and desired velocities for the tool space controller expressed in the task space reference frame.

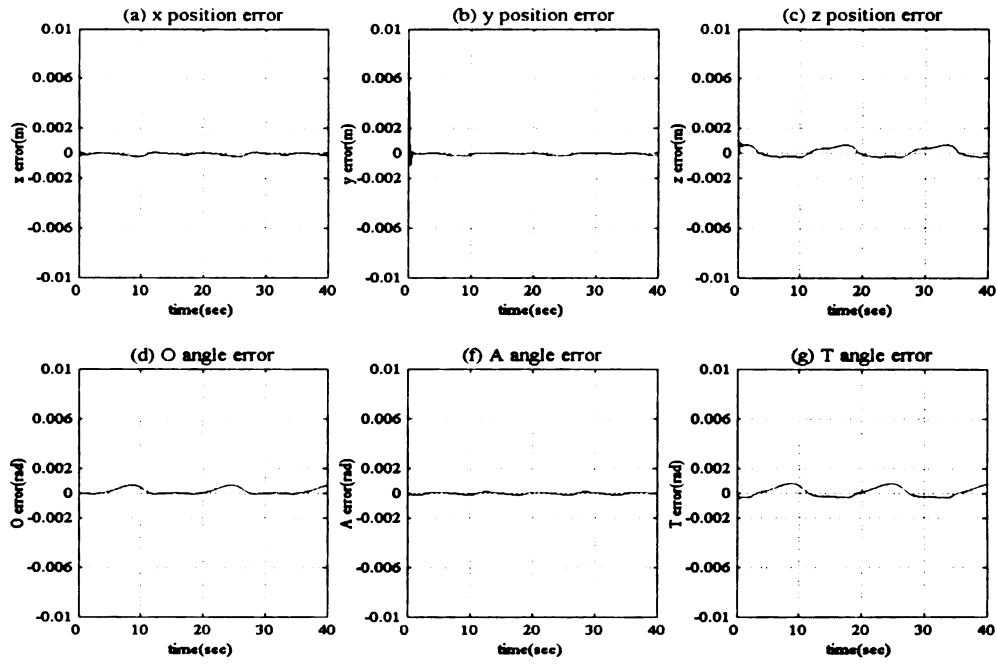


Figure 5.7: The errors in position for the tool space position controller expressed in the task coordinate frame

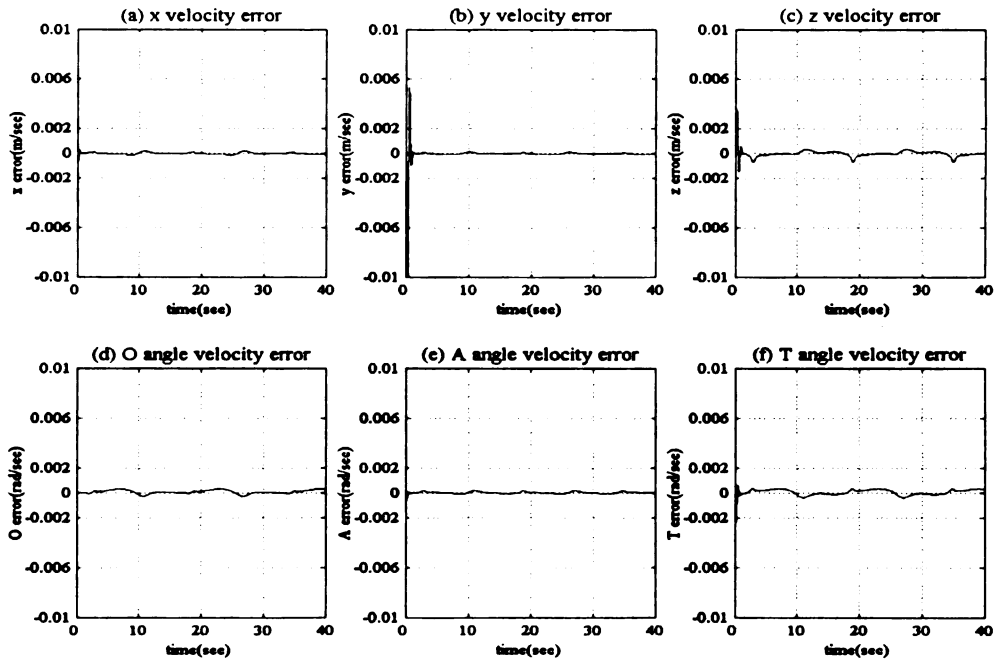


Figure 5.8: The errors in velocity for the tool space position controller expressed in the task coordinate frame.

5.3 Hybrid Force/ Position Control Experiments.

A hybrid task space force/position controller and a hybrid tool space force/position controller were simulated for the task of surface tracking. The task involved moving along the surface of a horizontal table while applying a constant force of 10 N perpendicular to the surface of the table. At the end of 12 seconds a change is affected in the magnitude of the desired force and it is ramped to 13 N within 1 second. The robot is initialized to an initial position such that $x = 0.4$ m, $y = 0.5$ m and $z = 0.2$ m. The initial orientation of the robot is such that it is aligned with the normal of the table and pointing towards the table. In this position the task space *OAT* angles are initialized to $O = 0$ rad, $A = 0$ rad and $T = 0$ rad. Thus the robot is positioned exactly at the surface of the table and it is pointing directly at the table surface. The robot moves towards the table under pure position control and contact is established. As soon as impact is detected the force control loop takes over and the interaction force with the table is regulated to the pre-specified desired value of 10 N. At the end of 10 seconds the robot is made to move along the surface of the table. In order to achieve the motion the desired position of the robot in the x and y direction is changed to 0.5 m from 0.4 m and 0.35 m from 0.5 m respectively. The same experimental setup is implemented for both the task level and the tool level hybrid controllers and the outputs of the two controllers are compared.

Figures 5.9 – 5.14 are the plots for the hybrid task space force/position controller performing the aforementioned task of surface tracking. Figure 5.9 (a) – (f) show the actual and desired position/orientation of the manipulator in the task space. Figures 5.10

(a) – (f) depict the actual and desired velocities in the task space. The Figure 5.11 (a) – (f) show the Cartesian position errors in the task space and Figure 5.12 (a) – (f) depict the Cartesian velocity errors in the task space for the task space hybrid force/position controller. Note that the position and velocity error in the z direction are not applied to the controller. This is because explicit force control is used in the z direction. Figure 5.13 (a) – (c) depict the actual and desired force profiles for the Cartesian task space x , y and z directions for the task space hybrid controller. The errors in the Cartesian space force in the x , y and z directions for the task space hybrid controller are shown in Figure 5.14 (a) – (c). Initially the manipulator is under position control. When the interaction force between the manipulator and the environment exceeds a threshold value, the z direction of the task space is switched to force control. This occurs $t \approx 2.5$ s. The force in the z direction is then regulated to a steady state value of 10 N. When $t = 10$ s the manipulator is commanded to move along the surface of the table while maintaining contact with a constant force of 10 N with the table surface. At the end of 12 s the desired force is ramped to 13 N within the interval of 1 s. Note that the manipulator is still moving along the table surface at that time. The position and velocity error in the z direction is quite large after impact, but the z direction is under force control thus the position and velocity error in the z direction is filtered out by the selection matrix. The force error in the z direction is also very small for steady state values of the desired force. The dynamic response of the force control loop is not very good, mainly due to the high gain nature of the feedback system, and it needs proper tuning.

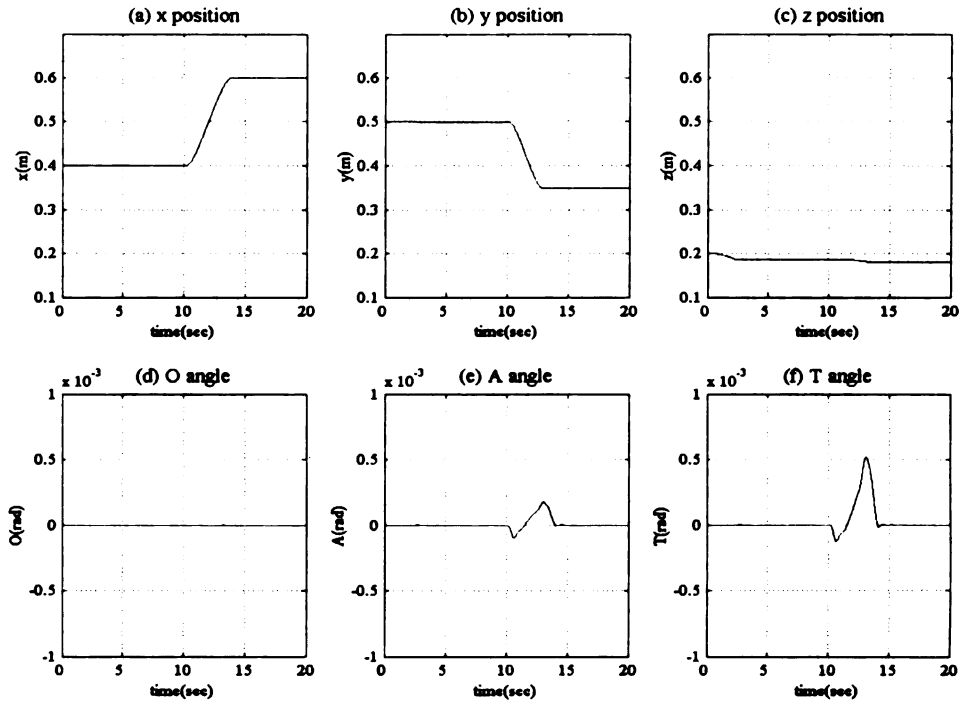


Figure 5.9: Actual and desired positions for task space hybrid force/position control.

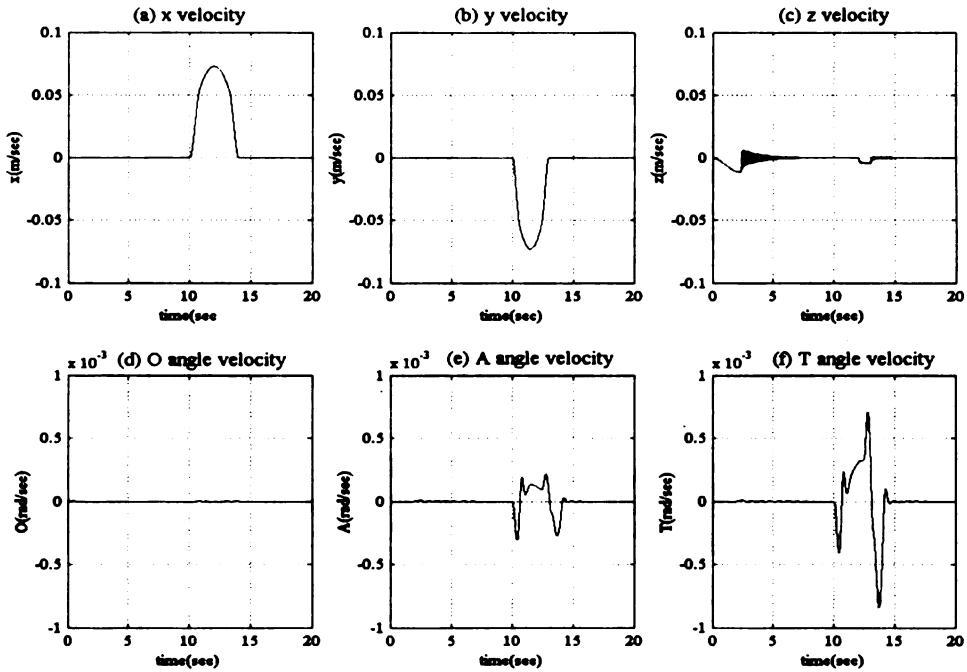


Figure 5.10: Actual and desired velocities for task space hybrid force/position control.

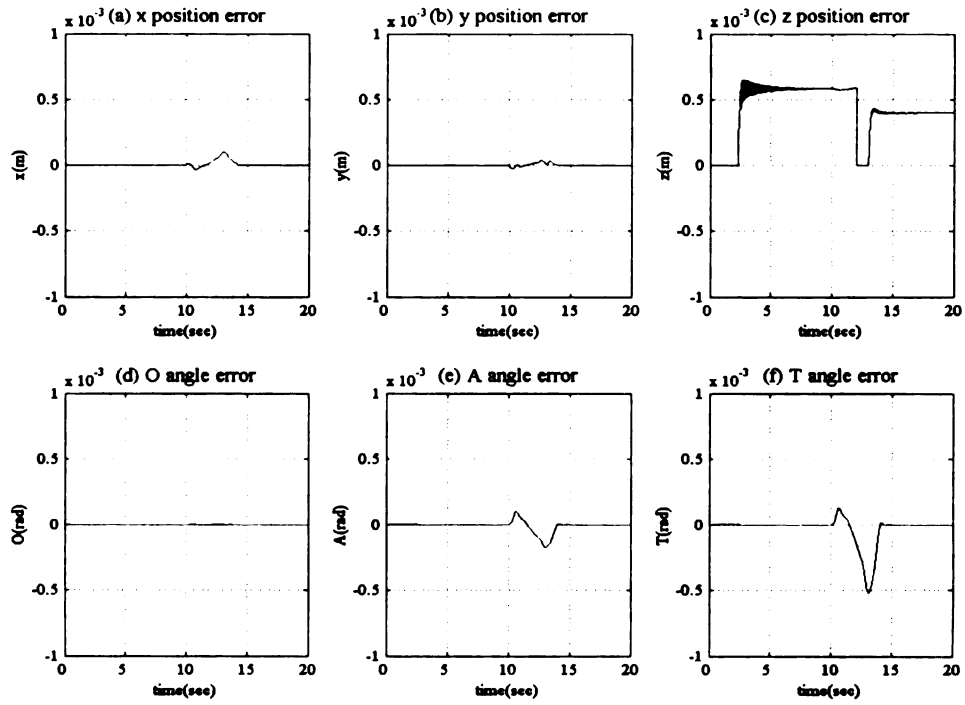


Figure 5.11: Error in position for task space hybrid force/position control.

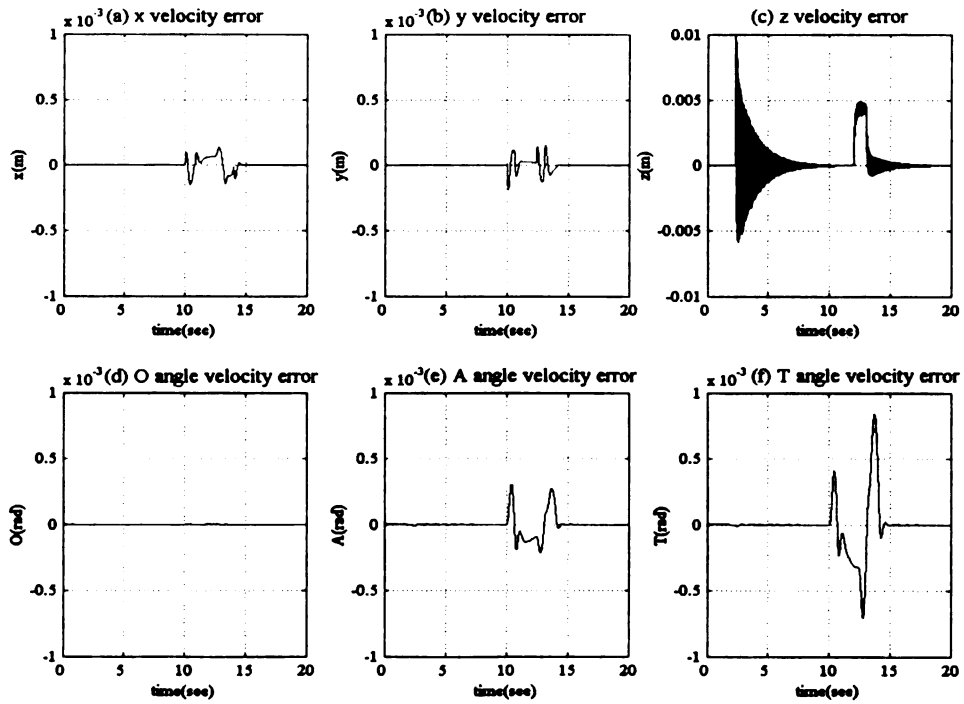


Figure 5.12: Error in velocity for task space hybrid force/position control.

