Planning and Control During Robotic Satellite Capture

by

Stephen K. Jacobsen

B.S. with Honors, Mechanical Engineering
University of Missouri-Rolla, 2001

Submitted to the Department of Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Mechanical Engineering

at the

Massachusetts Institute of Technology

June 2003

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY

JUL 0 8 2003

LIBRARIES

© Massachusetts Institute of Technology
All Rights Reserved

Signature of Author...
Department of Mechanical Engineering
May 9, 2003

Certified by.......................................
Steven Dubowsky
Professor of Mechanical Engineering
Thesis Supervisor

Accepted by.......................................
....
Ain A. Sonin
Chairman, Department Committee on Graduate Students

BARKER

Planning and Control During Robotic Satellite Capture

by

Stephen K. Jacobsen

Submitted to the Department of Mechanical Engineering
on May 9, 2003 in Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Mechanical Engineering

ABSTRACT

Robotic technologies have played an important role on several space missions in the past, and they are likely to play an increasing role on space missions in the future. Robots are particularly well suited for work in the dangerous and demanding space environment, which is an inherently inhospitable place for humans.

There are a number of mission scenarios that could rely heavily or exclusively on space-based robots. Among the most attractive in the near future is robotic capture of on-orbit satellites for refueling or repair. The high value of disabled satellites makes servicing preferable to replacement. The danger and difficulty of using astronauts or shuttle-based manipulators to capture an uncontrolled satellite mandate the use of a free-flying robot to carry out the mission.

However, using a space robot to perform complex tasks required for uncontrolled satellite capture presents a number of challenging planning and control problems. The research presented in this thesis breaks the satellite capture mission into phases, and analyzes each phase in more detail. Planning and control problems associated with tasks in each phase are identified and a number of them are addressed. First, an optimal approach trajectory planning method is developed that accounts for important considerations such as maximizing safety, mitigating plume impingement, and minimizing fuel usage. To verify that the planned trajectories are physically realizable, a trajectory tracking control method is developed and simulated using a free-flying robot with realistic sensor and actuator models. Second, the grasp phase is studied, and several different control methods are applied to the inertially referenced manipulator control for a simplified planar case and a more complex spatial manipulator. Third the relative motion damping portion of the stabilization phase is studied. The objectives and constraints for relative motion damping are laid out, and several methods for the control of relative motion are tested.

Algorithms that are proposed as solutions to these problems are implemented through simulations of representative systems. The results of these dynamic simulations are presented for the purpose of evaluating the effectiveness of the proposed algorithms.

Thesis Supervisor: Steven Dubowsky
Title: Professor of Mechanical Engineering

ACKNOWLEDGMENTS

There are a number of people who share the credit for making this work possible, and I would like to thank them for their support and contributions throughout the process. Thanks first of all to the National Space Development Agency of Japan for their financial support of this research. Thanks to the visiting scientists from NASDA who worked with me, Hiroshi Ueno and Shuichi Matsumoto, for sharing their knowledge and experience with space systems, and thanks to Dr. Noam Brook for collaboration on controls. Thanks to the post-docs, Chris Lee and Vivek Sujan, for their intellectual input and guidance. Thanks to the entire team at the FSRL for willingly helping with any problem and, in particular, each member of the NASDA group. Thanks to the National Science Foundation for additional financial support of this research. An extra measure of thanks goes to Dr. D. for his advice and vision on several topics, and for his directing and producing work that makes research possible.

Thanks to my friends at MIT, elsewhere in the country, and around the world for encouraging me when I needed it and distracting me when I wanted it. Special thanks to my family – Mom, Mike, Steve, Kaye, Dick, Ellen, Anna, William, John, Mary, Elizabeth, Ingrid, and all the rest – for helping me keep my feet on the ground while my mind was in outer space.

TABLE OF CONTENTS

FIGURES

TABLES

CHAPTER 1 INTRODUCTION

1.1 Robotics in Space

Robots perform important tasks in space development and exploration missions, and are likely to play an increased role on the future space missions. Space is an inherently dangerous environment for humans; radiation, extreme temperature variations, vacuum, and objects in orbit with high relative velocities pose real risks for astronauts. A significant expense and technical challenge during manned space missions is ensuring the safety of the crew and providing logistics for sustained living on orbit. Robots present advantages for work in space because they are robust to harsh conditions, replaceable, capable of exerting large forces, and less expensive to keep on orbit.

The Shuttle Remote Manipulator System (SRMS or Canadarm) is the earliest and most widely used orbital robotic manipulator, and it has assisted in satellite deployment, servicing, and construction missions since 1981 [King, 2001]. The Space Station Remote Manipulator System (SSRMS or Canadarm2), shown in Figure 1.1, is a larger, more powerful, and more advanced version of the original Canadarm. It is mounted on a mobile base on the International Space Station (ISS) and used to transport and install modules on the ISS. Other advanced manipulators that have been developed for use in space include the DLR Robot Technology Experiment (ROTEX), the European Robotics Arm (ERA), and the Japanese Experiment Module Remote Manipulator System (JEMRMS), shown along with the SPDM in Figure 1.2 [DLR, ESA, NASDA, NASA].

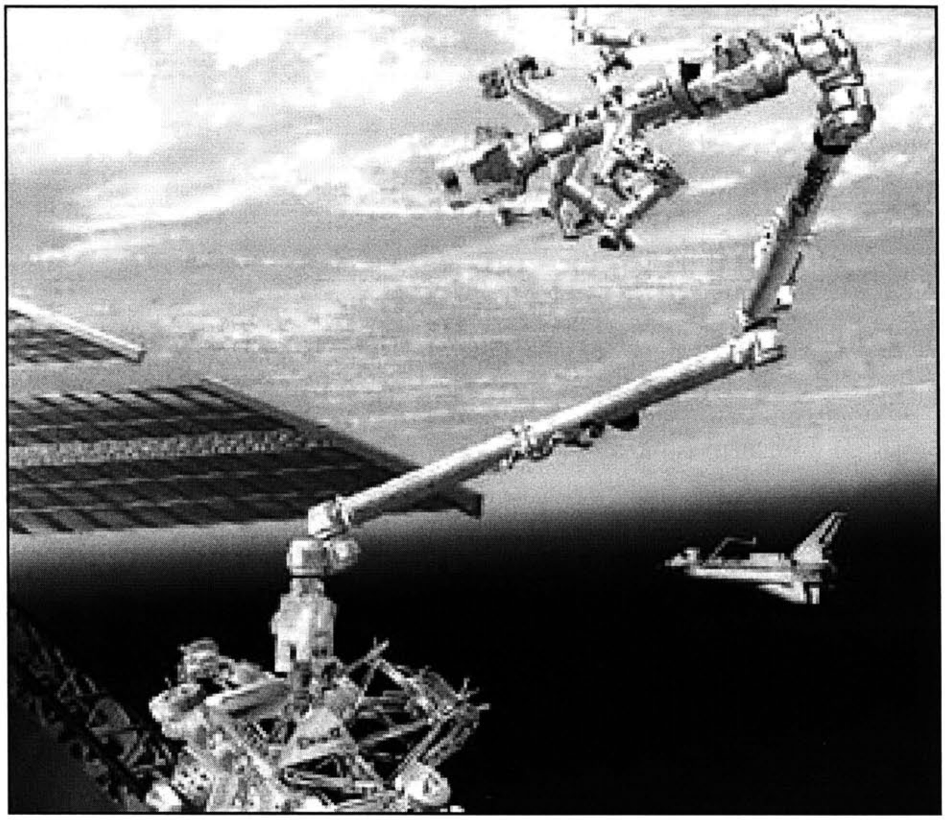


Figure 1.1: Canadarm2 and SPDM on the International Space Station [NASA]

Mission scenarios that could rely heavily on space robots in the future include the construction of large scale space structures, the collection of space debris, and the capture of satellites for servicing or repair. Robotic satellite capture is particularly attractive in the near future, because of the high value of satellites that are inoperative due to subsystem failure or improper orbital insertion and the potential to extend the life of operating satellites through refueling. In 1997, the NASDA Midori satellite with a cost of $759 million including $229 million in NASA hardware was left powerless by a simple mechanical failure when a solar panel did not deploy [NASA, 1997]. Behind the cost of the satellite and launch vehicle, insurance coverage is the largest expense for communications satellite companies. Military reconnaissance satellites could alter their orbits to respond to changing demands for intelligence if they could be refueled on orbit by a space robot. In the future, satellites could have a modular design based on Orbital Replacement Units (ORUs) so that malfunctioning subsystems could be more easily identified and replaced.

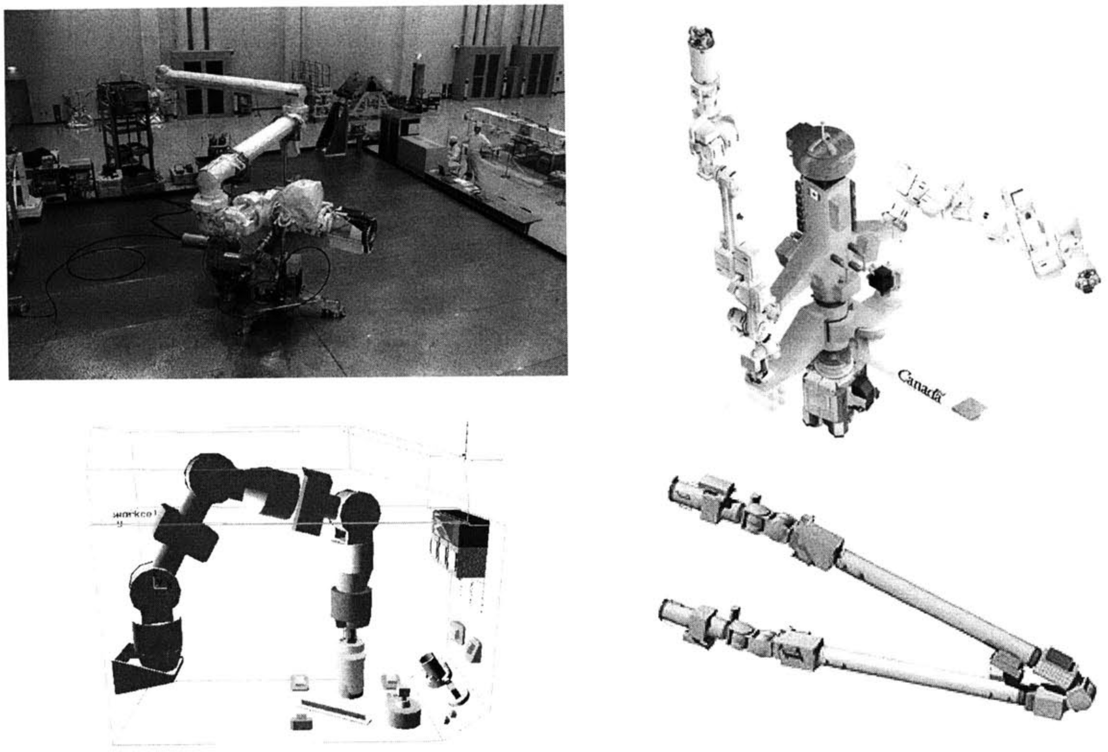


Figure 1.2: Space-Based Manipulators - JEMRMS, SPDM, ERA, and ROTEX (clockwise from top left) [NASDA, NASA, ESA, DLR]

The SRMS has been used during satellite capture and servicing missions, to grasp and retrieve satellites or provide a mobile base for astronauts while they repaired the satellites. It was used during the Hubble Space Telescope servicing mission, one of the more well known examples of on orbit satellite repair.

Engineering Test Satellite VII (ETS-VII), shown in Figure 1.3, demonstrated cooperative satellite capture by a tele-operated manipulator in 1998 [Oda, 2001]. The system consisted of two satellites, a chaser and a target, initially joined together and was launched by NASDA in order to verify navigation sensors, actuators, and control systems necessary to perform automated docking on orbit. The satellites relied on GPS receivers, laser radar, CCD cameras, and accelerometers for relative navigation. The control system featured a failure detection, isolation, and reconfiguration (FDIR) system, which was used extensively to react to and later deal with abnormal performance of attitude sensors and attitude control thrusters.