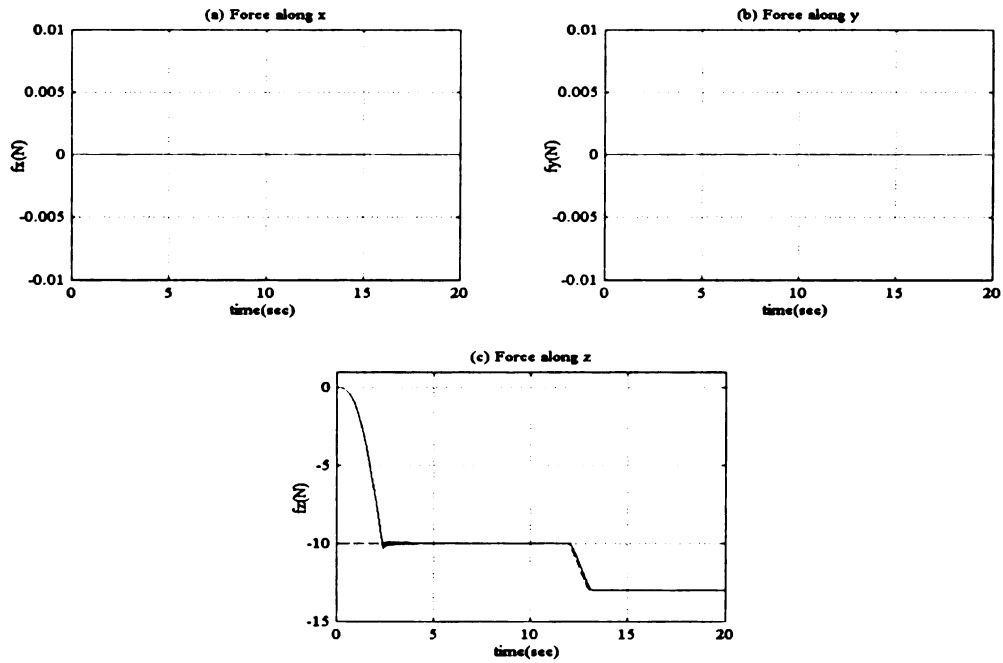


Figure 5.13: The actual and desired force profiles for the task space hybrid force/position controller.

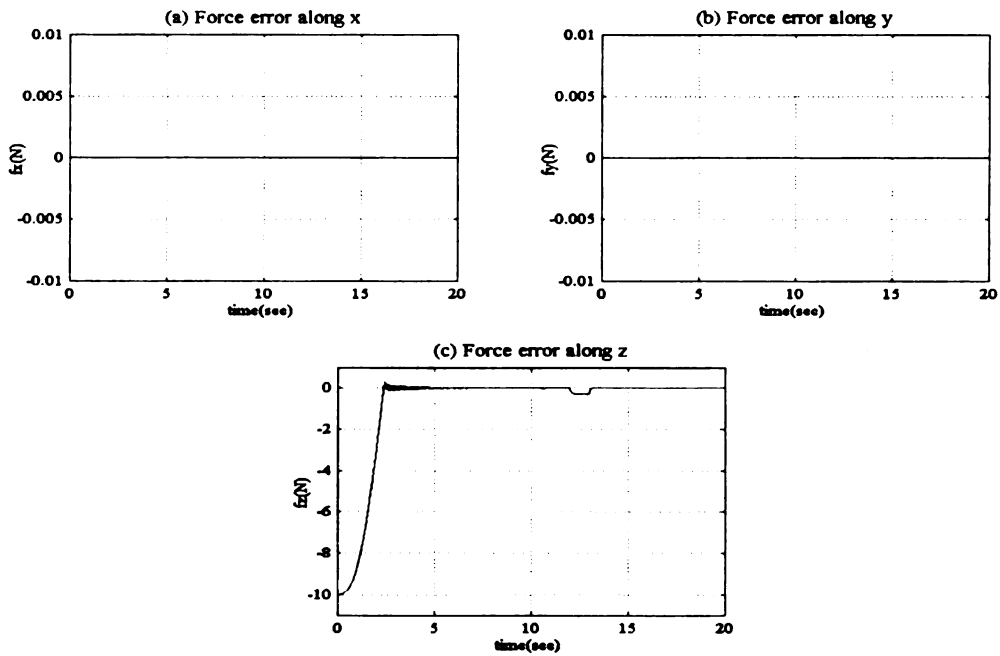


Figure 5.14: The error in the force profiles for the task space hybrid force/position controller.

The same experiment was carried out for the tool space controller. But in this case the robot was initialized such that the end-effector was at an angle with the table surface. The initial location of the end-effector in the task space was set to $x = 0.4$ m, $y = 0.5$ m and $z = 0.2$ m. The initial task space orientations were set to $O = -1.6$ rad, $A = 0.1$ rad and $T = -0.1$ rad. The robot was then commanded to move along the z direction under position control till impact was detected. After detection of impact, the z -axis of the tool frame was switched to force control. Similar to the case of task space control, at the end of 10 s the robot was commanded to move to the task space location $x = 0.6$ m and $y = 0.35$ m. At the end of 12 s the desired interaction force was ramped to 13 N within 1 second. The experiment was run for 20 s.

Figure 5.15 (a) – (f) show the actual and desired Cartesian task space positions and orientations of the manipulator under tool space hybrid force/position control. Figure 5.16 (a) – (f) show the actual and desired velocities in the task space. Figure 5.17 (a) – (f) depict the error in the task space position and Figure 5.18 (a) – (f) shows the task space velocity error. In this simulation, since the constraint surface is aligned along the task space axes, the errors in the task space position and velocity for the tool space hybrid force/position controller tend to zero. Also these plots are given in order to compare and contrast the performance of the task space and tool space hybrid force/position controllers. Further the errors in position and velocity expressed in the tool space, which are actually used to drive the tool space controller, are shown in Figure 5.19 and Figure 5.20. In the Figures 5.19 and 5.20, note that the position and velocity errors in the x , y , O , A and T directions asymptotically tend to zero.

The interaction forces expressed in the Cartesian task space are shown in Figure 5.21 (a) – (c). Note that interaction force is directed along only the task space z direction, which is perpendicular to the table surface. Figure 5.22 (a) – (c) show the interaction forces measured in the tool space frame of reference. The errors in interaction forces expressed in the tool space reference frame are shown in the Figure 5.23 (a) – (c). The force error in the z direction of the tool space reference frame is used to drive the force controller. Note that the measured interaction force is along the negative z -axis in the task reference frame while it is along the positive z -axis when expressed in the tool reference frame. This is due to the orientation of the tool reference frame. For this simulation the z -axis of the tool reference frame is directed along the negative z direction of the task frame.

For this simulation the end-effector was initialized so that it came into contact with the table surface at an angle with the table surface and not perpendicular to the table surface. So initially the end-effector is not aligned with the direction of the interaction force. Thus initially, some force transients are measured along the tool space x and y axes. But the orientation control on the manipulator is structured in such a fashion that the z axis of the end-effector aligns with the direction of the sensed interaction force. Hence the force measured along the tool space x and y axes tends to zero as the end-effector z axis aligns with the direction of the interaction force.

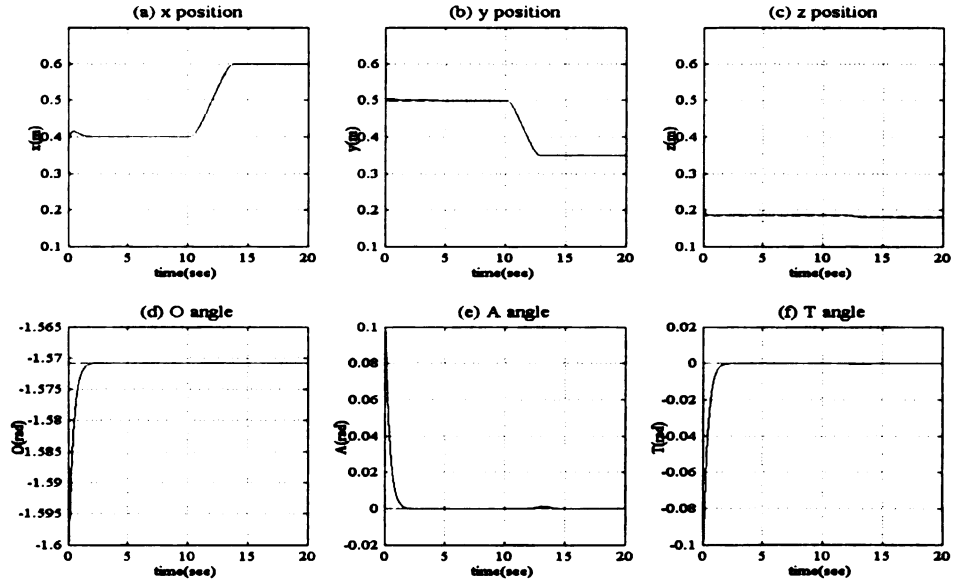


Figure 5.15: The actual and desired positions, expressed in the task space, for the tool space hybrid force/position controller.

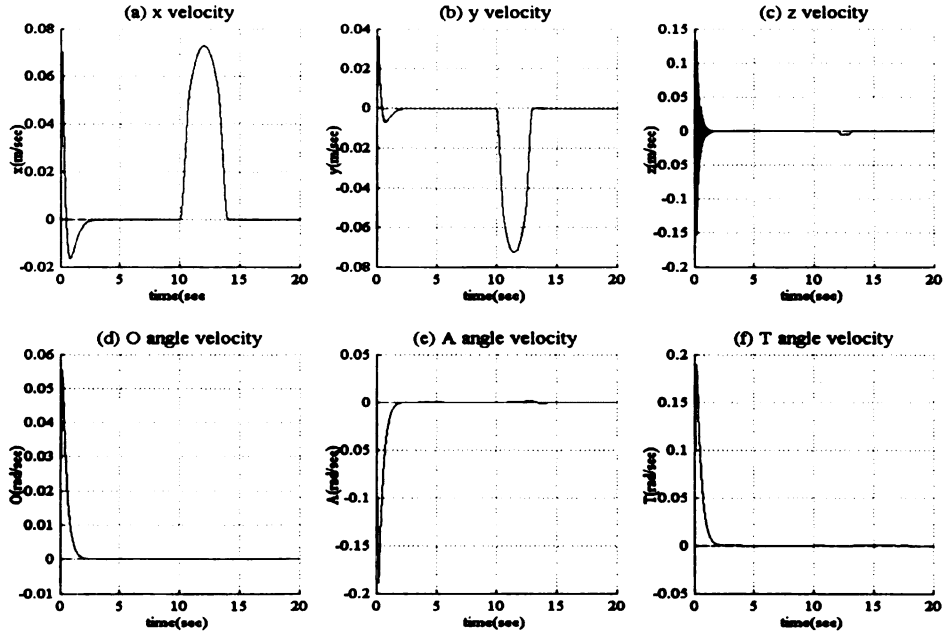


Figure 5.16: The actual and desired velocities, expressed in the task space, for the tool space hybrid force/position controller.

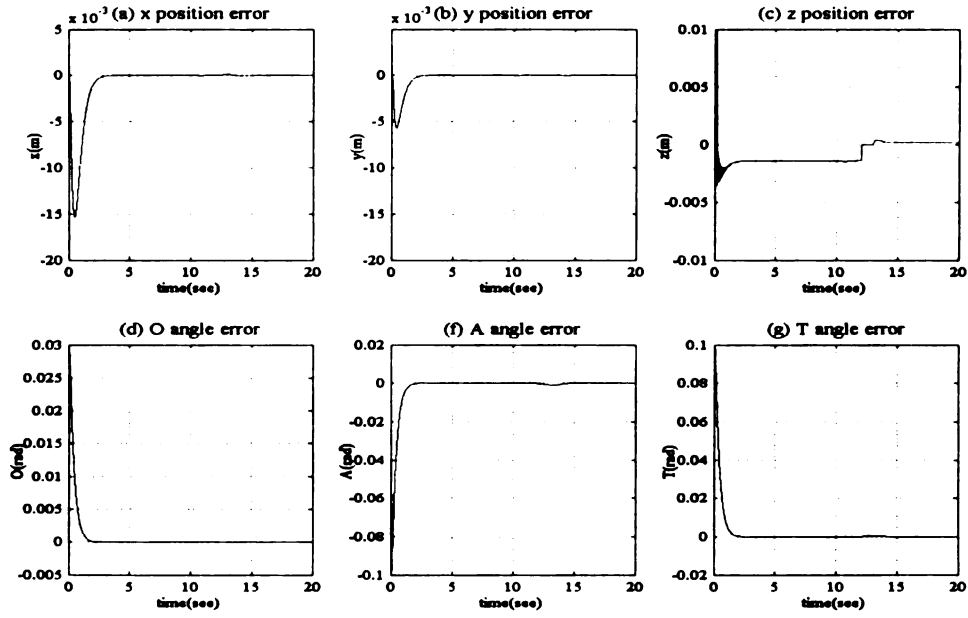


Figure 5.17: The position errors expressed in the task space for the tool space hybrid force/position controller.

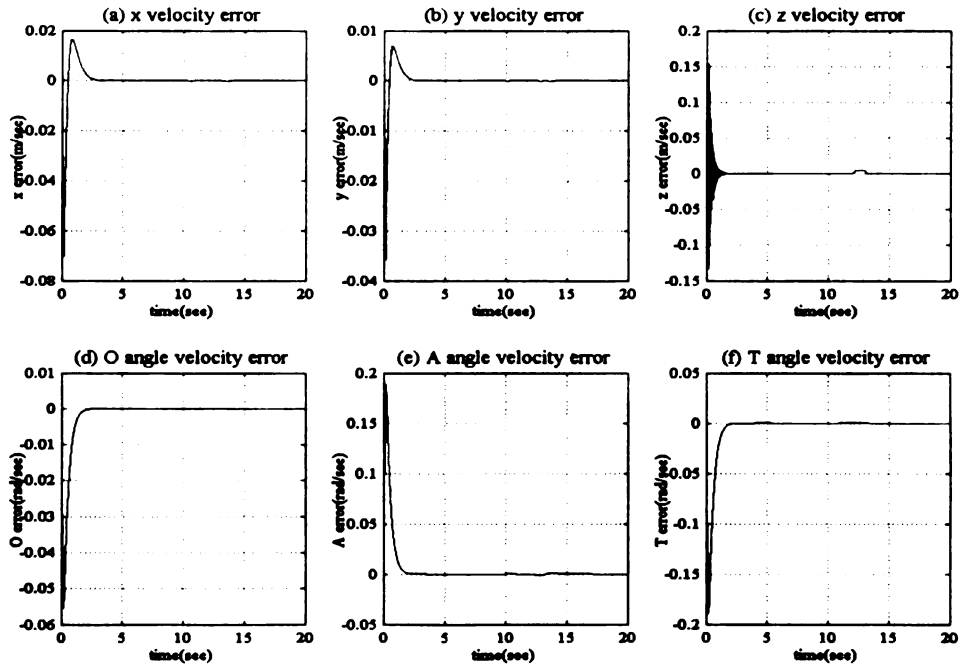


Figure 5.18: The errors in velocity, expressed in the task space, for the tool space hybrid force/position controller.

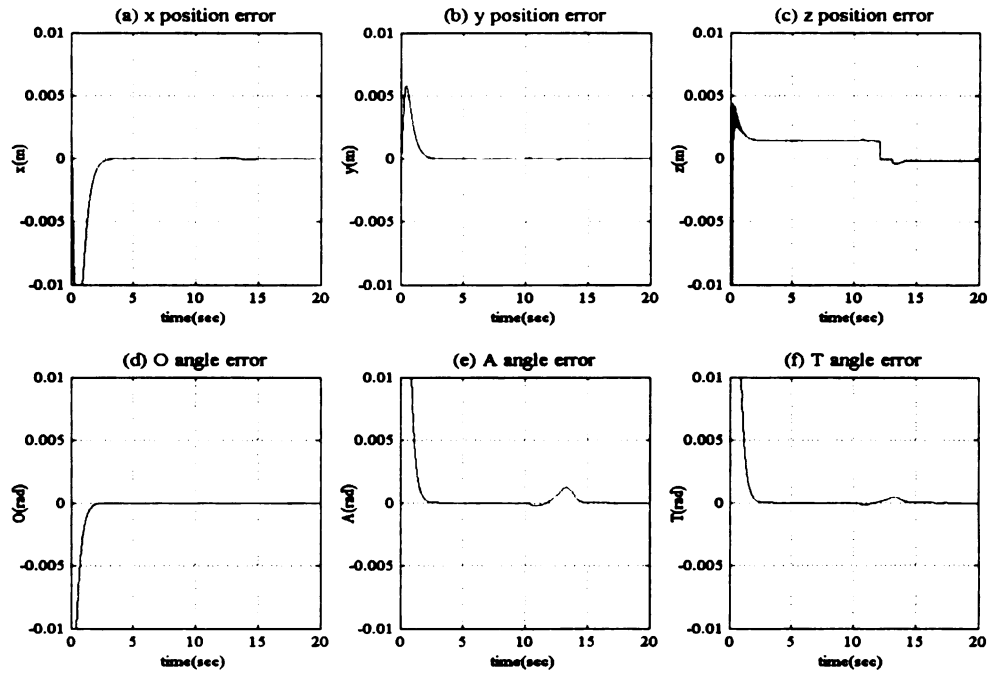


Figure 5.19: Position errors expressed in the Cartesian tool space for the tool space hybrid force/position controller.

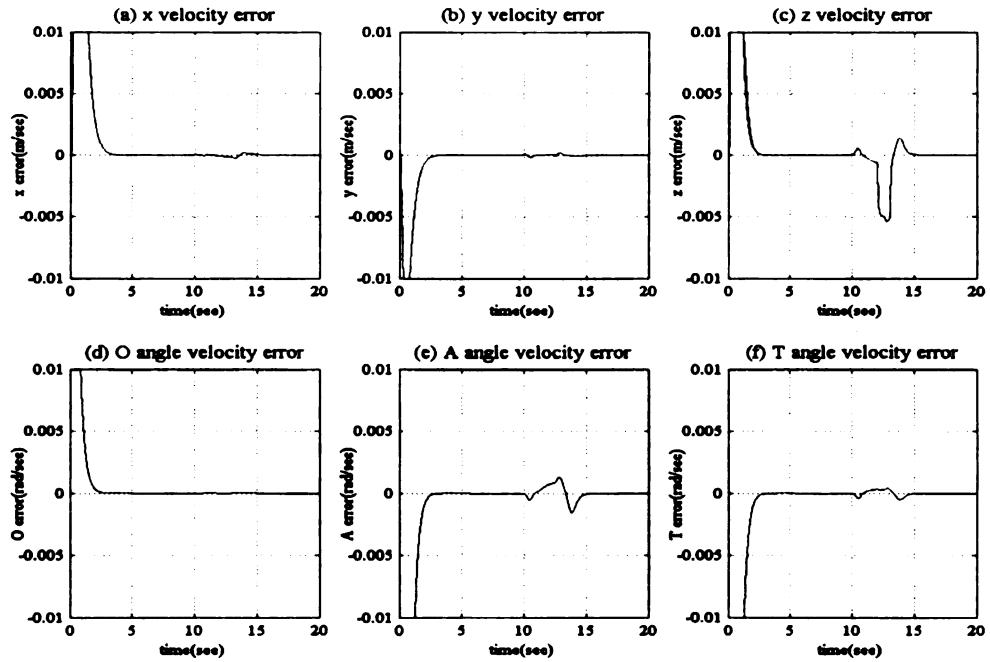


Figure 5.20: Velocity errors expressed in the Cartesian tool space for the tool space hybrid force/position controller.

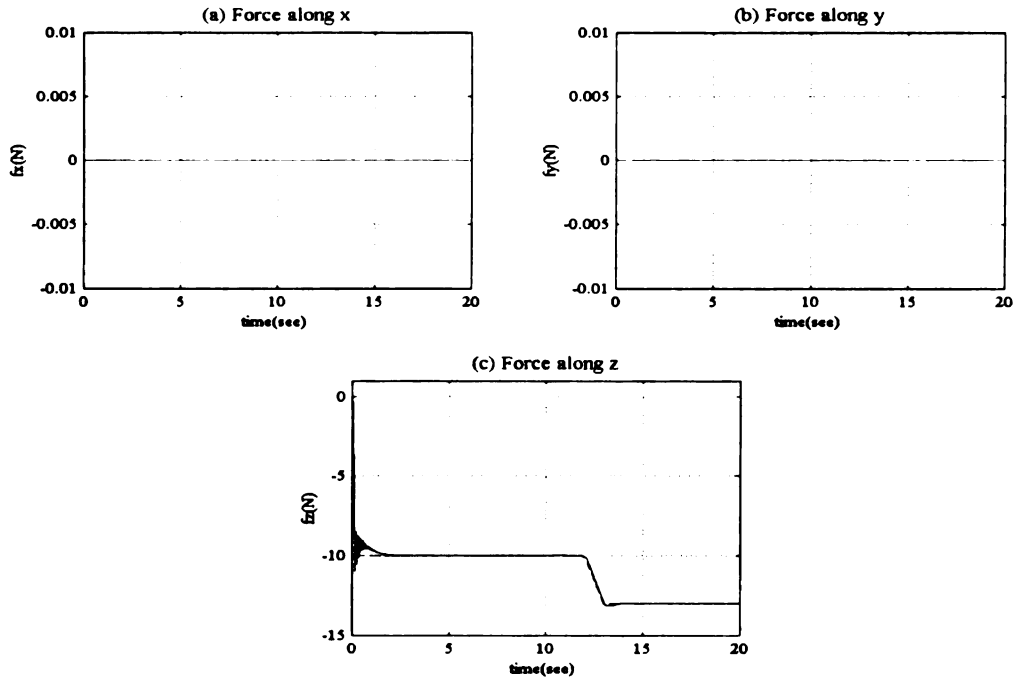


Figure 5.21: The actual and desired forces expressed in the task space for the tool space hybrid force/position controller.

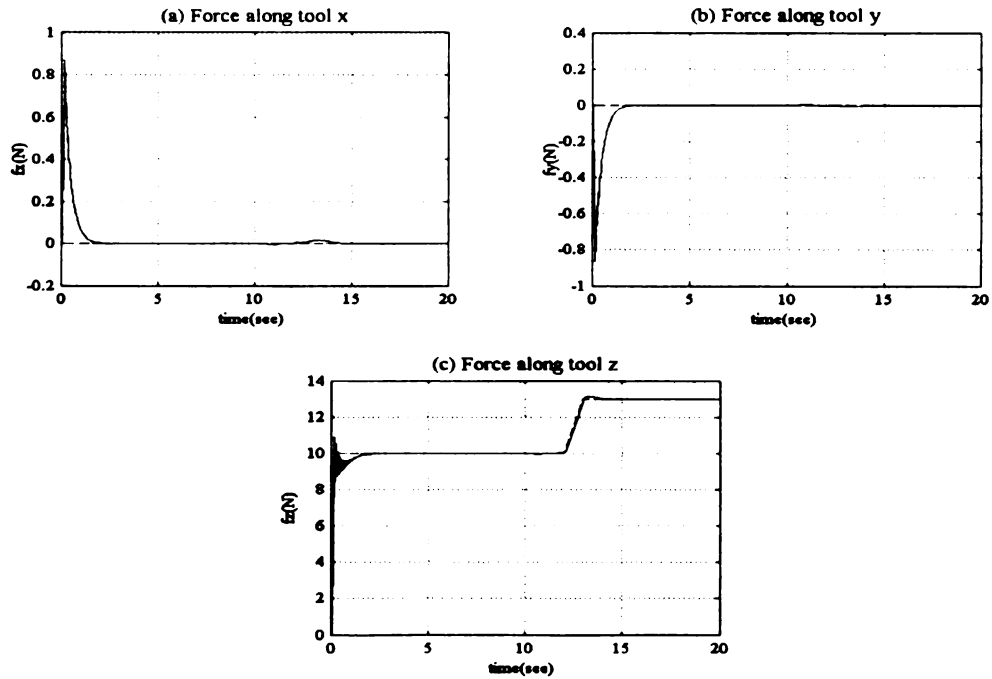


Figure 5.22: The actual and desired forces expressed in the tool space frame of reference for the tool space hybrid force/position controller.

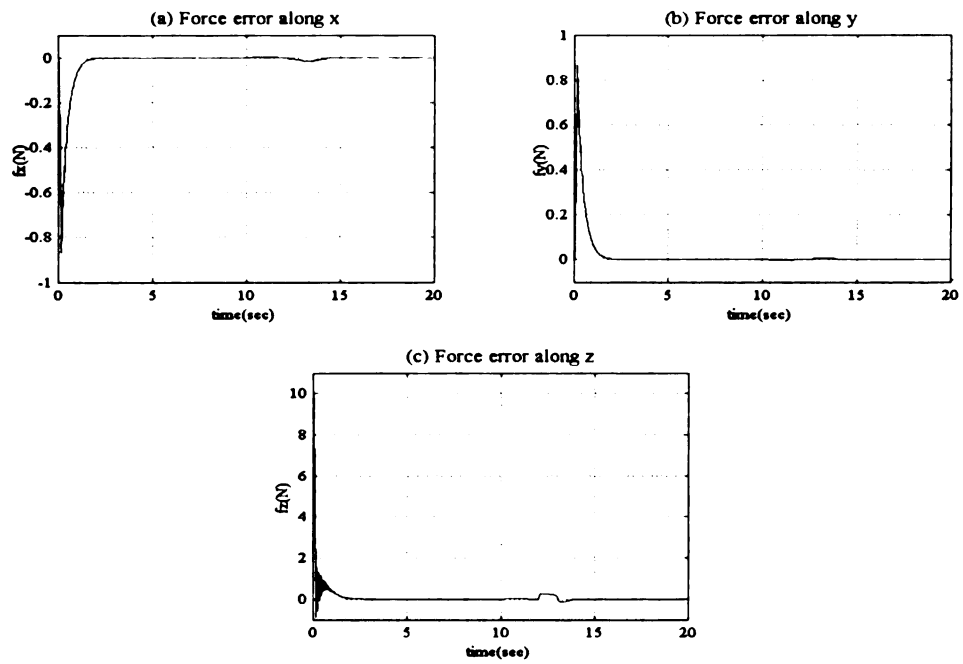


Figure 5.23: The force error expressed in the tool space frame of reference for the tool space hybrid force/position controller.

5.4 Tracking a Curve in Two Dimensions

When an object is completely unknown, no kind of task can be specified on it: some rough a priori knowledge is necessary. In this experiment a tool space hybrid force/position controller was simulated for the task of tracking a flat object whose contour is not known, but the plane in which the contour lies is known. This is similar to the problem defined in [30]. The surface to be tracked is simulated as a parabolic cylinder with its axis along the y axis of the task space. In the x - z plane the profile of the cylinder is a parabola. The equation of the parabola is:

$$z = 6x^2 - 3x$$

Note that the shape of the surface is not input to the controller directly. The controller uses the direction of the sensed interaction force to gauge the direction of the surface normal. Figure 5.24 shows the parabolic surface to be tracked.

The robot is initialized to a task space position such that $x = 0.0$ m, $y = 0.2$ m and $z = 0.0$ m. The orientation of the end-effector is set at $O = -\frac{\pi}{2}$, $A = 0$ and $T = 0$. Thus, in the initial position the robot is pointing along the negative task space z -axis. The robot is commanded to move along the negative task space z -axis. By moving in this direction the end-effector comes into contact with the surface and interaction forces are generated. The controller uses the direction of the interaction force to gauge the surface normal and orients the robot along the surface normal. The force control loop regulates the interaction force along the tool space z -axis to the desired set point value, *i.e.*, 10 N.

At the end of 10 seconds the robot is commanded to move, following a straight-line path, to the task space position $x = 0.25$ m, $y = 0.2$ m and $z = -0.375$ m. The desired force command along the z -axis of the tool space is maintained at 10 N. The robot follows the parabolic contour to the set reference point. When the robot reaches the desired set point, it is again commanded to move along a straight-line path to the task space position $x = 0.5$ m, $y = 0.2$ m and $z = 0$ m. The desired force command along the z -axis of the tool space is maintained at 10 N.

Figure 5.25 (a) – (f) show the actual positions and orientations of the manipulator expressed in the task space frame of reference. Note that the position in the x direction is continuously increasing with time while the position in the z direction first decreases with time, when the robot is following the downgrade of the parabola, and then increases with time when the manipulator is following the upslope of the parabolic surface. The position in the y direction remains constant at 0 m because the robot is not commanded to move along the y direction. Also note that the orientation angles O and T maintain a near constant value at $-\frac{\pi}{2}$ and 0 respectively. The A angle reflects the changes of the attitude of the end-effector with time. Also note the large initialization error in the A angle due to the initialization error in the orientation of the end-effector.

Figure 5.26 (a) – (c) show the interaction forces expressed in the task space frame of reference. Note that, from figure 5.26 (b), the force along the y direction is constantly at 0. When the manipulator is initially pointing along the negative task space x -axis the interaction force is regulated to approximately 10 N in that direction while it has a very small value along the task space z -axis. As the manipulator starts moving along the

surface, the orientation of the end-effector changes and so does the magnitude of the task space x and z components of the interaction force. When the manipulator is at the nadir of the surface, at approximately $t = 40$ s, the surface normal is directed along the positive task space z -axis. Note that at this time the force component along the task space x direction is zero while the force component along the task space z direction is regulated to 10 N. Figure 5.27 (a) – (c) shows the actual and desired interaction force expressed in the tool space frame of reference. From the figure note that the z component of the interaction force is regulated at 10 N whereas the x and y components are regulated to approximately zero values.