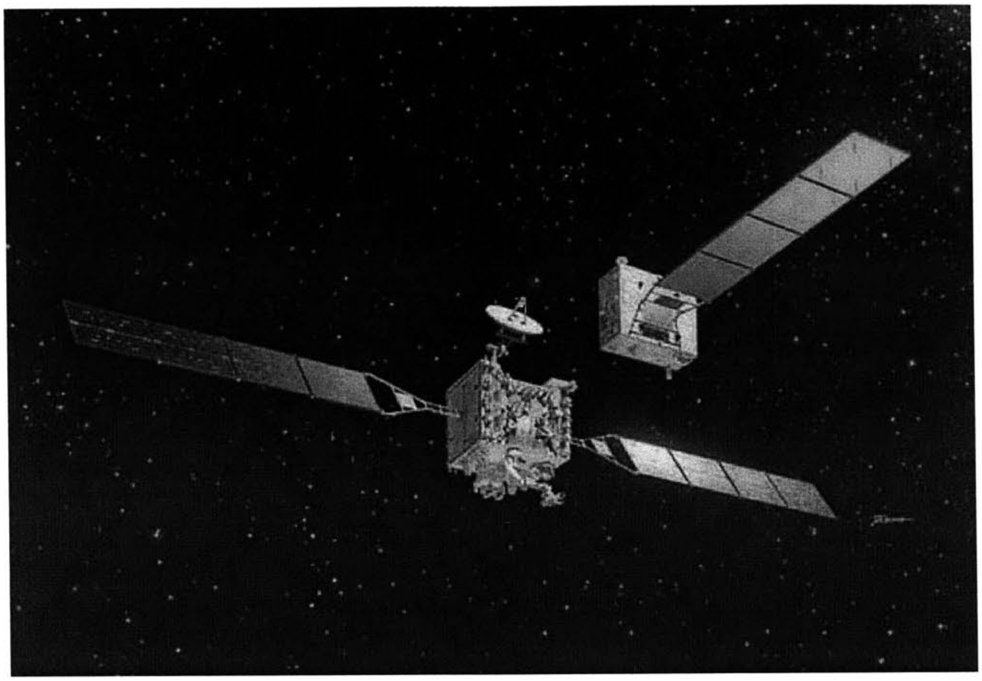


Figure 1.3: NASDA ETS-VII Target Satellite and Chaser [NASDA, 1997]

While the technologies demonstrated by these missions can be used for certain satellite capture missions, their use is limited to the capture of satellites that are cooperative and attitude stabilized. For example, the SRMS failed to capture the Spartan satellite that lost attitude control and was rotating at about 2 deg/s during the STS-87 mission in 1997 [NASA, 1997]. Existing systems do not provide the level of maneuverability and autonomy necessary for capturing targets with high spin rates or tumbling motion. Thus, more advanced robotic systems and planning and control algorithms are required for uncontrolled satellite capture. Development of these algorithms is the focus of this thesis.

A free-flying space robot equipped with a dexterous manipulator, such as the Hyper Orbital Service Vehicle (HOSV), is one advanced robotic solution that could be used. The HOSV relies on a mother ship to provide long term on orbit support equipment, including a large communications antenna, solar panels, fuel storage, and orbital transfer thrusters. By separating the functions that require significant space and power consumption yet are not required for the final capture tasks, the HOSV is made smaller and more flexible than a satellite based robotic system. The mother ship would also contain subsystem modules so that the HOSV could be reconfigured to perform various capture, maintenance, and inspection tasks.

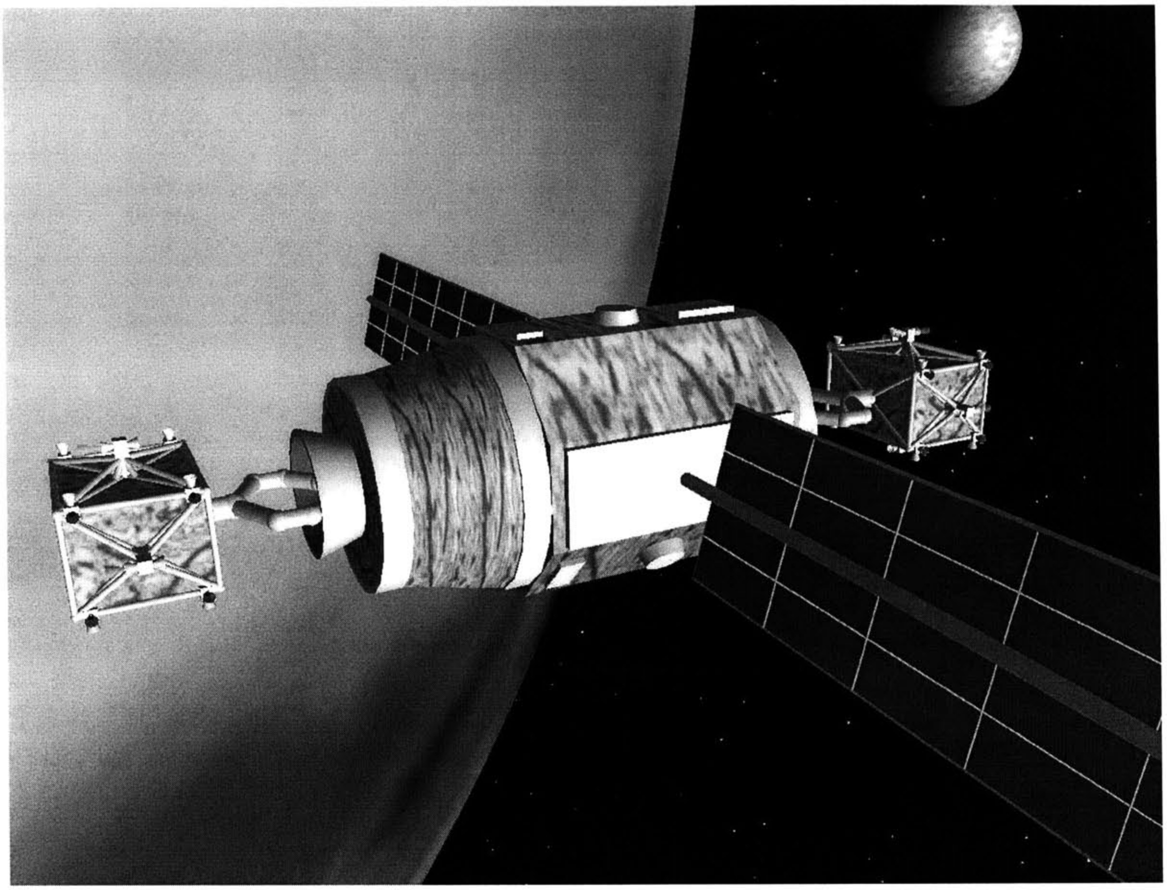


Figure 1.4: Two Robots Cooperatively Capturing a Satellite

1.2 Overview of a Satellite Capture Mission

An illustration of a representative satellite capture mission is shown in Figure 1.4. The sequence of tasks necessary for robotic satellite capture is shown in Figure 1.5 and can be roughly described by the following phases. These phases can be further subdivided into more specific tasks, and their may be considerable overlap between the phases.

1. Rendezvous – The space robot is launched and transported to the orbit of the target satellite by a spacecraft such as a rocket or Space Shuttle and released within 100-500 meters of the satellite.

2. Observation – Using onboard range and vision sensors combined with previous information about the target, the robot estimates the dynamic states of the target and makes a plan for approach and capture.

3. Approach – The robot uses thrusters and reaction wheels to control its position and attitude as approaches the satellite on a collision free approach trajectory and makes adjustment maneuvers along the way based on sensor data.

4. Grasp – In close proximity to the satellite, the robot uses its manipulator to track and grasp the specified hard point on the target, compensating for any position and velocity errors with manipulator motion to avoid high impact forces during contact.

5. Stabilization – The robot uses its manipulator and attitude control actuators to eliminate relative motion between itself and the target satellite then stabilizes the inertial motion of the combined system, stopping excessive spinning or tumbling.

6. Servicing – The satellite is either serviced on orbit by the space robot, the transport spacecraft, or returned to Earth for more demanding repairs.

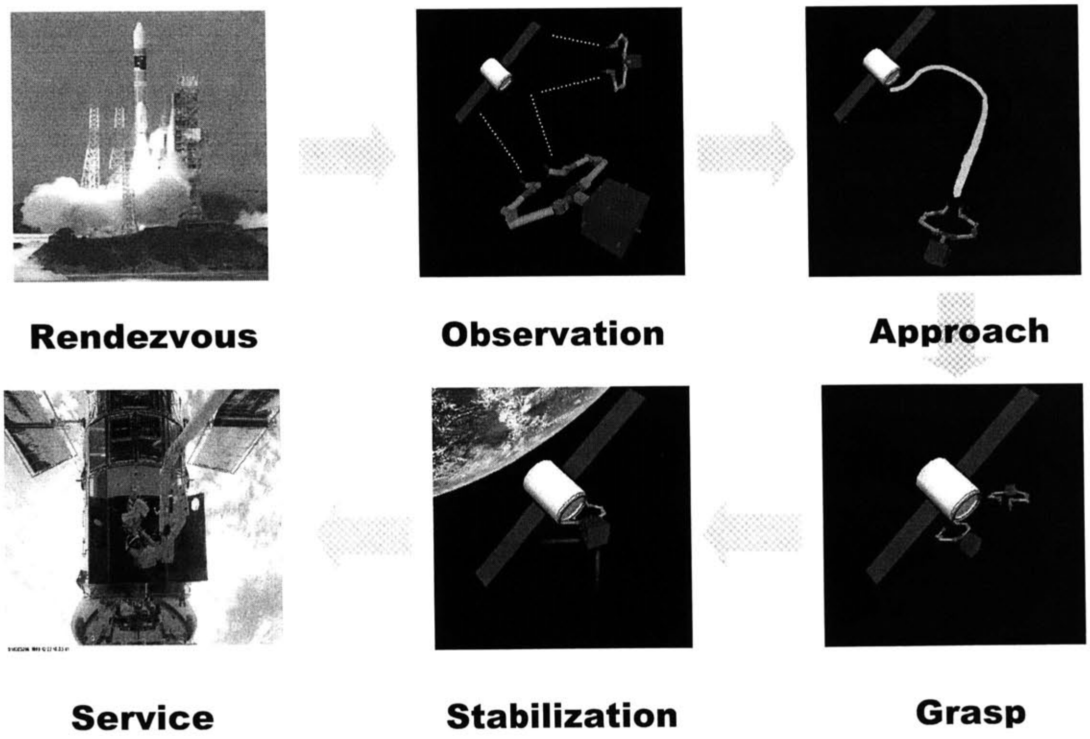


Figure 1.5: Phases in Satellite Capture Mission [Chris Lee, 2001]

This research addresses the approach, grasp, and stabilization phases in more detail. Key planning and control problems during each phase are identified, and algorithms are proposed as solutions to these problems. Simulations are used to evaluate the effectiveness of the algorithms for representative systems.

1.3 Literature Review

One important difference between the control of terrestrial and space based robots is that, in most cases, the dynamic interactions between a space-based manipulator, its payload, and its base cannot be ignored. The dynamic interactions can cause significant disturbance to the attitude and position of the base, which can severely affect the motion of the end-effecter and limit its ability to perform desired tasks. A considerable amount of research effort has been applied to investigating the problems of kinematics, dynamics, and control of space robotics, and a good collection of relevant literature can be found in [Xu and Kanade, 1993].

Vafa and Dubowsky [1987] introduced the concept of the Virtual Manipulator (VM) as a way to represent the motion of a manipulator mounted on a free floating base by an equivalent kinematic chain with its base at a fixed point in inertial space, the Virtual Ground (VG), which corresponds to the center of mass of the physical manipulator. This allows the end effecter to be controlled using traditional manipulator control techniques by using the Jacobian of the VM without tracking the motion of the base in inertial space. Torres and Dubowsky [1990] developed the Disturbance Map, which predicts the manipulator motions that will cause the maximum and minimum disturbance to the robot base and can be used to plan the most efficient trajectories possible.

Papadopoulos and Dubowsky [1989] analyzed the dynamics of free-floating robots and predicted the location of dynamic singularities. Because the base of a free-floating robot moves in reaction to manipulator motion, singularities are dependent upon the mass properties of the manipulator chain and cannot be predicted based solely upon manipulator kinematics, as with fixed base manipulators. The existence of dynamic singularities reduces the reachable workspace of a manipulator, and led to the development of the Path Dependent Workspace (PDW) and Path Independent Workspace

(PIW). A manipulator can reach any point in its PIW without encountering a dynamic singularity, while points in the PDW can be reached only if the correct end-effecter trajectory is chosen. Also, they developed the equations of motion of a space robot with the same form as those of a fixed-base manipulator. This was used to demonstrate that nearly all control methods for fixed-base manipulator can be applied to control of space robots.

The Generalized Jacobian matrix, proposed by Yoshida and Umetani [1989], is a modified form of the standard manipulator Jacobian that accounts for base motion due to dynamic interactions with the manipulator. It recognizes that the relationship between joint motion and inertial motion of a space robot is a function of the dynamic parameters of the system in addition to the kinematics. Using the conservation of system momentum, the degrees of freedom of the system are reduced to the number of manipulator joints, and the dynamic equations are written entirely as a function of the joint variables. Yoshida and Umetani used the Generalized Jacobian to develop Resolved Motion Rate Control (RMRC) and Resolved Acceleration Control (RAC) [1990] for the purposes of both manipulator trajectory control and attitude control.

The technical challenges of satellite capture are compounded if the target satellite has complex motion, such as nutation or tumbling. Therefore, Kawamoto et al. [2001] proposed a method of reducing the satellite motion to a stable spin about its principle axis of inertia through a series of mechanical impulses. The mechanical impulses could be supplied by either a manipulator end effecter or soft projectiles. They used simulation and experimental work to show that this method could be used to stabilize the motion of an uncontrolled satellite.

For planning manipulator motions, Nakamura and Mukherjee [1991] exploited the path dependent nature of space robots to plan motions which specify base orientation and manipulator configuration, using only actuation at the manipulator joints. For tasks requiring large forces, Papadopoulos and Gonthier [1995] develop a posture planning method for redundant manipulators based on min-max optimization scheme, such that the configuration that puts the smallest possible load on the actuators is chosen.

Some trajectory planning work has also been done for large scale free-flying motion as well as manipulator end-effecter motion. Sakawa [1999] used optimal control based on Pontryagin's maximum principle to minimize fuel consumption during maneuvers while satisfying required trajectory boundary conditions. Richards et. al. [2001] used Mixed Integer Linear Programming (MILP) to address the problem of planning paths to minimize fuel consumption while avoiding plume impingement during on orbit spacecraft rendezvous.

In addition to the missions described previously that studied the performance of robotic systems in space, experimental work has also been done in terrestrial laboratory settings to test guidance and control algorithms for space robots. The vacuum of space provides an environment where, in the absence of the effects of viscous friction, strong gravitational forces, or complex interaction with rough terrain, the behavior of robotic systems correspond well to that predicted by their equations of motion, encouraging the use of computer simulations for studying space robotics. However, researchers have used novel technologies to emulate a weightless environment to a limited extent and study space robotics within a laboratory environment.

At the MIT Field and Space Robotics Laboratory, Dubowsky et. al. [1994] developed the Vehicle Emulation System (VES) for studying the performance and control of manipulators mounted on mobile vehicles. The system consists of a manipulator mounted on a six degree of freedom parallel platform that can be programmed to emulate the dynamics of a moving vehicle, such as a free-floating or free-flying space robot.

Characterizing the contact dynamics that occur during when a space robot grasps a satellite is a challenging problem, due to the non-linear nature of the dynamics and the their dependence on position and velocity errors at the end-effecter. Yoshida et al. [1996] studied the problem of impact dynamics experimentally using the MIT VES, and used dynamic models to predict the motion of the bodies after contact. Based on the predicted motion, a method was developed to minimize the dynamic impulse during contact by altering the manipulator configuration and controller gains.

At the Stanford University Aerospace Robotics Laboratory (ARL), Ullman and Cannon [1991] used a robot floating on air bearings on a flat granite table to reproduce weightless motion in two dimensions. The two armed experimental robot contained on board fuel and power supplies, navigation sensors, camera, and computers for control.

The experimental system was used to demonstrate target tracking, navigation, and intercept and capture of a free-floating target.

Ejiri et. al. [1994] developed the Advanced Space Robot Testbed with Redundant Arms (ASTRA) to experimental demonstrate autonomous docking between a robot and rotating target satellite. The technical objectives of this experiment include performing visual estimation of the target satellite dynamic state, real time visual tracking control of target grasp points, and minimization of impulse during grasp by combining a flexible wrist mechanism with active impedance force control.

1.4 Thesis Outline

This thesis presents algorithms for planning and control during robotic satellite capture. The work was done within the context of a joint research program on space robotics between NASDA and the Field and Space Robotics Laboratory at MIT. The objective of this program was to advance the state of the art of space robotics and enhance the capability of NASDA to develop robotic systems for use in space. The strategy used during research was to analyze and model the dynamics of the robotic systems, and apply the resulting understanding of the system behavior to the development of effective planning and control algorithms.

The main body of this thesis consists of three chapters, each dealing with planning and control problems that arise during a particular phase of a satellite capture mission. The first chapter deals with approach trajectory planning and tracking, the second with manipulator control during grasp execution, and the third with stabilization of the robot and satellite system after grasp. The planning and control methods in each chapter are presented along with simulation results to evaluate their performance for representative scenarios. Finally, the significant conclusions from the study are summarized and topics for further research are suggested.

CHAPTER 2 APPROACH

2.1 The Approach Phase

Once the space robot is within close range (~ 100 m) of the target satellite and the target motion has been well estimated, the approach phase should be carefully planned and executed. The robot must plan an approach trajectory that is optimal with respect to performance metrics such as time and fuel consumption subject to constraints such as collision avoidance, sensor and actuator limitations, and plume impingement. As the approach is executed, the robot follows the desired trajectory and monitors it relative motion with the satellite, while tracking the motion of the grasp point on the target.

During approach, safety of the robot and satellite is the highest priority. A collision between the robot and satellite could damage delicate appendages, electronics, or optics and render worthless the large investment of time and money in a satellite servicing mission. The approach trajectory should be planned in such a way that it is collision free, the chance of collision is minimized in case of a navigation or control system failure, and a contingency escape maneuver can be enacted quickly if errors exceed acceptable levels.

The final conditions of the approach trajectory should be chosen such that the position, orientation, and velocity of the space robot are suitable for performing the grasp. To avoid large impulse forces due to the contact dynamics between the manipulator end effecter and grasp point, the motion of the robot and satellite should be well synchronized at the end of the approach phase, with minimal relative velocity between the bodies.

In this chapter, methods for approach trajectory planning are presented and applied to a representative case for satellite capture. Target satellite motion is analyzed to determine appropriate end conditions for the approach trajectories. Important considerations such as maximizing safety, mitigating plume impingement, and dealing with the effects of orbital mechanics are addressed, and solutions to these challenges are suggested. To verify that the planned trajectories are physically realizable, a trajectory tracking control method is developed and simulated using a free-flying robot with realistic models of sensor and actuator performance.

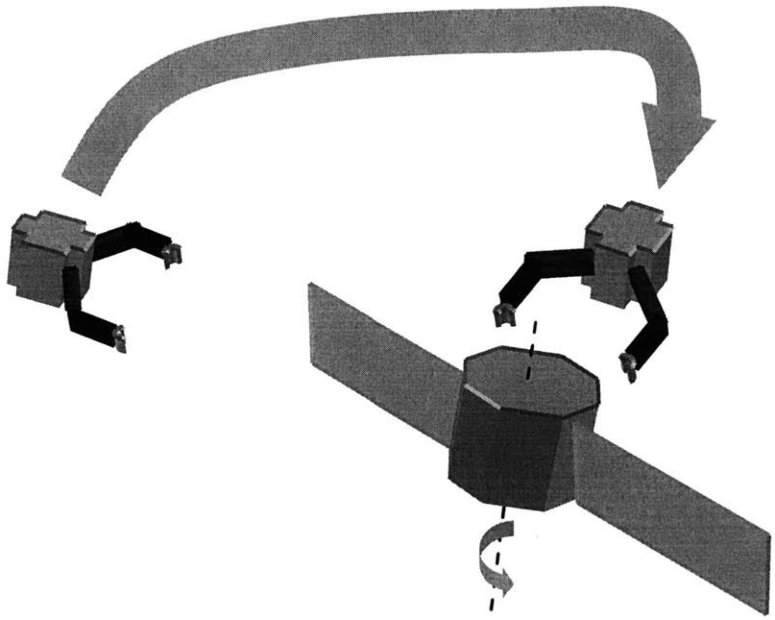


Figure 2.1: The Approach Phase

2.2 Approach Planning Methods

2.2.1 Target Assumptions

The target of a satellite servicing mission would most likely be a large expensive satellite whose value makes repair or refueling a preferable alternative to replacement. The geometry and inertial parameters of such a satellite would be well known, though there could be uncertainty in these parameters due to damage to the satellite or loss or usage of attitude control fuel.

Based on data from existing satellites in orbit, the mass of the target satellite would be in the range of one to ten tons. The target would most likely have a cylindrical or rectangular solid shape, 2 – 10 meters in length, and 1 – 3 meters in width. Appendages mounted on the target include solar panels measuring 10 – 30 meters in length, and an antenna from 1 – 30 meters. The location of points on the satellite strong enough to serve as grasping points would be known, and a likely choice for a grasp point is the Payload Attachment Fitting (PAF), which is used to secure the satellite to a rocket during launch. Other possible strong points include an attitude control thruster or base of an antenna, but a PAF is considered the primary choice for a grasping point

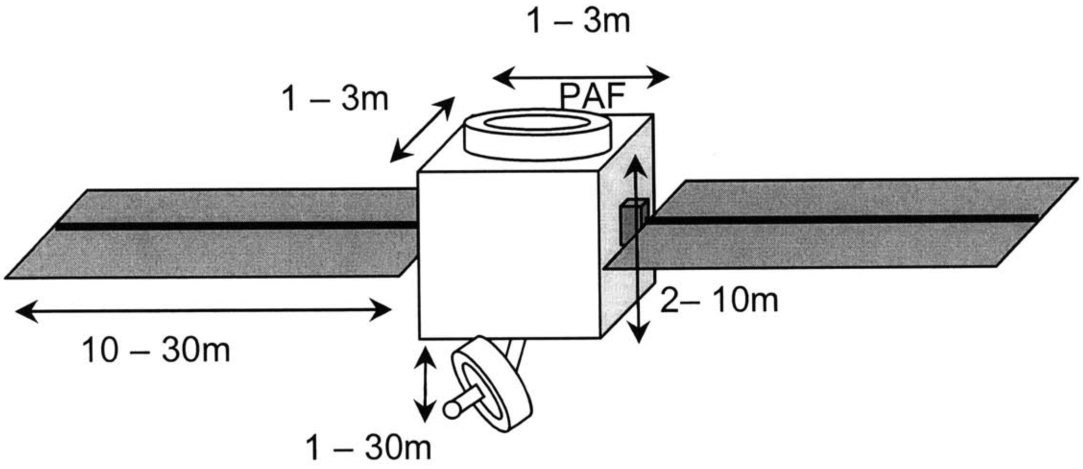


Figure 2.2: Representative Target Satellite

A satellite in low earth orbit (LEO) or geo-synchronous transfer orbit (GTO) is subject to dynamic disturbances due to gravity gradient torques and solar pressure torques. Also, the failure of an attitude control system could cause a satellite to begin tumbling or spinning, depending on the nature of the failure. However, because of asymmetric principal moments of inertia and energy dissipation through flexible appendages, the satellite will probably settle into a stable spin about its principal axis of inertia with small nutation, and any tumbling motion would most likely be slow. Based on the study of potential targets for servicing, the satellite is expected to be spinning at an angular velocity of less than 3 rpm or 18 deg/sec.

Relative motion between the space robot and target satellite is expressed using Hill's coordinate system, shown in Figure 2.3. The origin of the coordinate system is located at the center of mass of the satellite, the x axis is tangential to the satellite's direction of motion, the z axis is in the radial direction, and the y axis is perpendicular to the satellite's orbit, completing the right handed coordinate frame. The frame rotates with an angular velocity of Ω_t, equal to the orbital rate of the satellite such that the z axis is constantly oriented in the radial direction.

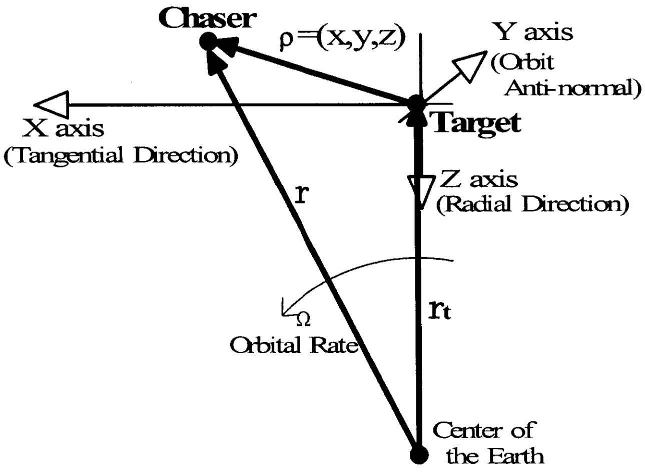


Figure 2.3: Hill's Coordinate System [Matsumoto, 2002]

Since Hill's coordinate frame moves with the satellite in its orbit and rotates at the same orbital rate, it is not an inertial reference frame. The relative motion between the satellite and another object on orbit measured in the Hill's frame is described by the following equations of orbital mechanics

$$\ddot{x} = 2\Omega_t \dot{z} + A_x \tag{2.1}$$

$$\ddot{y} = -\Omega_t^2 y + A_y \tag{2.2}$$

$$\ddot{z} = -2\Omega_t \dot{x} + 3\Omega_t^2 z + A_z \tag{2.3}$$

where A_x, A_y, A_z are external accelerations. This means that an object with an initial velocity relative to the satellite will follow a curved path rather than a straight trajectory in the Hill's frame. Over a distance of tens of meters, this curvature can be significant and should be accounted for during approach trajectory planning.