Figure 5.28 shows the profile, in the task space x - z plane, of the location of the end effector of the manipulator. The dashed line represents the location of the actual surface. We see that the profile traced by the end-effector is exactly parallel to the surface profile. In order to generate an interaction force the surface needs to be deformed. This deformation gives rise to the interaction forces upon contact of the end-effector with the environment [52]. Note that the two curves are nearly exactly parallel. This implies that the end-effector maintains a constant angle with the surface normal, which in this case is regulated to 0 rad.

In order to see if the end-effector is perpendicular to the surface we need to compare the surface normal vector with the z -axis of the tool space. Figure 5.29 plots the x , y and z components of the end-effector z -axis (solid line) and the surface normal vector (dashed line). The components of the end-effector z -axis can be calculated from the orientation angles OAT using the following formulae:

$$\begin{aligned}
 a_x &= S_O S_A C_T - C_O S_T \\
 a_y &= -C_O S_A C_T - S_O S_T \\
 a_z &= -C_A C_T
 \end{aligned}$$

From the figure it can be seen that the orientation of the end-effector coincides nearly exactly with the surface normal. Hence, it can be concluded that the end-effector remains perpendicular to the surface.

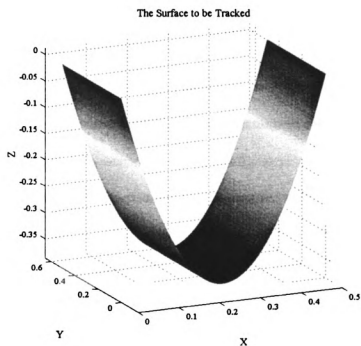


Figure 5.24: The cylindrical parabolic surface to be tracked.

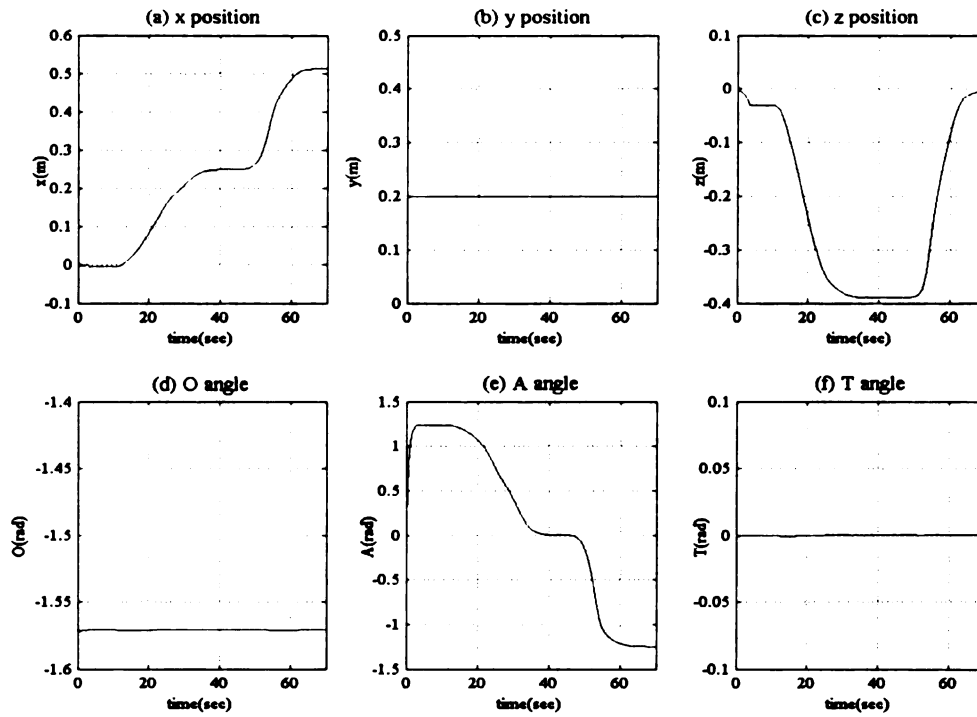


Figure 5.25: The actual positions and orientations of the manipulator.

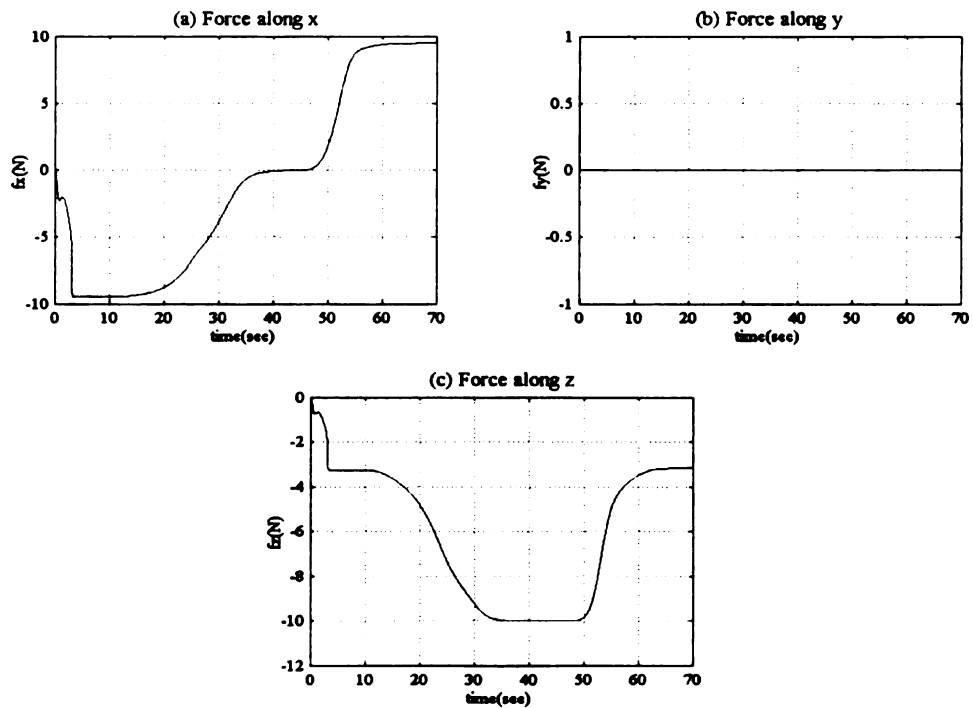


Figure 5.26: The interaction force expressed in the task space frame of reference.

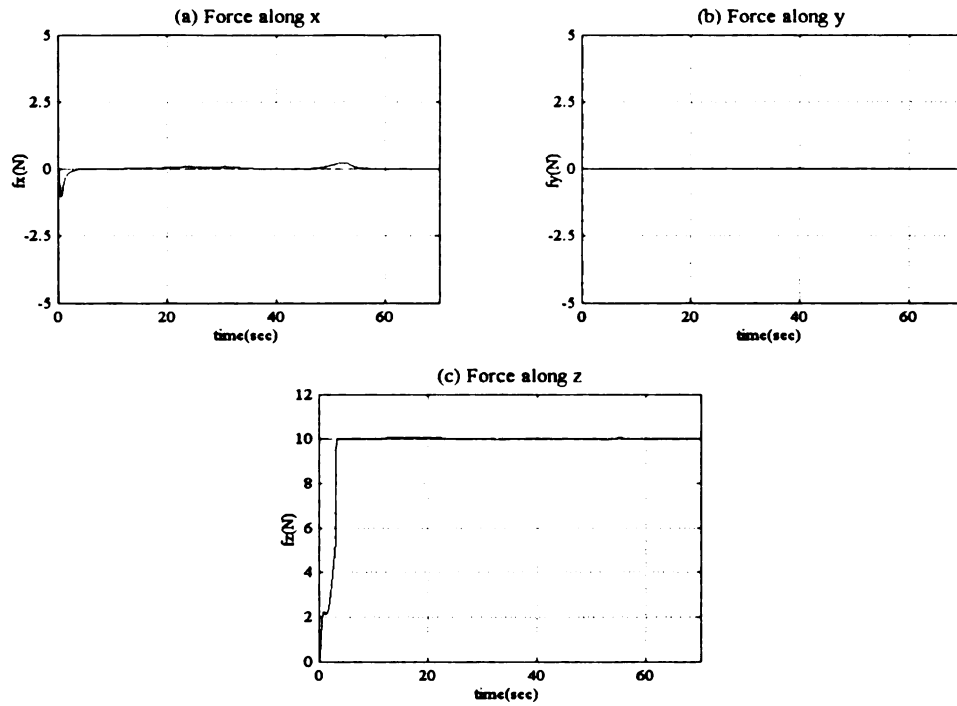


Figure 5.27: The interaction force expressed in the tool space frame of reference.

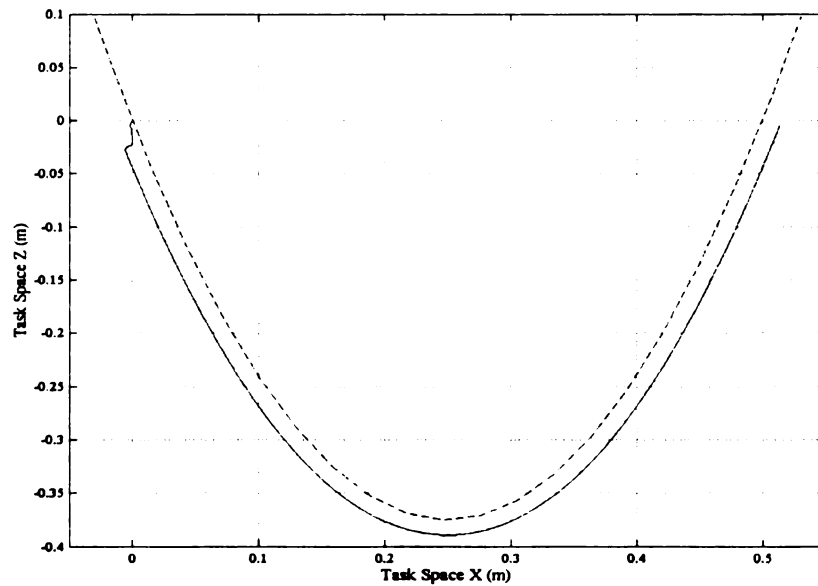


Figure 5.28: The surface profile and the actual profile of the end-effector location in the x - z plane in the task space.

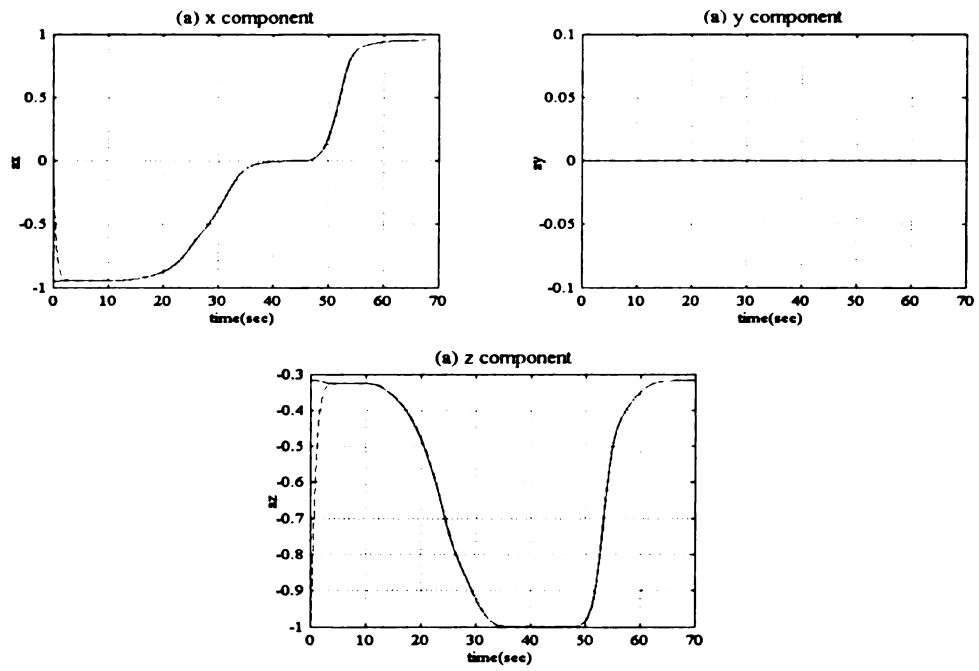


Figure 5.29: The task space x , y and z components of the surface normal and the z -axis of the end-effector frame

The same simulation was carried out with the desired force being time varying, rather than a constant value as was done in the previous experiment. The value of the desired force was taken to be a time varying sinusoid whose equation is given by:

$${}^{tool}f_z = 10 + \sin(0.2t)$$

Figure 5.30 (a) –(f) show the actual location of the end-effector for this experiment. Figure 5.31 (a) – (c) shows the actual interaction force expressed in the task space frame of reference. This figure is similar to Figure 5.26 except for the superimposed sinusoids on the x and the z component of the force profile. Figure 5.32 (a) – (c) shows the interaction force expressed in the tool space reference frame. Again this figure is similar to the Figure 5.27 except for the superimposed sinusoid along the z direction of the end-effector frame.

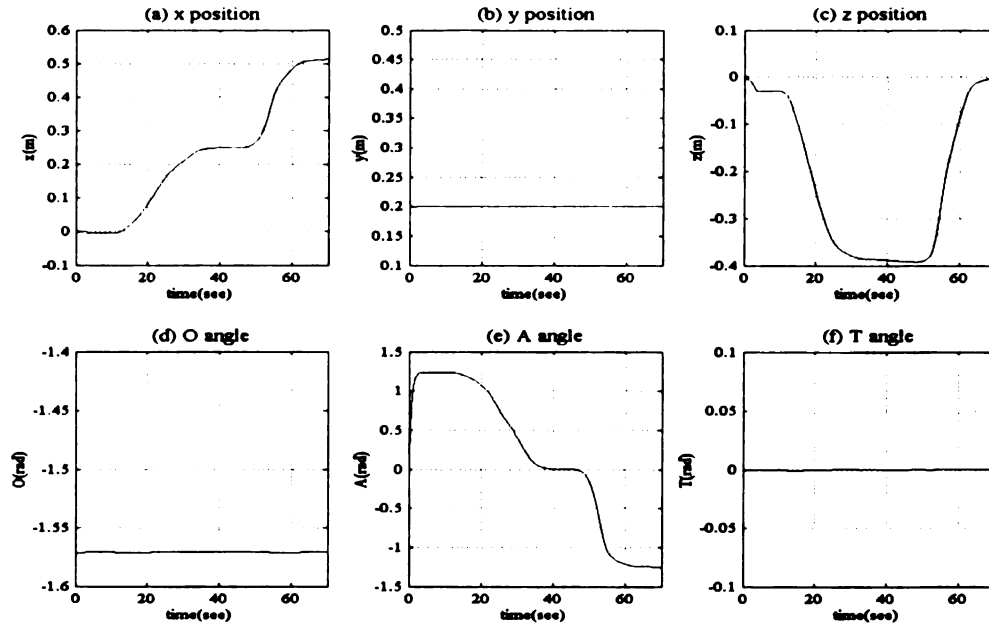


Figure 5.30: The actual position of the end-effector expressed in the task space frame of reference.