2.2.2 Trajectory Planning Using Orbital Mechanics

The relative motion between the space robot and target satellite due to orbital mechanics can be used to plan fly-by approach trajectories during satellite capture. [Matsumoto, 2001] developed a method which utilizes the equations of orbital mechanics to plan safe approach trajectories for certain cases of target satellite motion. The method uses the Clossey-Wiltshire (CW) solutions to Hill's equations to determine the initial position and velocity that will allow the space robot to approach the satellite on a collision free path and match the velocity of the desired grasp point when the robot is closest to the satellite, as shown in Figure 2.4. If the position and velocity errors at the grasp point are too large, the grasp is not executed and the robot escapes on a collision free trajectory.

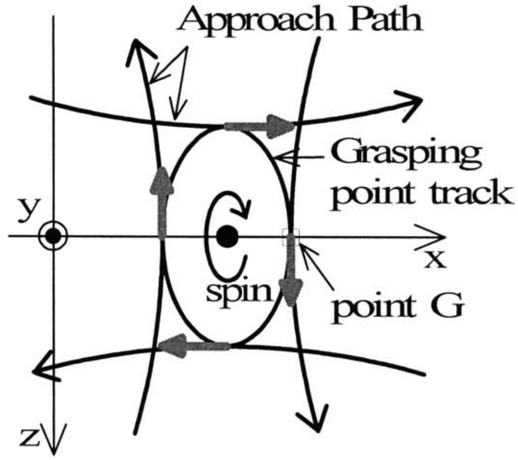


Figure 2.4: Approach Trajectories Based on Orbital Mechanics [Matsumoto, 2002]

The orbital mechanics fly-by approach planning method has the benefit that, after the initial trajectory adjustment maneuver to achieve the desired initial position and velocity, the approach trajectory is passive and does not require further trajectory tracking control as long errors are below an acceptable level. The fly-by trajectory allows for a collision free escape in the case of malfunction without high demands for computational power. However, the method may not be applied in the case where the target satellite has a high rate of spin or complex geometry, and is limited when planning the approach paths of multiple robots simultaneously. This is due to the fact that the possible shapes of passive approach trajectories are limited to those produced by the equations of orbital mechanics and are highly dependent upon velocity conditions.

2.2.3 Approach Trajectory Planning Using Numerical Optimization Methods

To cope with more complicated constraints such as targets with complex shapes, large space objects, and multiple robot systems, a trajectory planning method based on optimization techniques has been developed in this work. Optimization based trajectory planning is accomplished through three basic steps. First, the approach trajectory is expressed in terms of parameters that describe its kinematic shape along its course. Second, a cost function is defined, which evaluates the quality of the approach trajectory based on performance metrics. Finally, an optimization routine determines the values of the path parameters that yield the lowest total cost and, therefore, the optimal approach trajectory. The sequence of steps in the trajectory optimization algorithm is shown in Figure 2.5. The development and implementation of these steps is described in greater detail in the following sections.

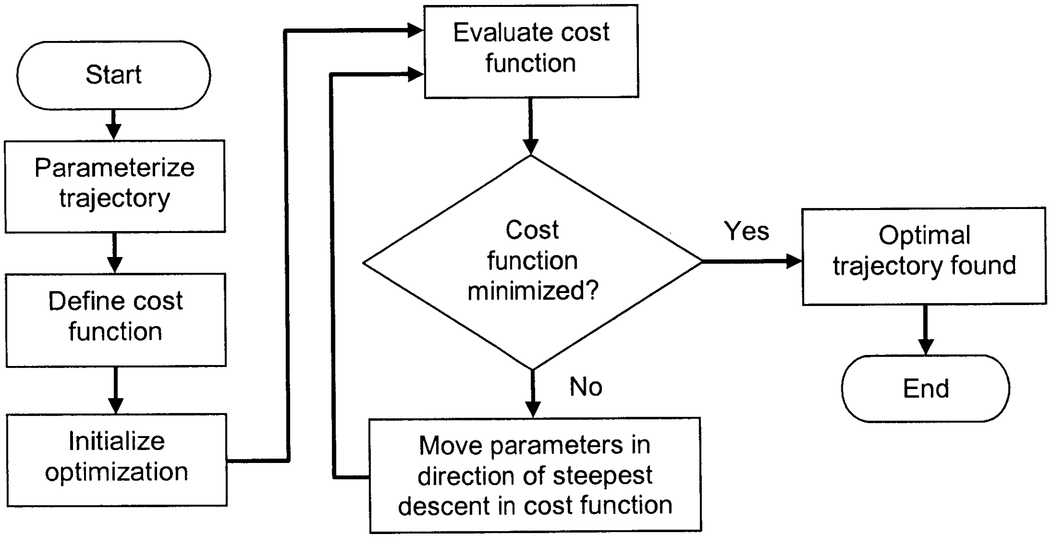


Figure 2.5: Approach Trajectory Optimization Algorithm

The performance metrics used in the cost function include important factors during satellite approach, such as safety, fuel usage, and total time required. Physical system constraints, such as maximum thruster force, can also be represented in the cost function. Preliminary analysis indicates that reasonable approach times and fuel consumption can be achieved using realistic thrusters, but maximizing safety during approach remains a critical concern.

The safety of an approach trajectory is more difficult to quantify than conventional performance metrics, such as fuel usage or time. There are many possible measures of safety during approach, such as distance to the satellite and its panels or time within an area close to the satellite. Using each of these in the cost function will produce a different choice of optimal path.

Metrics used for the cost function in this study are:

1. Safety – The safety metric applies a penalty proportional to relative velocity and distance in direction of satellite and panels, as shown in Figure 2.6. This is based on the assumption that the danger of a potential collision increases as the relative velocity between two bodies increases and the distance between them decreases.

$$s_p = \frac{\left(\overline{v}_{1\to 2} + \overline{v}_{2\to 1}\right)^2}{\left|\overline{r}_{1\to 2}\right|} \tag{2.4}$$

   where

$$\overline{v}_{1\to 2} = \overline{v}_1 \cdot \frac{\overline{r}_{1\to 2}}{\left|\overline{r}_{1\to 2}\right|} \tag{2.5}$$

2. Actuator limitations – Acceleration is limited by the maximum possible thruster forces. A term is included in the cost function that is minimal for low accelerations, but increases quickly for high acceleration.

$$a_p = e^{\alpha\left|\mathbf{a}_r\right|} \tag{2.6}$$

   where α is chosen depending upon the maximum allowable acceleration. The robot is treated as a point mass as it moves along the trajectory, so maximum acceleration is not calculated as a function of orientation.

3. Fuel – Fuel usage is estimated based on acceleration, the mass of the robot, and the specific impulse of fuel, I_{sp}.

$$f_p = \frac{m_r\left|\mathbf{a}_r\right|}{I_{sp}g} \tag{2.7}$$

4. Plume impingement – A plume impingement metric was developed for the cost function that adds a cost for acceleration of the free-flying robot away from the target satellite and its panels while it is within a certain distance of the satellite.

$$pi_p = \left(a_r \cdot \hat{e}_{r\to s}\right)^2 \tag{2.8}$$

If the robot is a distance $r \leq r_p$ from the satellite and $a_r \cdot \hat{e}_{r\to s} > 0$, this metric is applied, otherwise its value is set to zero. An analysis of the effects of plume impingement is included in Appendix A.

5. Time – The penalty applied for approach time is proportional to the total approach time squared.

$$t_p = t_r^2 \tag{2.9}$$

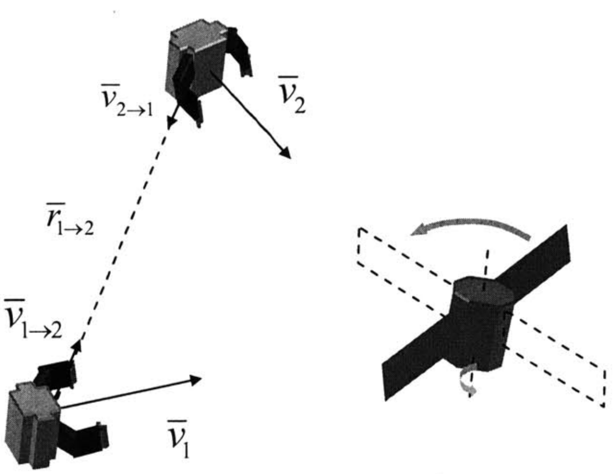


Figure 2.6: Safety Metric Based on Relative Velocity and Distance

The approach trajectory of the robot is defined by

$$T = f(\mathbf{v}) \tag{2.10}$$

where f is the form of path representation, and v is the vector of path parameters. In its most general form, the parameterization specifies both the trajectory of the robot body and the robot configuration at each point in time. Here, a subset of the general case is addressed; the translational position and velocity is accounted for by the path parameterization, but not the angular position or configuration of the robot.

The path is parameterized using polynomic splines with sufficient degrees of freedom on each spline section to match boundary conditions. Cubic splines or parabolic splines with an additional spline section fulfill the requirements of position and velocity boundary conditions for trajectory planning, and the latter are used in calculations for this case. Waypoints on each spline section combine with the initial and final boundary conditions to fully define the path.

The position at a given time t on the i^{th} spline section is given by

$$p_i(t) = x_i + v_i(t - t_i) + \frac{1}{2}a_i(t - t_i)^2 \tag{2.11}$$

where x_i and v_i at the position and velocity at the i^{th} waypoint and a_i is the constant acceleration between waypoints i and $i+1$. The velocity on the spline section is then

$$p_i'(t) = v_i + a_i(t - t_i) \tag{2.12}$$

Given the time and position at each waypoint and the initial velocity, the remaining constants can be found by

$$a_i = \frac{2}{(t_{i+1} - t_i)^2}\left(x_{i+1} - x_i - v_i(t_{i+1} - t_i)\right) \tag{2.13}$$

$$v_{i+1} = v_i + a_i(t_{i+1} - t_i) \tag{2.14}$$

For n waypoints, there must be at least n-1 spline sections defining the trajectory through the waypoints. Using quadratic splines, there are $3n$-3 coefficients to determine, but the requirements for position and velocity continuity and the initial and final velocity conditions create $3n$-1 constraints. To satisfy the additional constraints, the last section is split into two splines by placing a waypoint halfway between the last two points at

$$t_{n-\frac{1}{2}} = \frac{t_{n-1} + t_n}{2} \tag{2.15}$$

Writing the two last spline sections as $p_i(t) = a_i t^2 + b_i t + c_i$ for compactness, and applying the constraints of position and velocity continuity and boundary conditions, the conditions for the coefficients in the equations are

$$\begin{bmatrix} \frac{1}{2}t_{n-1}^2 & t_{n-1} & 1 & 0 & 0 & 0 \\ t_{n-1} & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}t_n^2 & t_n & 1 \\ 0 & 0 & 0 & t_n & 1 & 0 \\ \frac{1}{2}t_{n-\frac{1}{2}}^2 & t_{n-\frac{1}{2}} & 1 & -\frac{1}{2}t_{n-\frac{1}{2}}^2 & -t_{n-\frac{1}{2}} & 1 \\ t_{n-\frac{1}{2}} & 1 & 0 & -t_{n-\frac{1}{2}} & 1 & 0 \end{bmatrix} \begin{bmatrix} a_{n-1} \\ b_{n-1} \\ c_{n-1} \\ a_n \\ b_n \\ c_n \end{bmatrix} = \begin{bmatrix} x_{n-1} \\ v_{n-1} \\ x_n \\ v_n \\ 0 \\ 0 \end{bmatrix} \tag{2.16}$$

or

$$\mathbf{Ac} = \mathbf{x} \tag{2.17}$$

The coefficients of the last spline sections are then given by $\mathbf{c} = \mathbf{A}^{-1}\mathbf{x}$.

In general, the cost function to be optimized consists of a path integral and additional functions of path parameters

$$P(\mathbf{v}) = \int_0^{t_f} f_1(v) + f_2(v) + \ldots dt + g_1(v) + \ldots \tag{2.18}$$

based on performance metrics f_i and g_i. The cost function should be a smooth function of the path parameters so that the minimum cost can be readily found by an optimization routine.

For this study, the cost function was discretized for numerical optimization as

$$P(\mathbf{v}) = \sum_{\mathrm{m}} (\mathrm{k_a a_p} + \mathrm{k_f f_p} + \mathrm{k_s s_p} + \mathrm{k_{pi} pi_p})\Delta \mathrm{t} + \mathrm{k_t t_r}^2 \tag{2.19}$$

where m is the total number of time steps on the trajectory and k_i are weighting coefficients for the performance metrics.

For the case presented here, the robot is treated as a point mass on the path to simplify evaluation of the cost function at each point. In the more generalized case, robot configuration and end-state boundary conditions can be parameterized to allow variation in the final grasp velocity and configuration.

If the cost function used by the optimization routine contains local minima within the parameter space and the optimization routine does not rigorously search for the globally optimum solution, then supplying different initial parameter values to the routine may result in different paths being chosen as optimal. These different solutions may be equally desirable if they are caused by the existence of symmetry and yield the same value when evaluated by the cost function. Separate locally optimal solutions may, however, differ drastically from one another in their total cost, and care must be taken in the selection of initial parameter values when working with a cost function known to possess local minima. For this reason, a heuristically determined collision free path should be chosen as a starting point for an optimization routine.

By perturbing the set of parameters $\mathbf{v}$ and evaluating $P_i(\mathbf{v}_i)$, the algorithm moves $\mathbf{v}_i$ in the direction that produces the largest decrease in P and iteratively finds the parameter set $\mathbf{v}^*$ that minimizes P

$$\mathbf{v}^* = \min_{\mathbf{v}}(P(\mathbf{v})) \qquad (2.20)$$

The optimal trajectory is then described by

$$T^* = T(\mathbf{v}^*) \qquad (2.21)$$

A numerical approach to optimization is chosen for flexibility, and because the general cost function is not suitable for closed-form minimization. The Nelder-Mead minimization algorithm is used because it does not require an explicit representation of the gradient of $P(\mathbf{v})$, and is robust to the complex form of the cost function.

When the trajectory has been optimized for a given set of n waypoints, n is incremented and a new $P(\mathbf{v}^*)$ is found. This process is repeated while the addition of waypoints significantly improves the total cost of the path. When $\Delta P_{n\rightarrow n+1}$ falls below a specified tolerance, the stopping criteria is met and the optimization process is complete. The trajectory defined by the current parameter set is chosen as the optimal approach trajectory.

The computation time required for successful optimization on a given computer depends on the number of path parameters, the required tolerance for stopping the optimization routine, the complexity of the performance metrics, and the difference between the initial conditions and the optimal trajectory. For the representative cases presented in this study, optimization time was between one and ten minutes on a Pentium IV 1.5GHz system.

Optimization Results

The representative target satellite used to test the optimization-based approach planning method is the same as described in the section on target assumptions. The satellite is in a stable spin about its principal axis of inertial with an angular velocity of 5 deg/sec in the direction of the unit vector

$$\hat{e}_{\omega} = \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{bmatrix}^{T} \tag{2.22}$$

Optimization Results for Single Robot

The robot is initially located on the satellite's orbit 50 meters in the v-bar direction, with zero velocity relative to the satellite. At the end of the approach trajectory, the robot has synchronized its motion with the satellite at an offset distance from the target grasp point in preparation for grasp and stabilization.

The approach trajectory produced by optimization for this representative case is shown in Figure 2.7 and a summary of important trajectory data is given in Table 1 for the optimized trajectory and a straight line trajectory between the boundary conditions. The value of the safety metric is plotted vs. time for the approach trajectory in Figure 2.8. The metric reaches a maximum when the robot is closest to the satellite and panels, immediately prior to grasp.

Table 1: Summary of Trajectory Data

	Optimized Trajectory	Straight Line Trajectory
Time	58.9 sec	60 sec
Approx. fuel consumption	0.979 kg	1.37 kg
Maximum acceleration	0.154 m/sec^2	0.176 m/sec^2
Value of Safety Metric	5695	8615

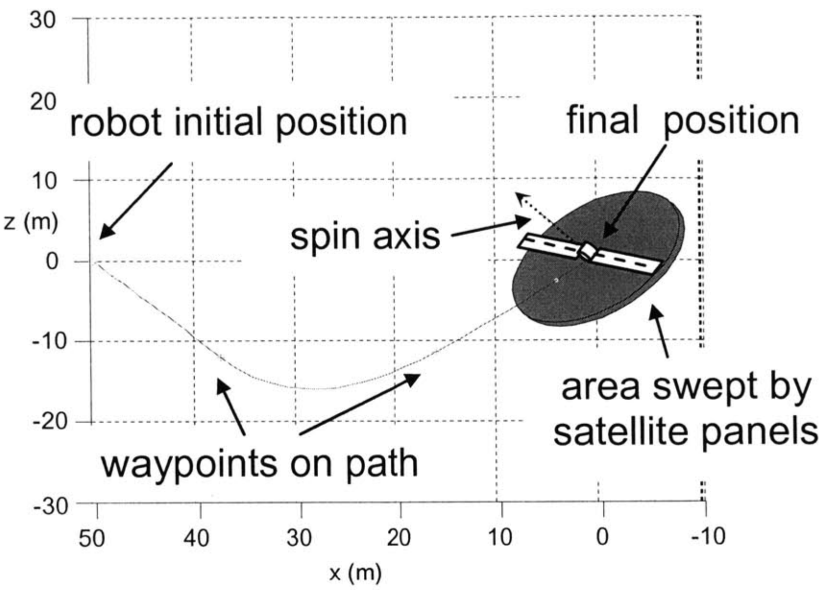


Figure 2.7: Optimal Approach Trajectory for Single Robot, x-z plane

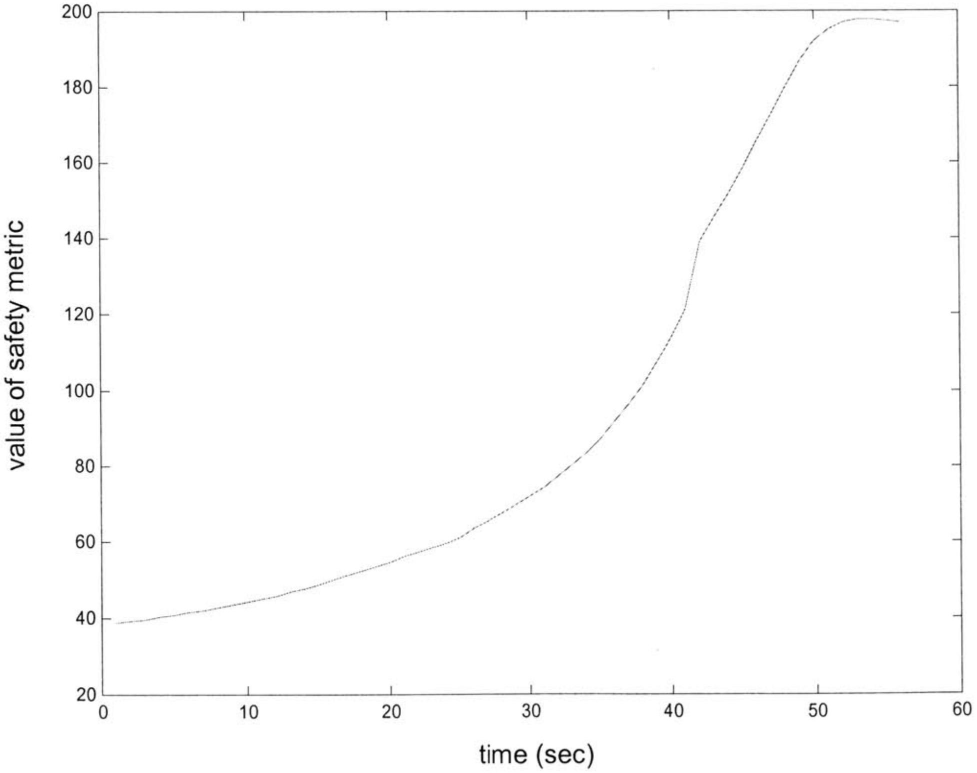


Figure 2.8: Value of Safety Metric vs. Time

Optimization Results for Multiple Robots

The optimization based trajectory planning method can be extended to the case involving multiple robots approaching a target simultaneously by evaluating the performance metrics for each body. By applying a safety metric to the relative motion between robots in addition to the relative motion of each robot with the target, the risk of collision between the robots can also be reduced.

In the representative case for multiple robots approaching simultaneously, the same target satellite is used and one robot starts at 50 m in the +v-bar direction while the other is initially at 50 m in the –v-bar direction. The result of optimization for these conditions is shown in Figure 2.9, where the robots approach on opposite sides of the spin axis to avoid each other as well as the satellite panels.

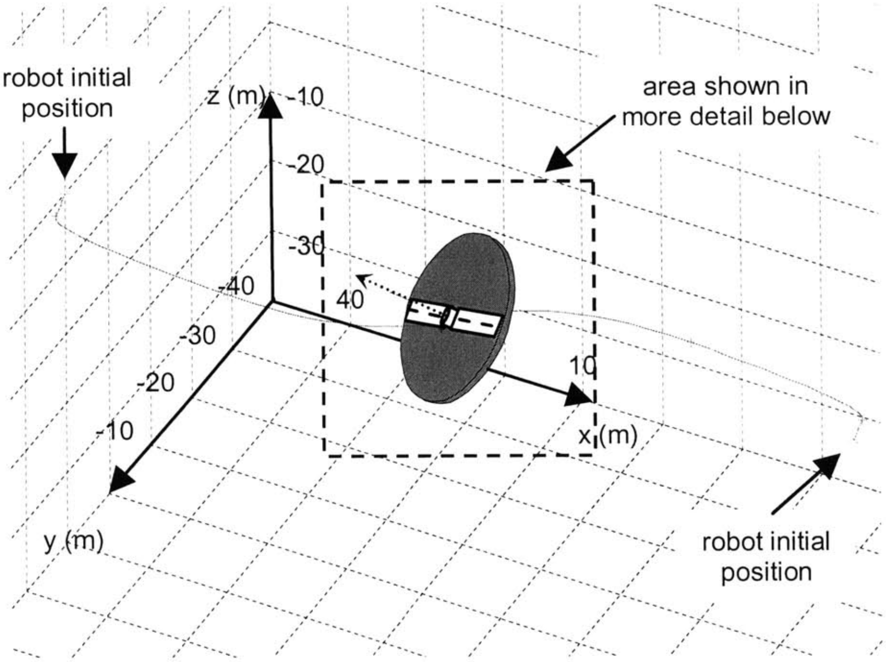


Figure 2.9: Optimal Trajectories for Multiple Robots Approaching Simultaneously

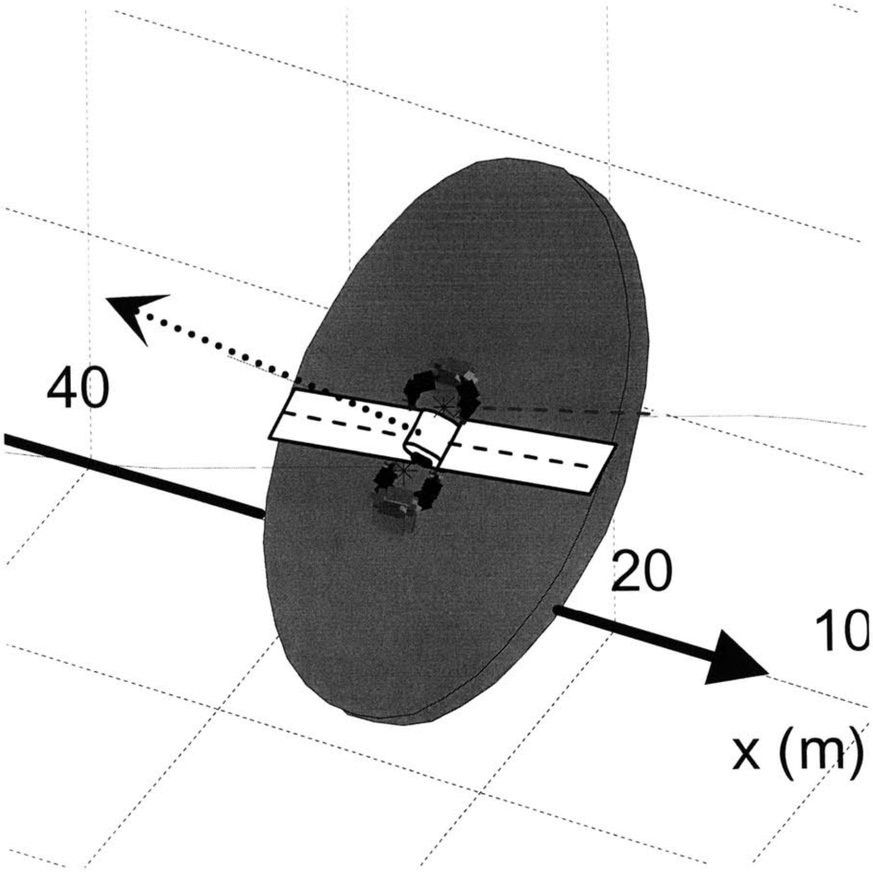


Figure 2.10: Multiple Robots Docking with Satellite

2.3 Control during Approach

At each point on the planned approach trajectory the desired accelerations are known. It is then necessary to solve for the equivalent force from each thruster which will produce the desired accelerations. The resultant forces and torques at the robot's center of mass are found by summing the individual contributions of the thrusters.