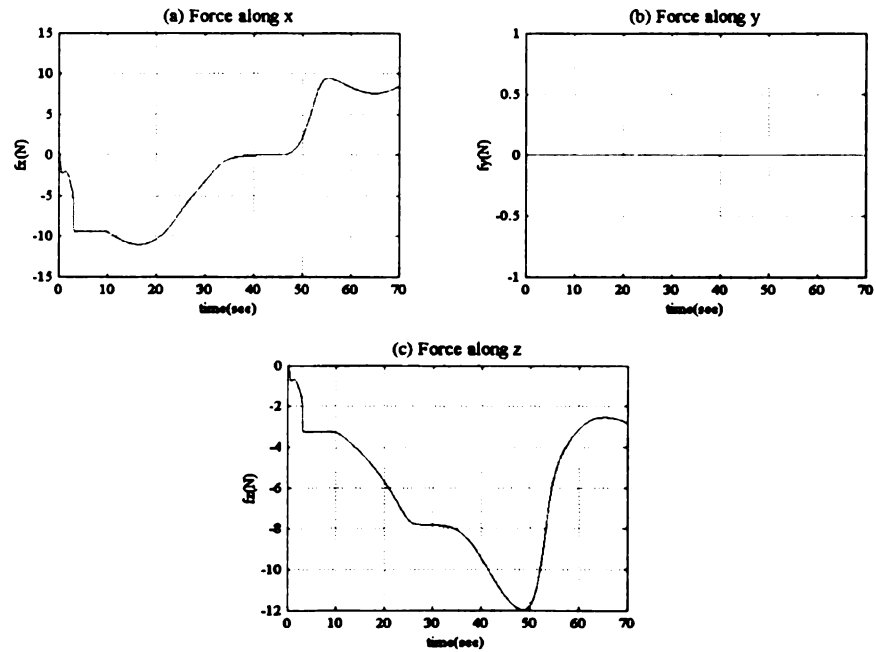


Figure 31: The interaction force expressed in the task space frame of reference.

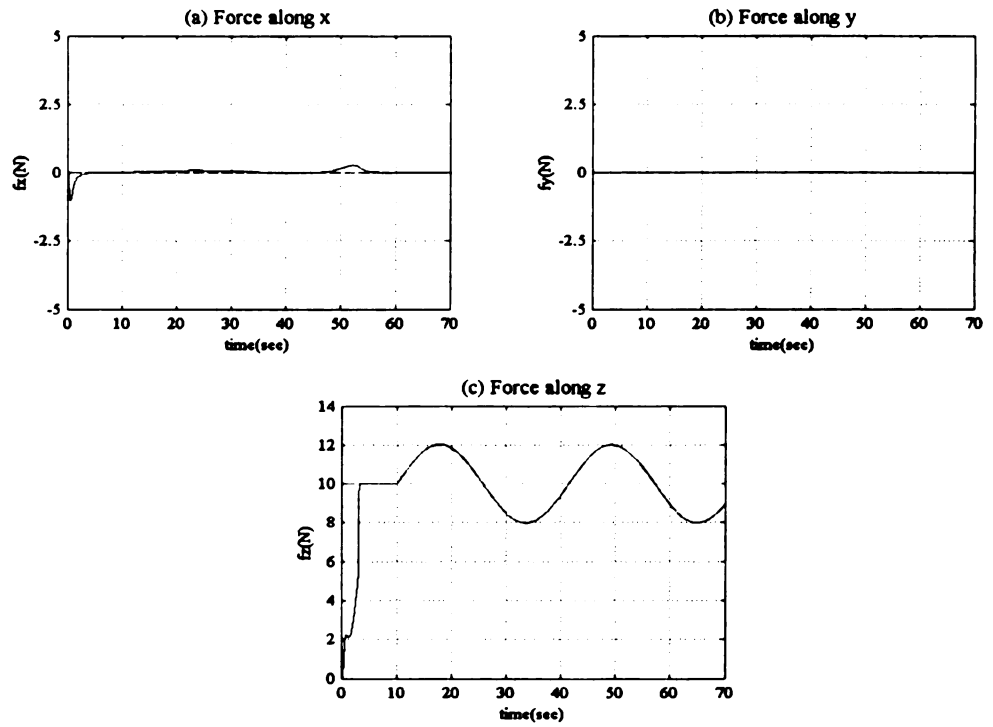


Figure 5.32: The interaction force expressed in the tool space frame of reference.

5.5 Tracking a Curve in Three-Dimensions

Extending the concept of two dimensional curve tracking further, the tool space hybrid force position controller was simulated to track a three dimensional contour. The contour was taken as a sector of a hollow sphere having the equation as:

$$x^2 + y^2 + z^2 = 0.5^2 .$$

The radius of the sphere is 0.5 m and it is centered at the origin of the task space. The sphere is hollow and the robot is assumed to be inside the sphere and the task is to track the inner surface of the sphere from within. For this simulation, the robot comes into contact with the sphere at an angle with the local surface normal. The initial position of the robot in the task space is $x = 0$ m, $y = 0.4$, $z = -0.3$, $O = -\frac{\pi}{2}$ rad, $A = -0.321751$ rad, and $T = 0.901832$ rad. At the end of 10 seconds the manipulator is commanded to move, following a straight-line path, to the task space position $x = 0.4$ m, $y = 0.25$ m and $z = -0.1658$ m. The desired force command along the z -axis of the tool space is maintained at 10 N. The robot follows the spherical contour to the set reference point.

Figure 5.33 shows a 3 dimensional plot of a part the spherical surface that the robot is supposed to track. Figure 5.34 (a) – (f) show the desired and actual position and orientation of the end-effector while it is moving along the surface of the sphere. The solid line represents the actual position and the dashed line represents the desired position. Note that the desired positions are generated as a straight-line from the initial point to the final reference point without any knowledge about the shape of the surface.

Figure 5.35 (a) – (c) show the plots of the interaction forces expressed in the task space frame of reference. Figure 5.36 (a) – (c) show the plots of the actual and desired x , y and z components of the interaction forces expressed in the tool space frame of reference. Note that the components of the interaction forces along the x and y axes are regulated to zero while the component of the interaction force along the z axis of the tool space is regulated to 10 N. In this case, all the points on the surface of the sphere satisfy the equation

$$x^2 + y^2 + z^2 = 0.25.$$

Now, for the end-effector to be in contact with the sphere surface, the end-effector x , y and z coordinates must satisfy the following equation

$$x^2 + y^2 + z^2 \geq 0.25.$$

In order to prove that the tip of the end-effector is in contact with the surface of the sphere, we need to use a metric that would indicate whether the tool tip is inside or

Hence we can use $m = x^2 + y^2 + z^2$ as a metric for this example.

Figure 5.37 plots the equation of the above metric. The dashed line is plotted at $m = 0.25 \text{ m}^2$. We see that the two lines are parallel, implying that the end-effector maintains a constant angle with the surface normal, which in this case is 0 rad.

In order to see if the end-effector is perpendicular to the surface we need to compare the surface normal vector with the z -axis of the tool space. Figure 5.38 (a) – (c) plot the x , y and z components of the end-effector z -axis (solid line) and the surface

normal vector (dashed line). Note that the two curves nearly coincide. This implies that the end-effector z-axis remains perpendicular to the surface of the sphere. Thus the interaction force is directed along the end-effector z-axis and can be completely regulated.

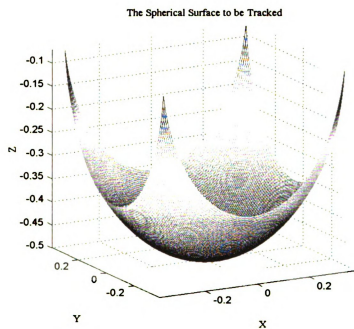


Figure 5.33: The spherical surface that is to be tracked.

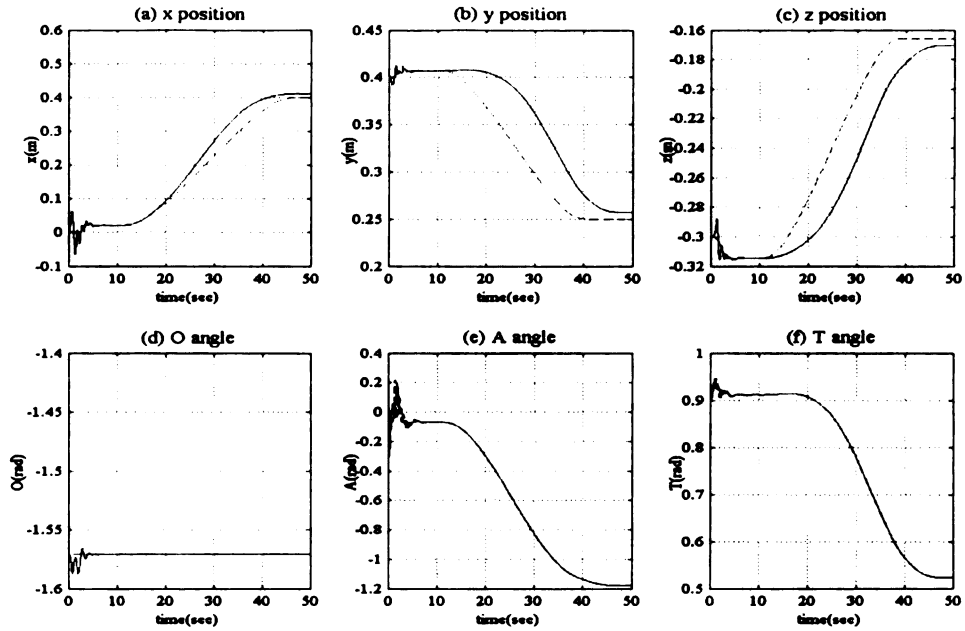


Figure 5.34: The actual and desired positions and orientations expressed in the task space frame of reference.

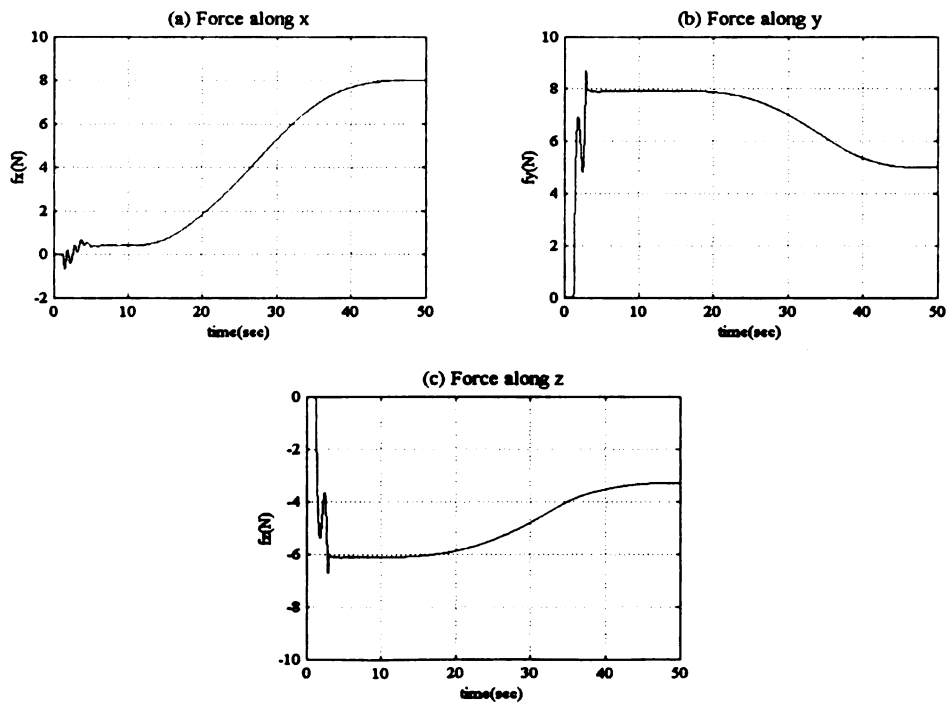


Figure 5.35: The actual interaction force expressed in the task space frame of reference.

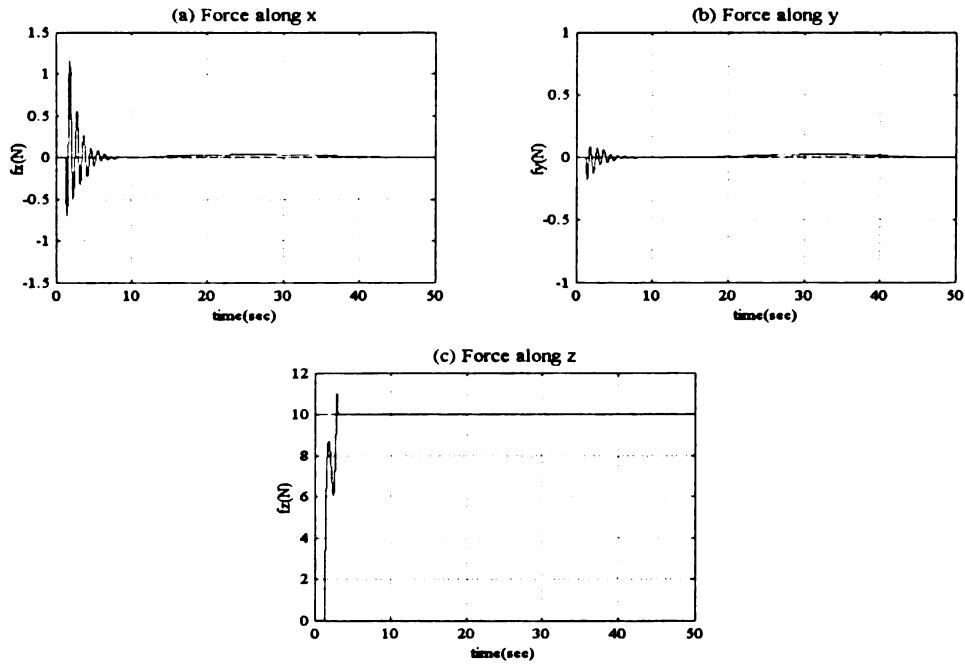


Figure 5.36: The actual and desired interaction forces expressed in the tool space frame of reference.

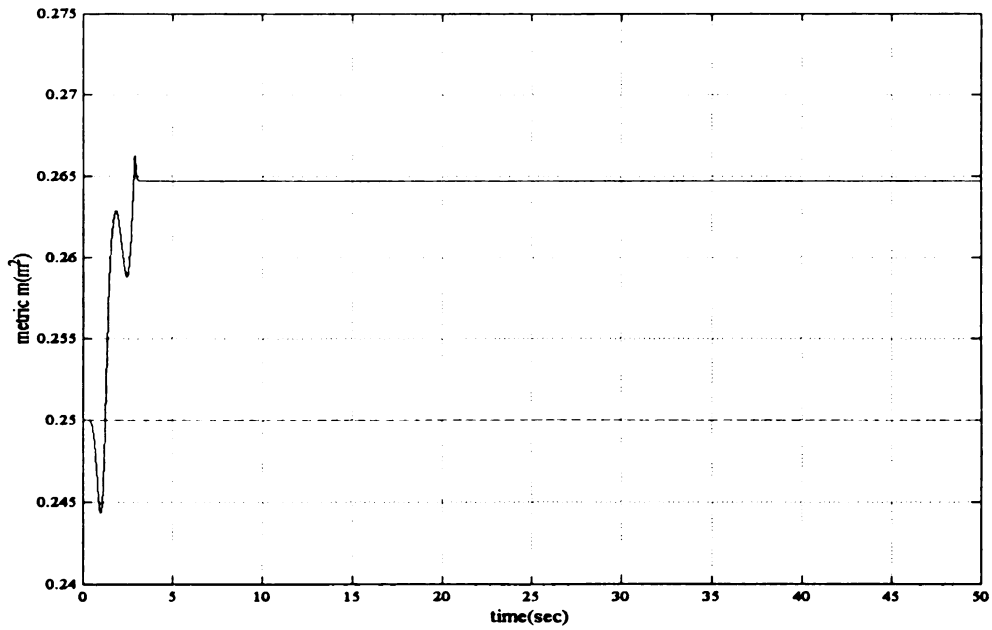


Figure 5.37: The metric $m = x^2 + y^2 + z^2$.

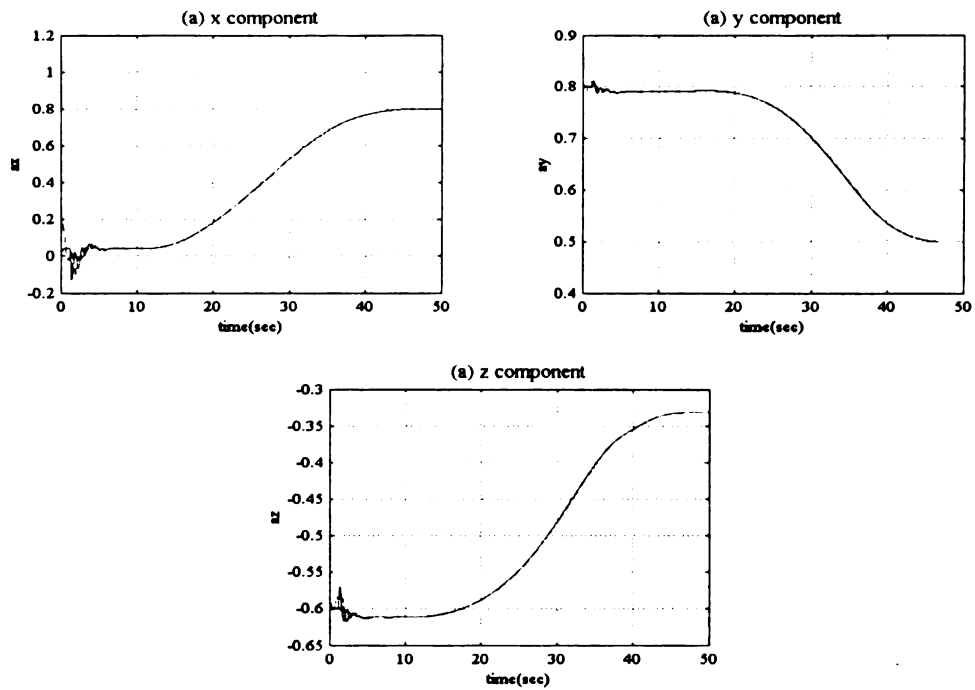


Figure 5.38: The task space x , y and z components of the surface normal and the z -axis of the end-effector frame

Chapter 6

6.1 Summary

In Chapter 3 the concept of control of robot manipulators in the robot tool space was introduced. From the simulation studies in Section 5.2 it was seen that it yielded similar results as a task space controller for tasks involving position control. In Chapter 4 hybrid force/position control was discussed. The inadequacy of the task space hybrid force/position controller for controlling the force in any arbitrary direction in the task space was pointed out. It was also noted that such a control strategy was not robust to interaction forces in arbitrary directions and the build up of the interaction forces could result in damage to the manipulator or the environment. To overcome this limitation of the task space hybrid control strategy, the concept of hybrid force/position control in the robot tool space was introduced in Section 4.3.

Stability analysis for robot tool space control was presented in Section 4.4. Stability analysis for the tool space hybrid force/position controller was also presented in Section 4.4. Simulation studies for the tool space hybrid force/position controller are presented in Chapter 5. In Section 5.3 the task space and tool space strategies for hybrid force/position control were compared and contrasted. The task to be performed was tracking a horizontal surface while simultaneously controlling the normal interaction force with the surface. Similar results were obtained for both the task space and tool space control strategies. Initializing the manipulator at an angle with the surface normal

for the tool space controller showed that the tool space control strategy was robust to errors in initialization of the manipulator.

Further in Section 5.3 the task of tracking a two-dimensional contour in a known plane was simulated. The contour to be tracked was a cylindrical parabola with its axis along the task space y -axis. The robot was initialized so that the end-effector was aligned with the task space z -axis. The robot was able to successfully track the contour without any prior knowledge about the contour, by just using the measured normal interaction force. In Section 5.4 the tool space hybrid force/position control approach was used to track an unknown three-dimensional contour.

6.2 Conclusions and Future Work

The strategy of tool space hybrid force/position control opens up the possibility for controlling the interaction forces that are constantly changing directions. Thus, using this strategy the manipulator can interact with poorly structured and unknown environments. Further, using this strategy, applications such as surface tracking, grinding, polishing, milling and deburring, which require the manipulator to come in contact with the workpiece can now be carried out on workpieces that are not pre-specified. This would in turn affect productivity by eliminating the down time spent in extensive path planning and reprogramming of the manipulator when dealing with different workpieces. Also, using this strategy, the force exactly along the normal to the surface can be regulated. This was not possible using the task space hybrid force/position controller that could regulate only the component of the force directed along the task space axes where force control had been planned. Further, unlike the task space hybrid

force/position control, this strategy is robust to impact along directions where motion control has been planned.

Though preliminary experimental results are provided in Chapter 5, many improvements and modifications need to be carried out on the proposed controller. First the restriction of the assumption of a frictionless contact needs to be tackled. Adaptive algorithms that can estimate the friction coefficients for surfaces need to be implemented. This would abet the controller to estimate the exact direction of the tangent even in the presence of friction. Also for improved tracking performance better force control algorithms ([73]) need to be implemented. Further, as mentioned in [71] and [72], for implementation in cutting and polishing operations, it is required to regulate the interaction force at a certain angle with the surface normal. The tool space hybrid controller could be modified to control the interaction forces at a specific angle to the contact surface. The current proposed strategy does not take into account the task of contact transition. It is assumed that the manipulator comes into contact with the environment at a extremely low velocities and thus the need for contact transition control is not realized. But for practical implementation of this controller, a scheme for contact transition needs to be adopted. Many schemes for contact transition have been proposed in literature ([25], [53]). The possibility of adopting them for the proposed controller needs to be looked into.

In conclusion, this thesis presents a new strategy for controlling constrained motion of manipulators in contact with environments whose shape is not known a priori. It has been shown experimentally that hybrid force/position control in the robot tool

space can be successfully used to control constrained motion of manipulators in contact with unknown environments.

Appendix A

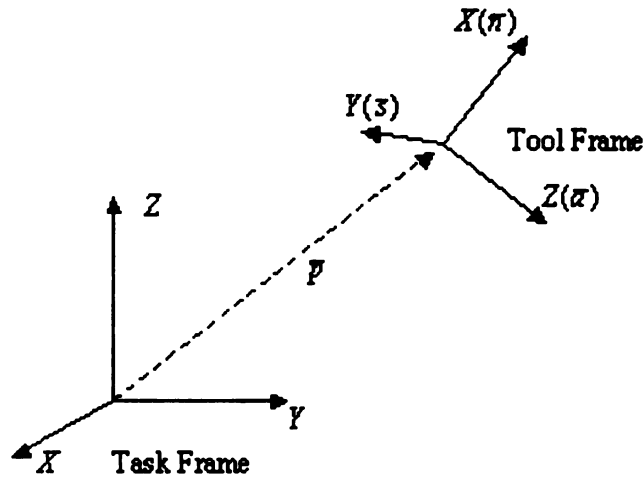
Orientation Singularity Avoidance

At the joint level control design, the measurements of joint angles are used directly in the feedback loop. Thus no forward kinematics equations need to be solved. In the task level control design, the joint angles and joint velocities have to be converted to positions and velocities in Cartesian space, which is a three-dimensional Euclidean space, through forward kinematics and the Jacobian matrix. For tool level control design also, the position, orientation and velocity of the end-effector are expressed in a Cartesian frame attached to the end-effector and they are derived from the joint angles and velocities through forward kinematics, coordinate transformations and the tool Jacobian matrix.

The output position of the end-effector consists of three components for linear position and three components for orientation in Cartesian space. It is known that the hand frame of the robot is related to a reference coordinate frame by a homogenous transformation T :

$$T = \begin{bmatrix} n & s & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix}_{4 \times 4} = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where ' n ' is the normal vector of the hand, ' s ' is the sliding vector of the hand, ' a ' is the approach vector of the hand and ' p ' is the position vector of the hand which points from the origin of the base coordinate system to the origin of the hand coordinate system. Figure A.1 shows the 4 vectors n , s , a and p . $[n \ s \ a]$ form a rotation matrix which represents the orientation of the end-effector and ' p ' is the translation vector representing the position of the end-effector.



· Figure A.1: The n , s , a and p vectors of the tool coordinate frame

Although the rotation matrix provides a convenient way to describe the orientation, nine elements are required to completely describe the orientation of a rotating rigid body. But the orientation of a rigid body in space can be described by using only three elements. Thus the orientation matrix representation introduces redundancy in the orientation representation. Moreover, the end-effector has only six degrees of freedom, so only six variables can be individually controlled. Hence instead of using three vectors in the rotation matrix, three angles can be used to represent and control the orientation.

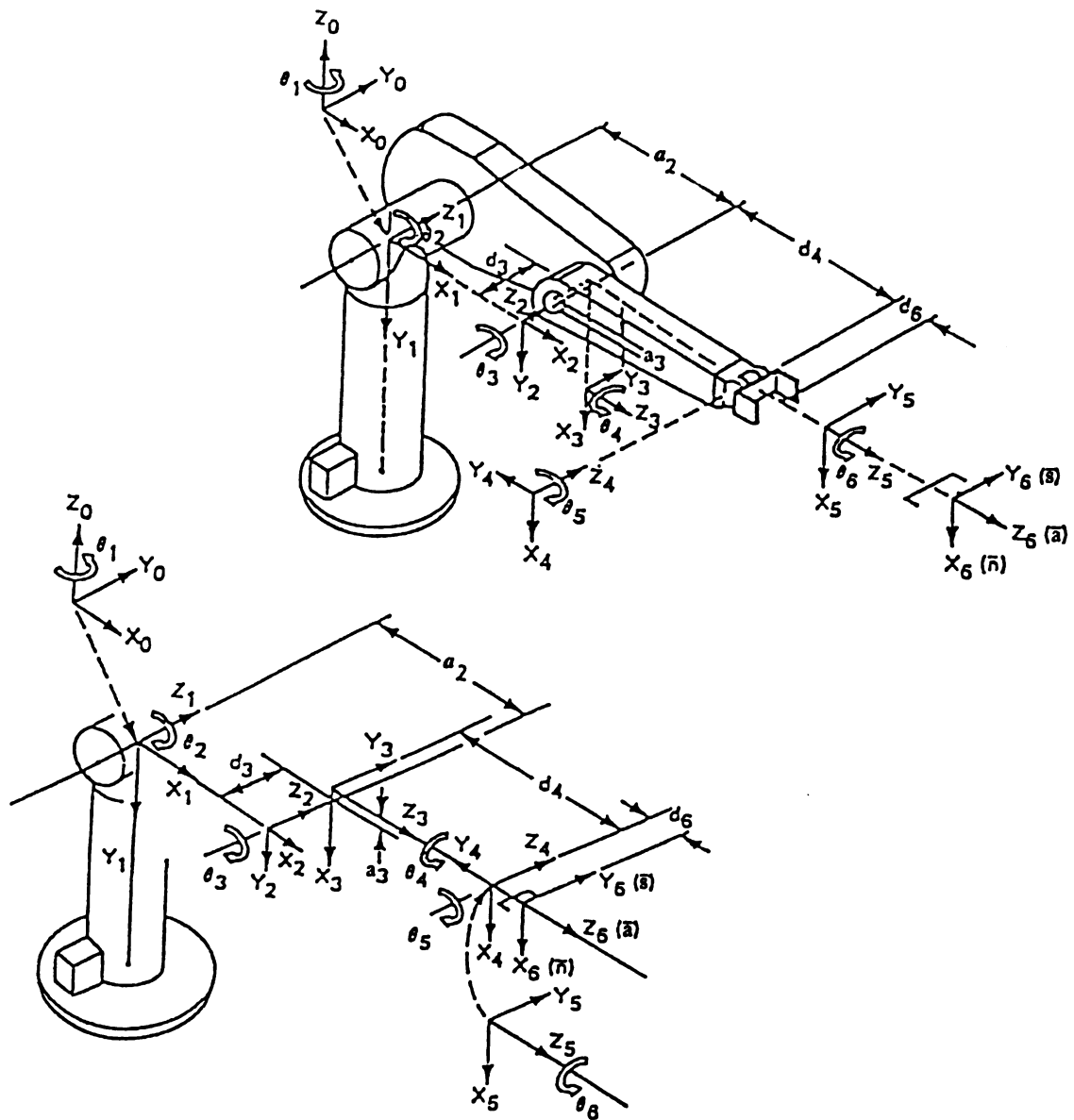


Figure A.2 Coordinate Assignments for the PUMA 560

Furthermore the elements in the rotation matrix in the transformation must be expressible by three angles.

The output velocity of the end-effector for both task level control and tool level control, consist of three linear velocity components and three angular velocity components in Cartesian space. The task space velocity is related to the joint velocities by the Jacobian matrix.

$$\begin{bmatrix} v \\ \omega \end{bmatrix}_{task} = J(q)\dot{q}.$$

The tool Jacobian, $\mathfrak{J}(q)$, relates the Cartesian tool space velocity and the joint velocities.

$$\begin{bmatrix} v \\ \omega \end{bmatrix}_{tool} = \mathfrak{J}(q)\dot{q}.$$

In the above equations $\dot{q}$ is the joint velocity, v is the linear velocity in Cartesian space and ω is the angular velocity in Cartesian space.

Here we note that the Cartesian space is a three-dimensional Euclidean space. The linear velocity and acceleration vectors in a Euclidean space are the first and second time-derivatives of the position vector ' p '. It is known, however, that the angular velocity, ω , is not the derivative of any intuitive orientation representation. Therefore the representation of the orientation requires extra consideration.

A.1 The Orientation Representation of Unimation

The orientation representation for a Unimation robot is quite an intuitive one, called OAT-angles, where O represents Orientation, A represents Altitude and T represents Tool angles. They are defined as:

$O \in (-\pi, \pi)$: the angle between world Y axis and the projection of the vector ' a ' on the world XY plane

$A \in (-\frac{\pi}{2}, \frac{\pi}{2})$: the angle formed between the vector ' a ' and a plane parallel to the world XY plane.

$T \in (-\pi, \pi)$: the angle formed between the vector ' s ' and a plane parallel to the world XY plane.

The advantage of this definition is in ease of visualization. However, there is no direct relationship between the rotation matrix and the angles. Hence it cannot be used in task or tool level orientation control.