$$\mathbf{F} = \sum_{i=1}^{n} \hat{d}_i t_i \tag{2.23}$$

$$\boldsymbol{\tau} = \sum_{i=1}^{n} \left(\mathbf{r}_i \times \hat{\mathbf{d}}_i \right) t_i \tag{2.24}$$

These can be rewritten as

$$\mathbf{F} = \mathbf{Dt} \tag{2.25}$$

$$\boldsymbol{\tau} = \mathbf{Qt} \tag{2.26}$$

where **D** contains column vectors $\mathbf{d}_i$, **t** is a vector of thruster values t_i, and **Q** contains column vectors $\mathbf{r}_i \times \hat{\mathbf{d}}_i$. **D** and **Q** are originally represented in the robot frame, so the resultant forces and torques are found by rotating them into the global coordinate system.

$$\begin{bmatrix} \mathbf{F}_0 \\ \boldsymbol{\tau}_0 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_r^0 & \mathbf{D} \\ \mathbf{R}_r^0 & \mathbf{Q} \end{bmatrix} \mathbf{t} \tag{2.27}$$

In general, the number of thrusters may be greater than the number of degrees of freedom of the system, so the required thruster values are found by pre-multiplying each side of the equation by the pseudo-inverse of the matrix above.

$$\mathbf{t} = \begin{bmatrix} \mathbf{R}_r^0 & \mathbf{D} \\ \mathbf{R}_r^0 & \mathbf{Q} \end{bmatrix}^+ \begin{bmatrix} \mathbf{F}_0 \\ \boldsymbol{\tau}_0 \end{bmatrix} \tag{2.28}$$

2.3.1 Model of Free-flying Robot

The free-flying robot model used during the trajectory tracking simulations is shown in Figure 2.11 and was created in the dynamic simulation package ADAMS (Automatic Dynamic Analysis of Mechanical Systems from MSC Software). The base consists of a one meter cube with a mass of 100 kg, and each arm is 25 kg. The robot has a total of 16 thrusters, mounted in groups of four on the top, bottom, right, and left sides of the base. The joints in the arms are locked during the approach phase, but the arms give the robot realistic inertial properties and mass distribution.

Because the motion of the free-flying robot is measured in the Hill's coordinate system, which is not an inertial reference frame, a realistic simulation environment for testing free-flying control methods should account for the effects of orbital mechanics. This is accomplished by adding fictitious forces to the center of mass of the space robot in ADAMS so that the robot's motion follows the motion predicted by equations of orbital mechanics. The forces acting in the x, y, and z directions are

$$F_x = 2m_r\Omega_t\dot{z} \tag{2.29}$$

$$F_y = -m_r\Omega_t^2 y \tag{2.30}$$

$$F_z = m_r\Omega_t\left(-2\dot{x} + 3\Omega_t z\right) \tag{2.31}$$

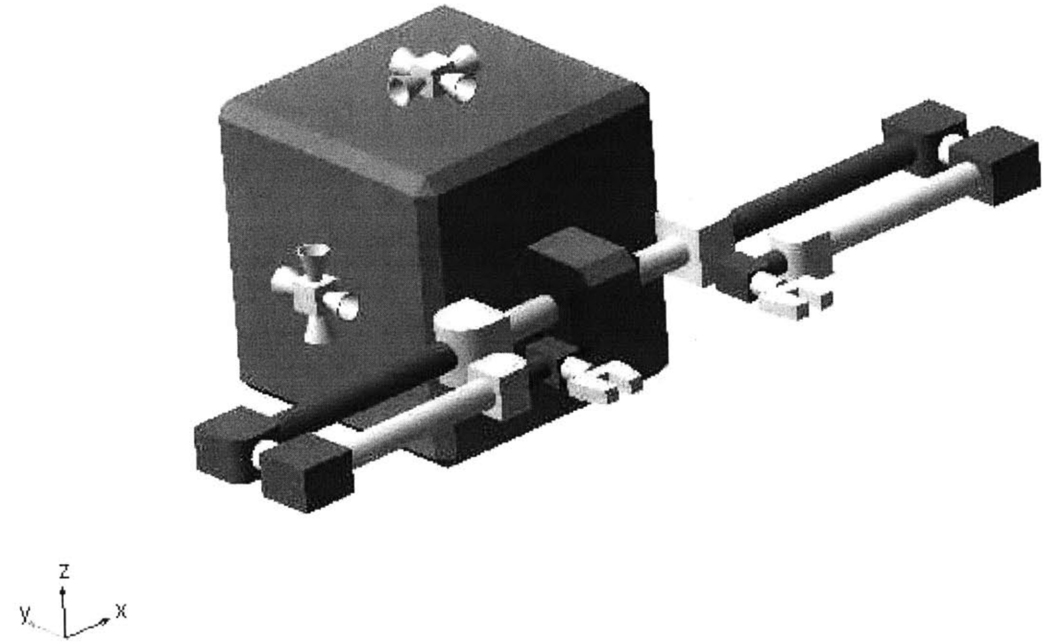


Figure 2.11: Free-flying Robot Model in ADAMS

2.3.2 Description of control system

The control system for a free-flying robot is shown in Figure 2.12. The objective of the system is to successfully follow the reference trajectory, which specifies the desired position and orientation of the robot as a function of time. Control inputs are passed to a dynamic simulation of the free-flying robot in ADAMS, and actual displacements and rotations are returned as outputs from the plant. Because the reference trajectory is specified in a coordinate frame moving with the satellite, the effects of orbital mechanics are added in the simulation environment to account for motion in a non-inertial frame. The actual trajectory is compared to the reference, and errors are processed with PD control. Errors in orientation and rotational velocity were calculated using the quaternion error vector method, described in [Sidi, 1997]. Control signals are combined with feedforward accelerations calculated from the desired trajectory to improve tracking performance. From the acceleration signals, the thruster and reaction wheel commands are found. The thruster commands are converted to pulses using pulse width modulation (PWM) to realistically represent the discrete actuators.

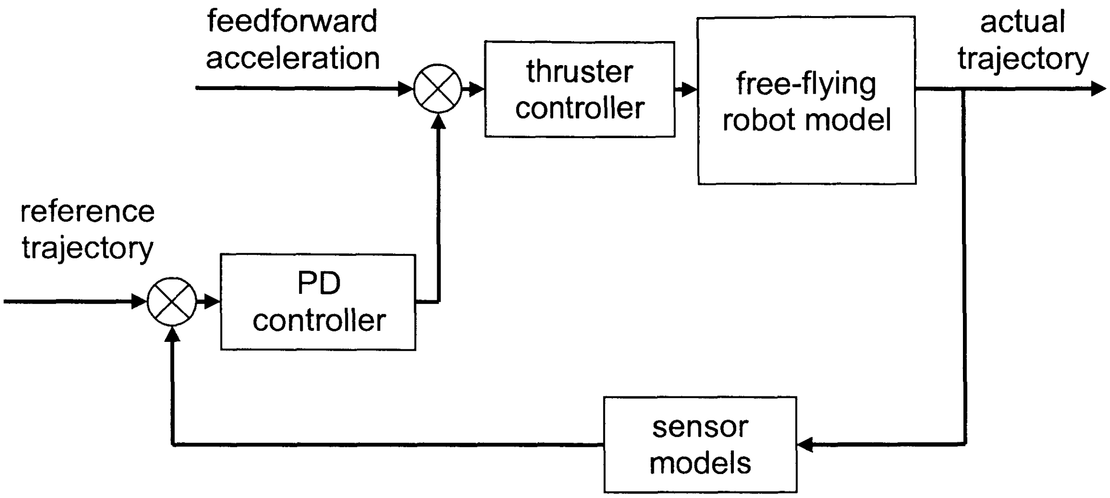


Figure 2.12: Free-flying Control System

2.3.3 Modeling of Sensors and Actuators

The performance of a realistic thruster is illustrated in Figure 2.13, along with the response of the ideal thruster, shown by the dashed lines [NASDA]. After the on command is given at t_{on}, there is a delay of $t_1 - t_{on}$ as the output increases to full power. The actual output force varies randomly about its average value F_{ave}, which is shifted from the desired force F_d. When the off command is given, there is another delay, $t_2 - t_{off}$, as the actual thruster force returns to zero.

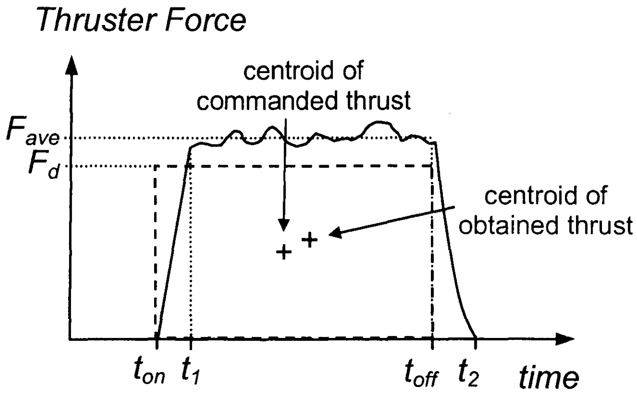


Figure 2.13: Model of Thruster

This model was implemented in MATLAB by adding random noise to the ideal thruster force. The random noise is characterized by a mean bias and a standard deviation about the bias. The thruster force was passed through a filter in MATLAB to limit the rate of change of the force between t_{on} and t_1 and between t_{off} and t_2. For the sake of simplicity and because their output can be more precisely controlled than that of the thrusters, the reaction wheels were modeled as three pure torques with saturation limits about the robot base center of mass.

Variation in sensor performance was modeled in a similar way, with a mean shift and standard deviation representing errors in sensor data. The performance of the orientation sensors was considered to be time-invariant, while the magnitude of errors in range estimates varied proportionally with distance from the target satellite. The range sensor model is based on the rendezvous laser radar (RVR) system used during final approach ($2m < d_s < 500m$) during the ETS-VII mission [Kawano, 1998]. Table 2 gives a summary of the performance specifications used when modeling the sensors and actuators.

Table 2: Sensor and Actuator Performance Specifications

	Operating Range	Bias	Standard Deviation
Hydrazine Thruster	0 – 10 N	0.02 N	0.0333 N
Reaction Wheel	± 1 N*m	0 N*m	0 N*m
Laser Range Sensor (RVR)	2 – 500 m	0.1 m	$(0.001*d_s)$ m
Orientation Sensor	0 – 360 deg	0.05 deg	0.02 deg

2.3.4 Simulation Results

The free-flying control system was tested for a number of different trajectories in order to analyze its performance for position and velocity control. Figure 2.14, Figure 2.15, and Figure 2.16 show the acceleration, velocity, and position, respectively, of the space robot in response to a commanded parabolic trajectory in the x direction with a constant acceleration of 0.1 m/sec^2. The desired trajectory is shown superimposed on the actual trajectory. As shown by the presented results, the control system tracks the desired trajectory well and produces very small errors in position and velocity. The errors in position and velocity are on the order of centimeters and centimeters per second using PD control. These errors in base position and velocity can be easily compensated for by the robot's manipulator because its maximum reach should be over a meter and it maximum endpoint velocity should be more than one meter per second.