A.2 Euler Angles and Euler OAT

Euler angles describe the orientation of a rigid body with respect to a fixed reference frame. It corresponds to a sequence of rotations, resulting in the rotation matrix $[n \ s \ a]$. There are many types of Euler angles. One type is defined as follows:

1. A rotation of ϕ angle about the Z axis.

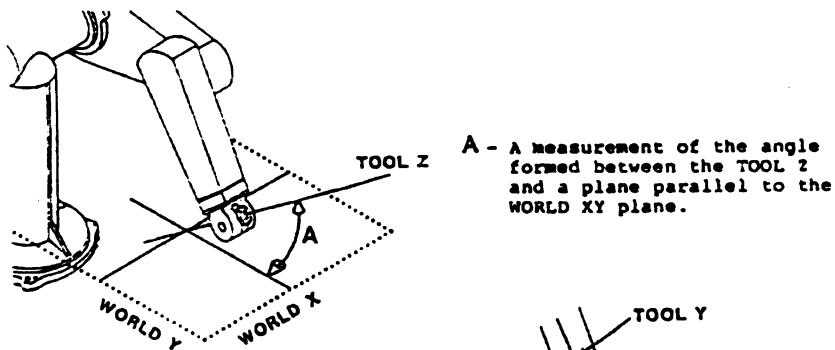
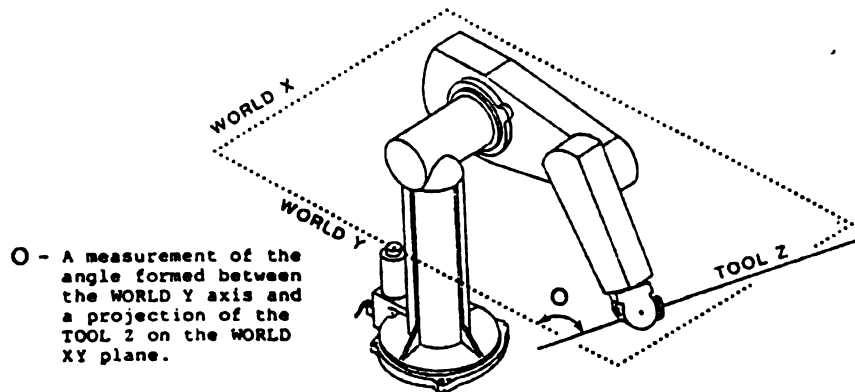
2. A rotation of ϕ angle about the rotated U axis.
3. Finally a rotation of ψ angle about the rotated W axis.

The resultant Eulerian rotation matrix is:

$$R_{\phi\theta\psi} = R_{z,\phi} R_{U,\theta} R_{W,\psi}.$$

[58] used one type of Euler angles for the PUMA 560 robot arm and considered it as the OAT solution of Unimation. However, it has been proved to be incorrect both by direct derivation and experimentally. The definition given by [58] is actually a different definition. The definitions of O and A are the same, but the definition of T is different. The Unimation OAT is not Euler angles and does not have a relation to the hand transformation matrix. Let's call Fu's angles Euler OAT to distinguish between the two definitions. The relationship between the hand transformation and the Euler OAT angles is given by:

$$\begin{bmatrix} n_x & s_x & a_x \\ n_y & s_y & a_y \\ n_z & s_z & a_z \end{bmatrix} = R_{z,O_e} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{bmatrix} R_{s,A_e} R_{a,T_e}.$$



T - A measurement of the angle formed between the TOOL Y and a plane parallel to the WORLD XY plane.

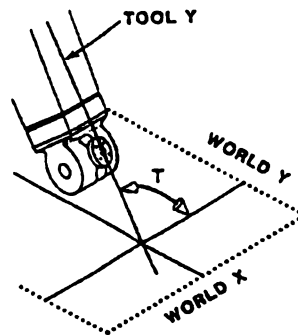


Figure A.3: The Unimation OAT definition.

The tool coordinate system is aligned with the base coordinate system of the robot initially where $O = A = T = 0$. Figure A.x and Figure A.y represent the Unimation OAT definition and the Euler OAT definitions respectively. The difference between the two definitions is evident from the figures. The kinematic characterization of the Puma 560 robot derived in [68] is based on the Euler OAT in [58], though originally it is considered as Unimation OAT according to [58]. The orientation control used in our Laboratory is based on the derivation of the OAT Jacobian and other relations presented in [60]. Fortunately, the misunderstanding does not affect our orientation control.

However, it is easy to see that when the tool ' a ' is perpendicular to the XY plane, the Euler OAT has no definition. Thus when the manipulator is perpendicular to the XY plane the Euler OAT cannot be defined. Also for tool level control, the actual position of the end-effector is always aligned with the tool space axis. Thus, irrespective of the position of the manipulator, the end-effector ' a ' is always perpendicular to the tool XY plane. Thus Euler OAT has no definition for the actual position of the end-effector in the tool frame of reference. A way needs to be found to overcome this singularity.

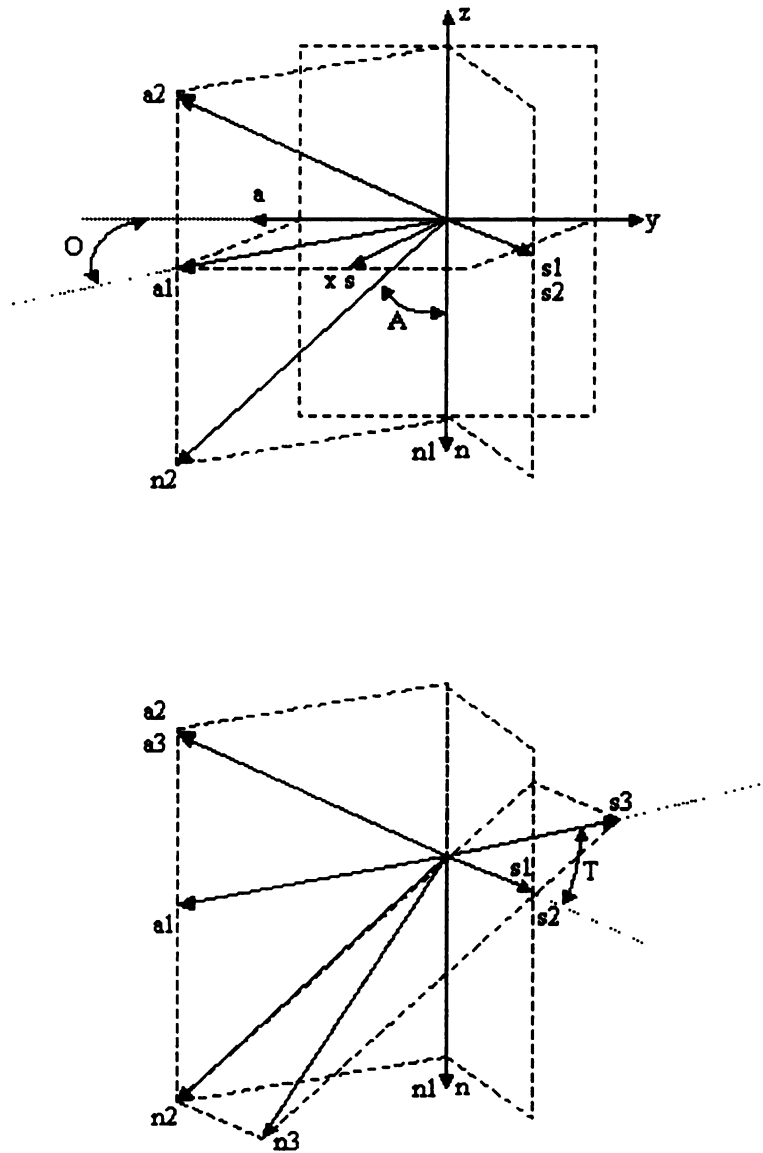


Figure A.4: The Euler OAT angles system

A.3 Singularity Avoidance

From the above discussion it is clear that the occurrence of the singularity mentioned in the previous section depends on the definition of the set of angles. The straightforward solution to this problem would be to define another set of angles which are not affected by this singularity. However, this solution is not realistic and it is time consuming because of the need for re-derivation of all the characteristics of the PUMA 560 robot arm kinematics related to the orientation definitions, and their re-implementations. Furthermore, singularities may occur in other directions, which might also be used frequently in future experiments.

The best solution for this problem is to relocate the singularity to a less frequently used orientation for this particular problem. To see this let's define:

$$[n' \ s' \ a'] = [n \ s \ a] \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix} = [a \ -s \ n].$$

Figure A.5 shows the definition of $[n' \ s' \ a']$ with respect to the original $[n \ s \ a]$.

After some elementary manipulations, the following can be arrived at:

$$\begin{aligned} T'_e &= \text{atan2}(s'_z, -n'_z) = \text{atan2}(-s_z, -a_z) \\ A'_e &= \text{atan2}(-a'_z, -n'_z CT'_e + s'_z ST'_e) = \text{atan2}(-n_z, a_z CT'_e - s_z ST'_e) \\ O'_e &= \text{atan2}(n'_y ST'_e + s'_y CT'_e, n'_x ST'_e + s'_x CT'_e) = \text{atan2}(a_y ST'_e - s_y CT'_e, a_x ST'_e - s_x CT'_e) \end{aligned}$$

,where ST'_e and CT'_e are the sine and cosine of the tool angle T'_e respectively. It is easy to see that the singularity is relocated to an orientation that is not frequently used.

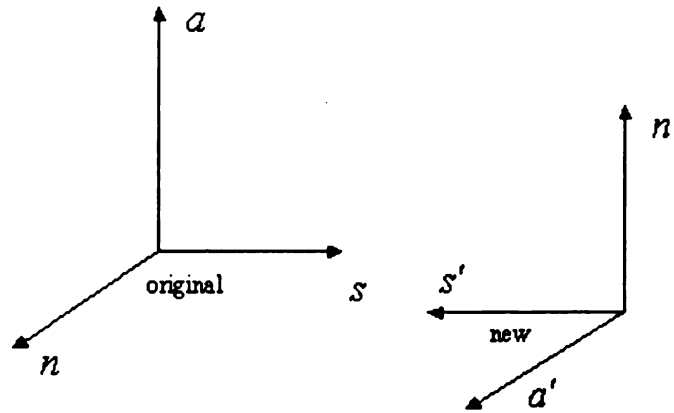


Figure A.5: The definition of $[n' \ s' \ a']$

Appendix B

The Tool Jacobian

In task level control the Jacobian matrix J is used to convert the joint velocities to the Cartesian space velocities. Similarly for tool level control the tool Jacobian matrix, $\mathfrak{J}$, is used to convert the joint velocities to the Cartesian tool space velocity. This appendix presents the equations used to calculate the tool Jacobian matrix for the Puma 560 robot arm.

Cartesian space velocities consist of 2 components viz. the linear velocity and angular velocity. The linear velocity is the first derivative with respect to time of the linear position. However, the angular velocity allows no such intuitive definition. The task Jacobian matrix for the PUMA 560 robot arm is given in [68]. This Jacobian matrix relates the task space velocities with the joint space velocities as:

$$\dot{y} = \begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} = \begin{bmatrix} v \\ \omega \end{bmatrix} = J(q)\dot{q} \quad (\text{B.1})$$

B.2 The Forward Kinematics

The forward kinematics of a robot manipulator relate the Cartesian position and orientation of the end-effector of the manipulator with respect to the base coordinate

frame to the measured joint angles. Form [68] the forward kinematics of the PUMA 560 manipulator are given as follows:

$$T = {}^0A_6 = \begin{bmatrix} n & s & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{B.2})$$

where the column vectors are given as:

$$\begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \begin{bmatrix} C_1[C_{23}(C_4C_5C_6 - S_4S_6) - S_{23}S_5C_6] - S_1[S_4C_5C_6 + C_4S_6] \\ S_1[C_{23}(C_4C_5C_6 - S_4S_6) - S_{23}S_5C_6] + C_1[S_4C_5C_6 + C_4S_6] \\ -S_{23}(C_4C_5C_6 - S_4S_6) - C_{23}S_5C_6 \end{bmatrix}$$

$$\begin{bmatrix} s_x \\ s_y \\ s_z \end{bmatrix} = \begin{bmatrix} C_1[-C_{23}(C_4C_5S_6 + S_4C_6) + S_{23}S_5S_6] - S_1[-S_4C_5S_6 + C_4C_6] \\ S_1[-C_{23}(C_4C_5S_6 + S_4C_6) + S_{23}S_5S_6] + C_1[-S_4C_5S_6 + C_4C_6] \\ S_{23}(C_4C_5S_6 + S_4C_6) + C_{23}S_5S_6 \end{bmatrix}$$

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} C_1(C_{23}C_4S_5 + S_{23}C_5) - S_1S_4S_5 \\ S_1(C_{23}C_4S_5 + S_{23}C_5) + C_1S_4S_5 \\ -S_{23}C_4S_5 + C_{23}C_5 \end{bmatrix}$$

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} C_1[d_6(C_{23}C_4S_5 + S_{23}C_5) + d_4S_{23} + a_2C_2 + a_3C_{23}] - S_1[d_6S_4S_5 + d_3] \\ S_1[d_6(C_{23}C_4S_5 + S_{23}C_5) + d_4S_{23} + a_2C_2 + a_3C_{23}] + C_1[d_6S_4S_5 + d_3] \\ d_6(C_{23}C_5 - S_{23}C_4S_5) + d_4C_{23} - a_2S_2 - a_3S_{23} \end{bmatrix}$$

where S_1 and C_1 represent the sine and cosine of the joint angle q_1 and so on.

a_2, a_3, d_3, d_4 and d_6 are the link parameters as described in [68].

It is important to note that the forward kinematics convert the coordinates of a point expressed in the end-effector frame to the coordinates of that same point expressed in the base frame with the help of the joint angles. The joint angles and the link

parameters help to express the Cartesian position and orientation of the coordinate frame attached to the i^{th} link with respect to that of the $(i-1)^{th}$ link. The position and orientation of the end-effector in the n^{th} coordinate attached to the end-effector is given by $I_{4 \times 4}$. Thus T gives the position and orientation of the end-effector in the base coordinate frame.

Let $[h_\alpha]_{4 \times 4}$ be the homogenous coordinates representing position and orientation of a point in the end-effector coordinate frame. The homogenous coordinates, $[y_\alpha]_{4 \times 4}$, of that point in the base frame are given by:

$$[y_\alpha]_{4 \times 4} = T_{4 \times 4} [h_\alpha]_{4 \times 4}$$

Thus the forward kinematics can be used to express the position and orientation of a point expressed in the end-effector frame.

B.3 Tool Space Velocity

The task space and the tool space are 3 dimensional Euclidean spaces. Thus one would require 6 dimensional vectors to represent the position and velocity in these spaces. For the position vector 3 dimensions would represent linear position and the other 3 dimensions would represent orientation. For the velocity 3 dimensions would be used for representing the linear velocity and the other 3 dimensions for representing angular velocity. Let T be the homogenous transformation that relates the homogenous tool space position and orientation, $[h_\alpha]_{4 \times 4}$ to the homogenous task space coordinates $[y_\alpha]_{4 \times 4}$. Equation B.2 gives the homogenous transformation T . The relation between the velocity

in the task space and the velocity in the tool space can be expressed with the help of T as [57]:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix}_{tool} = \underbrace{\begin{bmatrix} n_x & n_y & n_z & (p \times n)_x & (p \times n)_y & (p \times n)_z \\ s_x & s_y & s_z & (p \times s)_x & (p \times s)_y & (p \times s)_z \\ a_x & a_y & a_z & (p \times a)_x & (p \times a)_y & (p \times a)_z \\ 0 & 0 & 0 & n_x & n_y & n_z \\ 0 & 0 & 0 & s_x & s_y & s_z \\ 0 & 0 & 0 & a_x & a_y & a_z \end{bmatrix}}_{\lambda} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix}_{task}. \quad (B.3)$$

Thus

$$\dot{h} = \lambda(q) \dot{y},$$

where $\dot{h}$ and $\dot{y}$ represent the velocities in tool and task spaces respectively

Thus the velocity in the tool space can be derived from the velocity in the task space. For the relationship of the acceleration in the tool space to the acceleration in the task space, the above equation needs to be differentiated once with respect to time:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\omega}_x \\ \ddot{\omega}_y \\ \ddot{\omega}_z \end{bmatrix}_{tool} = \dot{\lambda}(q, \dot{q})_{6 \times 6} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix}_{task} + \lambda(q)_{6 \times 6} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\omega}_x \\ \ddot{\omega}_y \\ \ddot{\omega}_z \end{bmatrix}_{task}. \quad (B.4)$$

B.4 Method to Calculate the Tool Jacobian

In the preceding section, the relation between the task space velocity and the tool space velocity was derived. In this section we will develop the relationship between the joint space velocity and the tool space velocity, which is called the tool Jacobian. From the Equation B.1 and Equation B.3, it is easy to deduce the following relation.

$$\dot{h} = \lambda(q)\dot{y} = \lambda(q)J(q)\dot{q} = \mathfrak{J}(q)\dot{q} \quad (\text{B.5})$$

where $\dot{h}$ is the 6×1 vector of the tool space linear and angular velocities and $\dot{y}$ is the 6×1 vector of the task space linear and angular velocities. $\mathfrak{J}(q)$ is the tool Jacobian. The expression for $J(q)$ for the Puma 560 manipulator is given in [68] by T.J. Tarn, A.K. Bejczy *et. al.*. Its derivation is based on the notion of infinitesimal translations and rotations of the tool (Whitney, 1972 [74]).