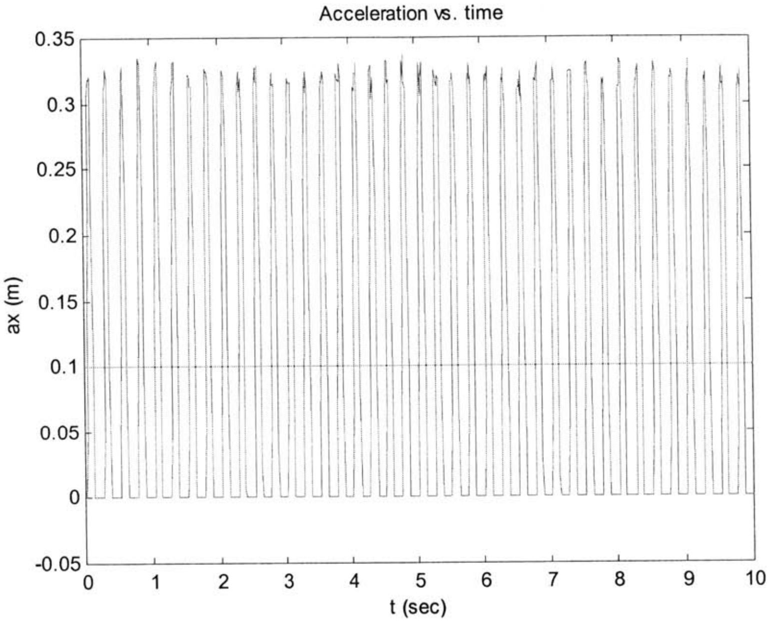


Figure 2.14: Desired and Actual Acceleration vs. Time

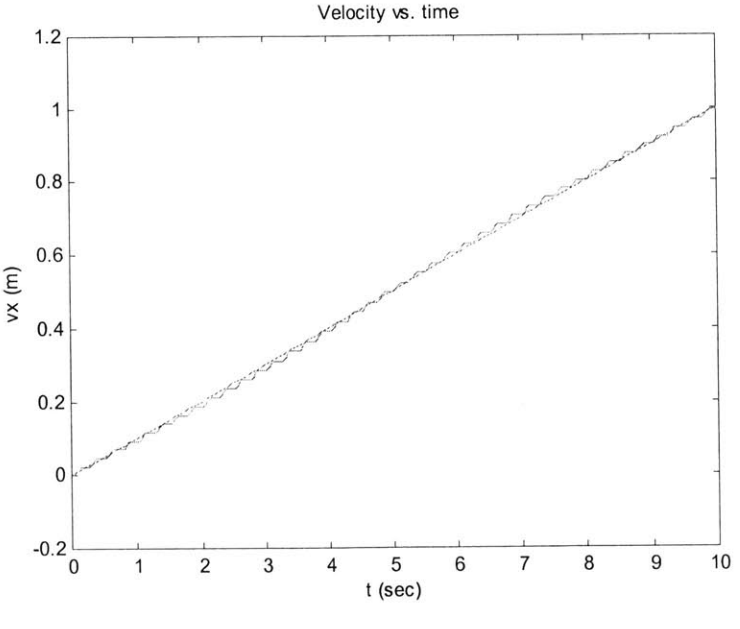


Figure 2.15: Desired and Actual Velocity vs. Time

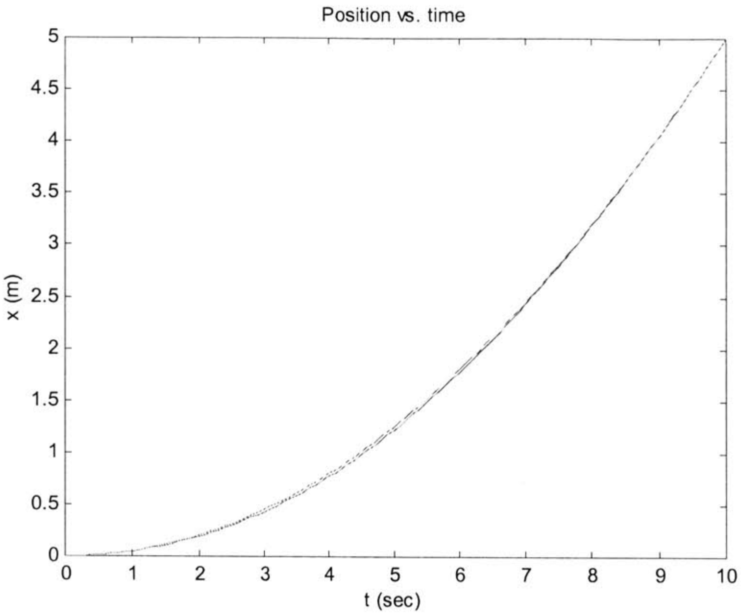


Figure 2.16: Desired and Actual Position vs. Time

2.3.5 Sensitivity to Variations in Sensor and Actuator Performance

The test trajectory chosen for the sensitivity study consists of a constant acceleration in the negative x direction (a V-bar approach toward the satellite) of 0.1 m/sec^2 for 10 seconds, starting from a position 50 m in the +x direction from the satellite. As the robot accelerates in the –x direction, the attitude control system is to hold its orientation constant. Final position and velocity are most important during rendezvous operations, so these end conditions were used as performance metrics during the study.

Initially, the system was tested with all bias and noise parameters in the sensor and actuator models set to zero, and then tested with all parameters set to their nominal values to approximate sensitivities to each input. The control system appeared to be most sensitive to variations in sensor performance, so the simulation was run for several different values of μ_r and σ_r, the shift and random noise in the range sensor. The dependence of final position and velocity on these values is shown in Figure 2.17 and Figure 2.18.

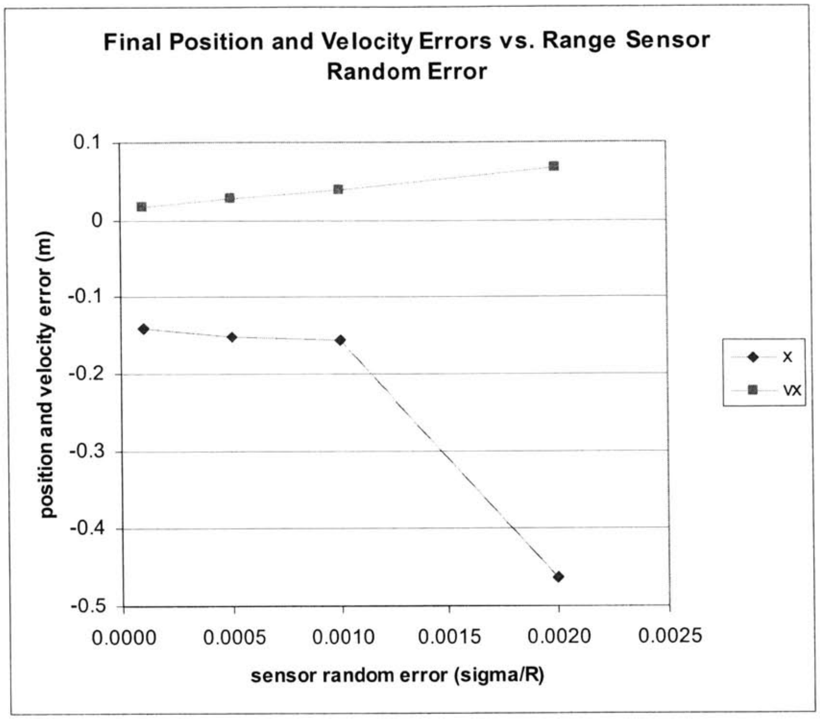


Figure 2.17: Sensitivity to RVR Random Error

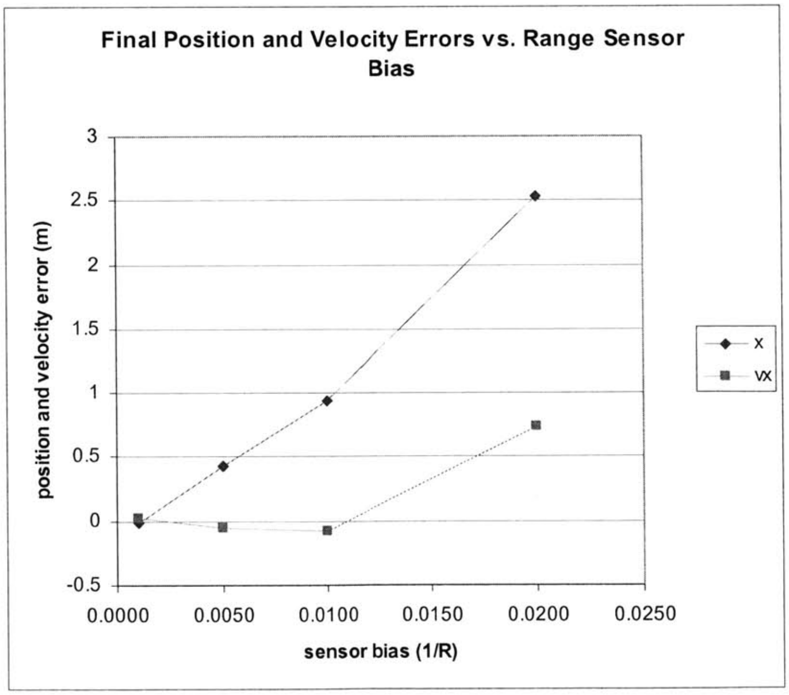


Figure 2.18: Sensitivity to Range Sensor Bias

The magnitude of the errors in position and velocity increase with the increasing errors in position estimates from the RVR system, and appear to increase non-linearly for higher values of μ_r and σ_r. Sensor errors are magnified by the control gains, causing saturation in the actuators and degrading the tracking capability of the free-flying robot control system. This sensitivity analysis shows that, although the free-flying control system works well for expected variations in sensor and actuator performance, larger variations could prevent the robot from accomplishing its mission.

2.4 Chapter Summary

This chapter presented an approach trajectory planning method based on optimization techniques and applied it to two representative cases. The method accounts for important considerations such as maximizing safety, mitigating plume impingement, and minimizing fuel usage by minimizing a cost function made up of performance metrics. A trajectory tracking control method was developed and simulated using a free-flying robot with realistic models of sensor and actuator performance to demonstrate that the planned trajectory was physically realizable.

CHAPTER 3 GRASP

3.1 The Grasp Phase

At the end of the approach trajectory, the space robot is close enough to the target satellite that the grasp point is within the workspace of an on board manipulator and the grasp phase begins. The objective of the grasp phase is to make contact with the satellite and establish a secure connection so that torques can be applied to the target to bring its motion under control. The robot reaches out with its manipulator, synchronizes the motion of its end effecter with the desired grasp point, and a grasping mechanism attaches securely to the satellite.

Satellites which are current candidates for servicing missions are not designed for cooperative capture by a free-flying robot, so they are not equipped with clear visual markers, relative motion sensors, or docking mechanisms designed to interface with a manipulator end effecter. The lack of cooperation from the target satellite presents several technical challenges and increases the demands on system performance for the space robot. A payload attachment fitting (PAF) or similar "hard point" capable of bearing a load is a likely target to be used as a grasp point.

The gripping mechanism used will have a significant influence on performance specifications for position and velocity control of the manipulator during grasp. The tolerance of the gripper to positioning error determines the required accuracy with which the end effecter must track the motion of the grasping point. An end effecter equipped with passive damping could help to absorb small velocity errors during contact.

During the grasp phase, the robot performs inertially referenced end-point motion control. Because relative motion with the target satellite defines the desired trajectory of the manipulator, the robot requires information on the position and orientation of the end effecter in inertial space in addition to feedback from joint sensors. Uncertainty in the inertial position and attitude data for the robot base will propagate through as errors in pose at the end effecter, so accurate sensor data on the relative motion between the robot and satellite are vital to successful control during the grasping phase.

This chapter starts by developing a dynamic model for a free-floating robot to aid in control design and simulation. Several different control methods are applied to the inertially referenced manipulator control for a simplified planar case, then some of the methods are applied to a more complex redundant spatial manipulator. The results of these methods are compared and their relative advantages are discussed.

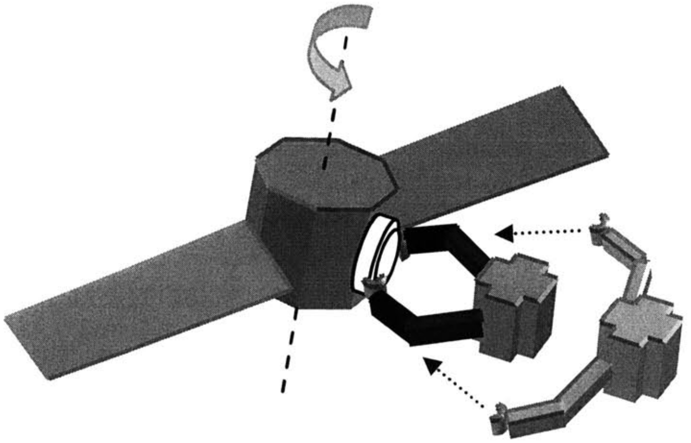


Figure 3.1: The Grasp Phase

3.2 Dynamic Model of a Free-floating Robot

The dynamic equations for a free-floating robot can be developed using Lagrangian methods. For a given robot with an n link manipulator, the kinetic energy of the system can be written as

$$K = \frac{1}{2}\sum_{i=0}^{n}\left[\underline{\omega}_i^{\ T} I_i \underline{\omega}_i + m_i \underline{v}_i^{\ T}\ \underline{v}_i\right] \tag{3.1}$$

The potential energy of the system is set to zero in the space environment. If the robot is assumed to be free-floating rather than flying, such that there are no external forces and zero initial momentum, then the center of mass of the system remains constant can be chosen as the origin of the inertial coordinate system.