The tool space controller also requires time derivative of the tool Jacobian in order to decouple the tool space axes. The expression for the derivative of the tool Jacobian can be simply derived using the Equation B.5. Differentiating $\mathfrak{J}(q)$ once with respect to time:

$$\dot{\mathfrak{J}}(q, \dot{q}) = \lambda(q)\dot{J}(q, \dot{q}) + \dot{\lambda}(q, \dot{q})J(q) \quad (\text{B.6})$$

The expressions for $\dot{\lambda}(q, \dot{q})$ and $\dot{J}(q, \dot{q})$ can be easily derived by differentiating $\lambda(q)$ and $J(q)$ once with respect to time respectively.

REFERENCES

- [1] D.E. Whitney, "Historical Perspective and State of the Art in Robot Force Control," International Journal of Robotics Research, 6(1):3-14, Spring 1987
- [2] R. Goertz, "Fundamentals of general-purpose remote manipulators," Nucleonics 10:36-42
- [3] R. Rothchild and R. Mann, "An EMG-controlled force sensing proportional rate elbow", Proceedings 1966 Symposium on Biomedical Engineering.
- [4] H. Raibert and J. Craig, "Hybrid position/force control of manipulators," Transactions of the ASME, Journal of Dynamical Systems, Measurement and Control. 102:126-133, 1981.
- [5] N. Hogan, "Impedance control: An approach to manipulation: Part i – Theory, Part ii – Implementation and Part iii – Applications," Journal of Dynamical Systems, Measurement and Control, 107:1-24, March 1985.
- [6] R. Colbaugh, H. Seraji and K. Glass, "Impedance control for dexterous space manipulators," Proceedings of the 31st Conference on Decision and Control, 1992
- [7] A. Haze, K. Jezernik and S. Uran "Robust Impedance Control," Proceedings of the 1998 IEEE International Conference on Control Applications Sept 1998.
- [8] M. Mason, "Compliance and force control for computer controlled manipulators," IEEE Transactions on Systems, Man and Cybernetics, SMC – 11(6):418-432, 1981.

- [9] K Shin and C. Lee, "Compliant control of robotic manipulators with resolved acceleration," Proceedings of 24th Conference On Decision and Control, Dec. 1985.
- [10] T. Stepien, L. Sweet, M. Good and M Tomizuka, "Control of tool/workpiece contact force with application to robotic deburring," IEEE Journal of Robotics and Automation Vol. RA-3, No. 1, Feb. 1987.
- [11] R. Vlope and P. Khosla, "A theoretical and experimental investigation of impact control of manipulators," International Journal of Robotics Research 12(4):352-365, August 1993.
- [12] R. Anderson and M. Spong, "Hybrid Impedance Control for Robot Manipulators," IEEE Journal of Robotics and Automaiton Vol 4 No. 5 Oct 1988.
- [13] R. Fetherstone, S Thiebaut and O. Khatib, "A general contact model for dynamically decoupled force/motion Control" Proceedings IEEE International Conference on Robotics and Automation 1999.
- [14] T. Yoshikawa, "Dynamic hybrid position/force control of robot manipulators – description of hand constraints and calculation of joint driving force," IEEE Transactions on Robotics and Automation, RA –3(5):386-392, October 1987.
- [15] O. Khatib, "A unified approach for motion and force control of robot manipulations. The Operational Space Formulation," IEEE Journal of Robotics and Automation, RA – 3(1): 43-53, February 1987.

- [16] H. Kazerooni, B. Waibel and S. Kim, "On the stability of robot compliant motion control: Theory and Experiments," Transactions of ASME, Journal of Dynamical Systems, Measurements and Control, 112:417-425, 1990.
- [17] N. McClamroch and D. Wang, "Feedback stabilization and tracking of constrained robots," IEEE Transactions on Automatic Control, 33(5): 419-426, 1988.
- [18] S. Schneider and R. Cannon, "Object impedance control for cooperative manipulation: Theory and experimental results," IEEE Transactions on Robotics and Automation, 8(3): 383-395, June 1992.
- [19] T. J. Tarn, A. K. Bejczy and Ning Xi, "Hybrid position/force control of redundant robot with consideration of motor dynamics," 13th Annual Allerton Conference on Communication, Control and Computing, pp. 702-711, 1992.
- [20] R. Zhen and A. Goldenberg, "An adaptive approach to constrained robot motion control," IEEE International Conference on Robotics and Automation Vol. 2, 1995.
- [21] B. Yao and M. Tomizuka, "Robust adaptive constrained motion and force control of manipulator with guaranteed transient performance," IEEE International Conference on Robotics and Automation, Vol. 1, 1995.
- [22] L. Whitcomb, S. Arimoto and F. Ozakki, "Experiments in adaptive model-based force control," IEEE International Conference on Robotics and Automation, 1995.

- [23] D. Lokhorst and J. Mills, "Implementation of a discontinuous control law on a robot during collision with a stiff environment," IEEE International Conference on Robotics and Automation, 1990.
- [24] P. Akella, V. Parra-Vega, S. Arimoto and K. Taniguchi, "Discontinuous model-based adaptive control for robots executing free and constrained tasks," IEEE International Conference on Robotics and Automation, 1994.
- [25] J. Hyde and M. Cutkosky, "Contact transition control: An experimental study," IEEE International Conference on Robotics and Automation, 1993.
- [26] B. Brogliato, P. Orhant and R. Lozano, "On the transition phase in robotics-part 1: Impact models and dynamics, part 2: Control," Proceedings of the IAFC Symposium on Robot Control, Vol. 2, pp. 562-576, Italy, 1994.
- [27] G. T. Marth, T. J. Tarn and A. K. Bejczy, "Stable phase transition control for robot arm motion," IEEE International Conference on Robotics and Automation, 1995.
- [28] D. Chiu and S. Lee, "Robust jump impact controller for manipulators," Proceedings of the 1995 IEEE/RSJ Intl. Conf. On Intelligent Robots and Systems, Vol.1, 1995.
- [29] B. Bona and M. Indri, "Exact Decoupling of the force-position control using the operation space formulation," Proceedings of the 1992 IEEE International Conference on Robotics and Automation, May 1992.

- [30] A. Fedele, A. Fioretti, C. Manes, G. Ulivi, "On-line processing of position and force measures for contour identification and robot control," IEEE International Conference on Robotics and Automation, 1993.
- [31] H. Bruyninckx and J. De Schutter, "A systematic derivation of online motion constraint identification equations for model based compliant motion," IEEE Conference on Robotics and Automation, 1992.
- [32] F. Lange and G. Hirzinger, "Iterative self improvement of force feedback control in contour tracking," IEEE Conference on Robotics and Automation, 1992.
- [33] E. Goddard, Y. Zheng and H. Hemami, "Dynamic Hybrid velocity/force of compliant motion over globally unknown objects," IEEE Transactions on Robotics and Automation, 1987.
- [34] C. An and J. Hollerbach, "Kinematic stability issues in force control," IEEE International Conference on Robotics and Automation, Vol.2, pp. 897-903, 1987.
- [35] C. An and J. Hollerbach, "Dynamic stability issues in force control," IEEE International Conference on Robotics and Automation, Vol.2, pp. 897-903, 1987.
- [36] W. Fisher and M. S. Mujtaba, "Sufficient stability condition for hybrid position/force control," IEEE International Conference on Robotics and Automation, 1992.
- [37] J. Wen and S. Murphy, "Stability analysis of position and force control for robot arms," IEEE Transactions on Automatic Control, Vol. 36, No. 3, March 1991.

- [38] W. Townsend and J Salisbury, "The effect of coulomb friction and stiction on force control," IEEE International Conference on Robotics and Automation, 1987.
- [39] T. Yabuta, A.J. Chona and G. Beni, "On the asymptotic stability of the hybrid position force scheme for robot manipulators," Proceedings of IEEE Conference on Robotics and Automation, 1988.
- [40] N Hogan, "On the stability of manipulators performing contact tasks," IEEE Journal of Robotics and Automation 4(6): pp. 677-686, December 1988.
- [41] T. Chung, "An inherent stability problem in Cartesian compliance and an alternative structure of compliance control," IEEE Transactions on Robotics and Automation, 7(1), 1991.
- [42] S. Eppinger and W. Seering, "Three dynamic problems in robot force control," IEEE Transactions on Robotics and Automation, 8(6), December 1991.
- [43] E. Paljug, V. Kumar, T. Sugar and X. Yun, "Some important considerations in force control implementations," Proceedings of IEEE Conference on Robotics and Automation, pp. 1270-1275, 1992.
- [44] S. Eppinger and W. Seering, "Understanding bandwidth limitations in robot force control," Proceedings of IEEE Conference on Robotics and Automation, 1987.
- [45] W. Fisher and M.S. Mujtaba, "Hybrid position/force control: A correct formulation," International Journal of Robotics Research, Vol. 11(4), August 1992.

- [46] W. McCormic and H. Schwartz, "An investigation of impedance control for robot manipulators," *International Journal of Robotics Research*, Vol. 12(5), Oct. 1993.
- [47] R. Volpe and P. Khosla, "A theoretical and experimental investigation of explicit force control strategies for manipulators," *IEEE Transactions on Automatic Control*, Vol. 38(11), November 1993.
- [48] D. Meer and S. Rock, "Coupled system stability of flexible-object impedance control," *IEEE Conference on Robotics and Automation*, pp. 1839-1845, 1995.
- [49] J. Merlet, "C-surface applied to the design of a hybrid force/position robot controller," *IEEE Conference on Robotics and Automation*, pp. 1055-1059, 1987.
- [50] J. Duffy, "The fallacy of modern hybrid control theory that is based on 'Orthogonal complements' of twist and wrench spaces," *Journal of Robotic Systems*, 7(2), 1990.
- [51] M. Vukobratovic, "How to control robots interacting with dynamic environments," *Journal of Intelligent and Robotic Systems*, Vol. 19, pp. 119-152, 1997.
- [52] Yunying Wu, "Force regulation and contact transition control," Ph.D. Thesis, Washington University, Saint Louis, 1996.
- [53] T. J. Tarn, Y. Wu and Ning Xi and A. Isidori, "Force regulation and contact transition control," *IEEE Control Systems Magazine*, Vol 6(1), February 1996.
- [54] E. Degoulange, P. Dauchez and F. Pierrot, "Force control of an industrial PUMA 560 under environmental constraints: Implementation issues and experimental results," *IEEE International Conference on Robotics and Automation*, 1993.

- [55] H. Bruyninckx, J. DeSchutter and S. Dutre, "The 'Reciprocity' and 'Consistance' based approaches to uncertainty identification for compliant motion," IEEE International Conference on Robotics and Automation, 1993.
- [56] P. Sinha and A. Goldenberg, "A unified theory for hybrid control of manipulators," IEEE International Conference on Robotics and Automation, 1993.
- [57] R. P. Paul, "Robot manipulators: Mathematics, Programming and Control," The MIT Press, 1981.
- [58] K. S. Fu, R. C. Gonzalez and C. S. G. Lee, "Robotics: Control, Sensing, Vision and Intelligence," McGraw-Hill Book Company, New York, 1987.
- [59] R. J. Schilling, "Fundamentals of Robotics: Analysis and Control," Prentice-Hall, Inc., Englewood Cliffs, 1990.
- [60] T. J. Tarn, A. K. Bejczy, A. Isidori and Y. Chen, "Nonlinear feedback in robot arm control," Proceedings of the 23rd Conference on Decision and Control, Dec.1984.
- [61] Y. F. Zheng and H. Hemami, "Mathematical modeling of a robot collision with its environment," Journal of Robotic Systems, Vol. 2(3) pp. 289-230, 1985.
- [62] Unimate Industrial Robots and Cartesian coordinates, 397K1, January 1983.
- [63] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," Journal of Dynamical Systems, Measurement and Control, Vol. 108, pp. 163-171, September 1986.

- [64] M. Kircanski, N. Kircanshi, D. Lekovic and M. Vukobratovic, "An experimental study of resolved acceleration control of robots at singularities: Damped least-squares approach," *Journal of Dynamical Systems, Measurement and Control*, Vol. 119, pp. 97-101, March 1997.
- [65] S. Chiaverini, B. Siciliano and O. Egeland, "Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator," *IEEE Transactions on Control System Technology*, Vol. 2(2), pp 123-134, June 1994.
- [66] D. N. Nenchev, "Tracking manipulator trajectories with ordinary singularities: A null space approach," *International Journal of Robotics Research*, Vol. 14(4), 1995.
- [67] Jindong Tan and Ning Xi, "Hybrid system design for singularityless task level robot controllers", *Proceedings of the IEEE International Conference on Robotics Automation*, April 2000.
- [68] T. J. Tarn, A. K. Bejczy, G. T. Marth and A. K. Ramadori, "Kinematic characterization of the PUMA 560 manipulator," *Robotics Laboratory Report*, SSM-RL-91-15, December 1991.
- [69] G. Antonelli, S Chiaverini and N. Sarkar, "An explicit force control scheme for underwater vehicle-manipulator systems," *Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 136-141, 1999.
- [70] P. Fraisse, F. Pierrot and P. Dauchez, "Virtual Environment for Robot Force Control," *IEEE International Conference on Robotics and Automation* 1993.

- [71] T. Kurfess, D Whitney and M. Brown, "Verification of a dynamic grinding model,"
Journal of Dynamical Systems, Measurement and Control, Dec 1988 Vol. 110 pg.
403 409.
- [72] A. Rodic & M. Vukobratovic, "Regulator of Minimal variances in Hybrid Control
strategy of Manipulation Robots," IEEE International Conference on Robotics and
Automation. May 1992.
- [73] J. DeSchutter, "Improved force control laws for advanced tracking applications,"
IEEE International Conference on Robotics and Automation, 1988.
- [74] D. E. Whitney, "The mathematics of coordinated control of prosthetic arms and
manipulators," Journal of Dynamic Systems, Measurement and Control, Vol. 122,
pp. 303-309, 1972.

MICHIGAN STATE LIBRARIES



3 1293 02305 3634