Papadopoulos [1989] demonstrated the development of dynamic model of a free-floating space robot using energy methods. In the absence of potential energy, *K* is equal to *T* in the Lagrange equations

$$\frac{d}{dt}\left\{\frac{\partial T}{\partial \dot{q}}\right\}-\frac{\partial T}{\partial q}=\tau \tag{3.2}$$

This produces a set of *6+n* equations of motion for the system of the general form

$$H(q)\ddot{q}+C(q,\dot{q})\dot{q}=\tau \tag{3.3}$$

Because the base is free to translate and rotate, the system has *6+n* degrees of freedom. However, the condition of zero initial momentum and the dynamic coupling between the manipulator joint motions and base motion can be used to reduce the order of the system. Total system momentum is given by

$$\sum_{i=0}^{n} I_i\underline{\omega}_i+m_i\underline{v}_i=0 \tag{3.4}$$

The translational and rotational velocities of the links can be written as functions of the base and joint velocities, such that (3.4) can be re-written as

$$I_b\underline{\dot{x}}_b+I_m\underline{\dot{q}}=0 \tag{3.5}$$

where $\underline{\dot{q}}$ are the joint velocities. Solving for the base velocity

$$\underline{\dot{x}}_b=-I_b^{-1}I_m\underline{\dot{q}} \tag{3.6}$$

and substituting this expression into the original dynamic equations reduces the number of equations from *6+n* to *n*, producing the new set of equations

$$H^*(q)\ddot{q}+C^*(q,\dot{q})\dot{q}=\tau \tag{3.7}$$

where H* and C* are not dependent upon the position and angular velocity of the robot base. This method for finding the equations of motion is demonstrated for a free-floating robot with two revolute joints in Appendix C.

The end-effecter velocity, $\underline{\dot{x}}_m$, can be written as

$$\underline{\dot{x}}_m = \begin{bmatrix} \underline{v}_b + \underline{\omega}_b \times (\underline{r}_{b\to m}) \\ \underline{\omega}_b \end{bmatrix} + J_m \underline{\dot{q}} = J_b \underline{\dot{x}}_b + J_m \underline{\dot{q}} \tag{3.8}$$

Substituting the relation between $\underline{\dot{x}}_b$ and $\underline{\dot{q}}$ once more gives

$$\underline{\dot{x}}_m = (-J_b I_b^{-1} I_m + J_m)\underline{\dot{q}} = J^* \underline{\dot{q}} \tag{3.9}$$

This new manipulator Jacobian, J*, is called the generalized Jacobian, and it is a function of the inertial parameters of the system as well as the manipulator kinematics [Yoshida and Umetani, 1989]. It can be seen that as the inertia of the robot base I_b increases relative to I_m, J* → J_m. The generalized Jacobian accounts for the motion of the base in reaction to manipulator movements, and is, therefore, the appropriate Jacobian to use when calculating control torques.

A comparison between the standard fixed-base Jacobian and generalized Jacobian for an example two-link manipulator is shown in Figure 3.2, plotted over a range of several different values for the mass of the robot base. With the mass of the manipulator held constant, the mass of the base is increased and the ratio of the two is given on the x-axis of the plot. As the ratio increases, the values of the elements of the generalized Jacobian approach those of the fixed-base Jacobian, and the physical performance of the free-floating robot becomes nearly identical to that of the fixed-base system. The Jacobians are also dependent upon the system configuration, but it is clear that there is significant difference between the values when the mass of the base is less than five times the mass of the manipulator. For a smaller agile space robot supported by a mother ship, the manipulator is likely to be as much as half of the mass of the base. Therefore, dynamic properties of the robot base must be taken into account for precise manipulator control. Based on the work of Papadopoulos [1990], the matrices H*, C*, and J* can be used to implement fixed base control algorithms for free-floating robots.

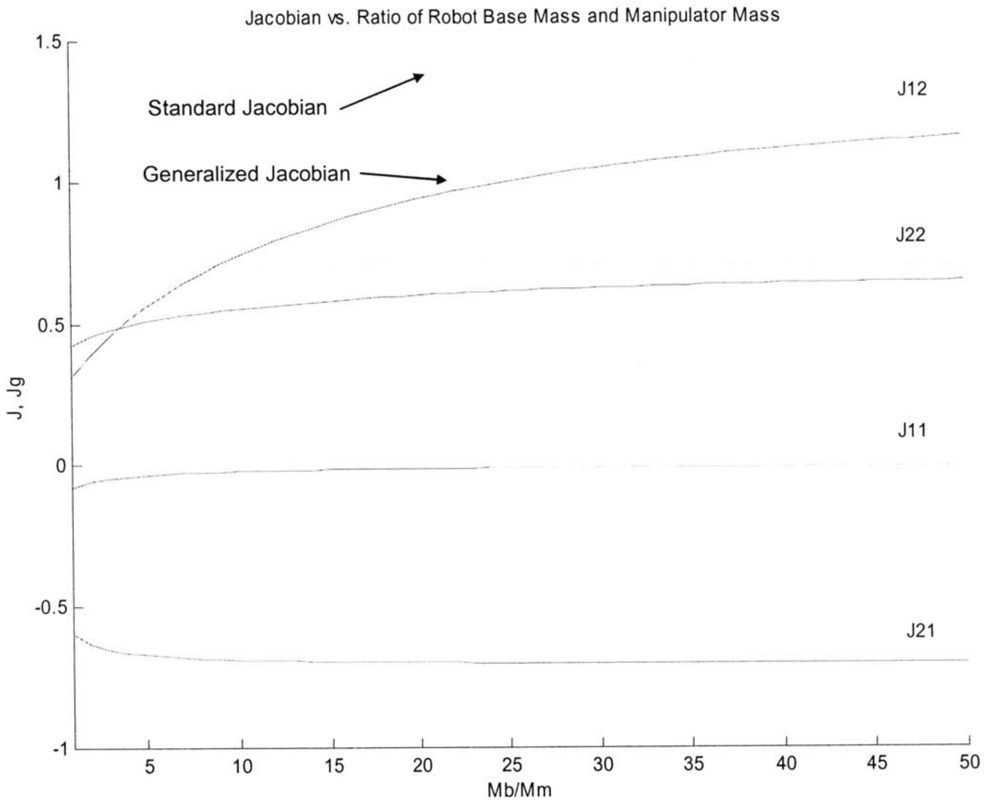


Figure 3.2: Jacobian vs. Ratio of Base Mass to Manipulator Mass

3.3 Control before grasp

3.3.1 Jacobian Inverse Control

To establish a baseline case for the evaluation of inertially referenced endpoint control methods, a PD controller was applied to the grasp task. The control law for a basic PD controller using the standard fixed-base manipulator Jacobian is given by

$$\boldsymbol{\tau} = \mathbf{J}^{-1}\left[\mathbf{K}_p\left(\mathbf{x}_d - \mathbf{x}\right) + \mathbf{K}_d\left(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}\right)\right] \tag{3.10}$$

For a manipulator with redundant links, the Jacobian is not symmetric, and the inverse $\mathbf{J}^{-1}$ in the above control laws is replaced with a pseudoinverse $\mathbf{J}^{+}$.

3.3.2 Generalized Jacobian Inverse Control

To account for base reactions to manipulator dynamics, the fixed-base Jacobian in the above control law is replaced with the generalized Jacobian.

$$\boldsymbol{\tau} = \mathbf{J}_G^{-1}\left[\mathbf{K}_p\left(\mathbf{x}_d - \mathbf{x}\right) + \mathbf{K}_d\left(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}\right)\right] \tag{3.11}$$

When the robot's manipulator is considerably less massive than the robot base itself, the manipulator motion approximates that of a fixed-base manipulator. However, if the mass of the manipulator is significant relative to the base, which is more likely in the case of a smaller space robot, the dynamic coupling between the base motion and manipulator motion cannot be ignored.

For this reason, accurate knowledge of a space robot's dynamic parameters is especially important for precision trajectory control. These parameters may be time varying, due to the consumption of attitude control fuel, for example, or not precisely known if the robot is manipulating a payload with unknown inertial parameters. The need for precise trajectory control and the uncertainty about system dynamic parameters combined with the inherent nonlinearity of manipulator dynamics indicates that nonlinear control methods, such as robust or adaptive control, may be appropriate ways to perform trajectory control and estimate unknown parameters.

3.3.3 Adaptive Trajectory Control

An adaptive controller can be used to estimate unknown or uncertain parameters in a system, and use the parameters to determine the control inputs for the system. One type of adaptive controller, demonstrated by Slotine and Li [1991], calculates the control inputs required to produce the desired motion using a model of the system containing parameter estimates. An adaptation law is defined such that the parameters converge to values that provide good trajectory tracking performance when used in the system model.

Once the equations of motion of the system are in the standard form, a matrix **Y** can be defined such that

$$H\ddot{q}_r + C\dot{q}_r = Ya \tag{3.12}$$

where

$$\dot{q}_r = \dot{q}_d - \Lambda\tilde{q} \tag{3.13}$$

$$\tilde{q} = q - q_d \tag{3.14}$$

q_d is the desired position in joint space and **a** is a vector of dynamic parameters which are to be estimated. This can be factored out of the equations of motion because **H** and **C** depend linearly on the mass properties of the manipulator. The control law is then

$$\tau = Y\hat{a} - K_D s \tag{3.15}$$

containing a feedforward term based on the estimate of **a** and a PD term **K**$_\mathbf{D}$**s**, where **s** is the velocity error term defined by

$$s = \dot{\tilde{q}} + \Lambda\tilde{q} \tag{3.16}$$

The parameter estimates are updated based on **Y** and the error term, **s**

$$\dot{\hat{a}} = -\Gamma Y^T s \tag{3.17}$$

where Γ is a symmetric positive definite matrix. The parameter estimates are updated until the trajectory error term **s** goes to zero. The resulting parameter estimates are sufficiently accurate to allow the manipulator to track the desired trajectory. However, if the commanded trajectory is simple, the parameter estimates may not be close to the actual inertial parameters even though trajectory tracking errors are driven to zero.

3.3.4 Robust Trajectory Control

If there are unmodeled dynamics or if there are uncertain or varying parameters in the space robot system, a robust controller is a method that can be used to ensure that the robot is able to track the desired trajectory in spite of inaccuracy in the system model. A sliding condition can be defined that guarantees the system will converge to the desired trajectory if the controller satisfies the sliding condition. The controller is, thus, robust to uncertainty and variation in the system model.

A Lyapunov function, an "energy" type of positive definite scalar function for the system [Slotine and Li, 1991], is chosen as

$$V(t) = \frac{1}{2}[s^T H s] \tag{3.18}$$

Then $\dot{V}$ becomes

$$\dot{V}(t) = s^T(H\ddot{q} - H\ddot{q}_r) + \frac{1}{2}s^T\dot{H}s \tag{3.19}$$

Replacing $H\ddot{q}$ with the expression for the system dynamics produces

$$\dot{V}(t) = s^T(\tau - H\ddot{q}_r - C\dot{q}_r) \tag{3.20}$$

The best estimate of τ that would make $H\dot{s}$ go to zero is

$$\hat{\tau} = \hat{H}\ddot{q}_r + \hat{C}\dot{q}_r \tag{3.21}$$

so the control input is chosen as

$$\tau = \hat{\tau} - k\,\mathrm{sgn}(s) \tag{3.22}$$

The sliding condition for the multiple actuators in the manipulator is given by

$$\dot{V}(t) = -\sum_{i=1}^{2} \eta_i |s_i| \tag{3.23}$$

which is satisfied if k in the control law is chosen such that

$$k_i \geq \left|\tilde{H}\ddot{q}_r + \tilde{C}\dot{q}_r\right| + \eta_i \tag{3.24}$$

where $\tilde{H}$ and $\tilde{C}$ are the maximum errors in model parameter estimates. To eliminate excessive switching of the control input, the feedback term in the control law can be smoothed out by replacing *k sgn(s)* with *k sat(s/ϕ)*, where ϕ is the thickness of the boundary layer inside which the feedback term is interpolated.

3.3.5 Simulation Results

Planar Manipulator

The control methods were initially tested on the free-floating robot with a simplified planar manipulator with friction in the joints, shown in Figure 3.3. Because of the kinematics of a spatial manipulator with multiple revolute joints and the dynamics coupling joint motions to base motions, H* and C* quickly become large even for a simple manipulator. When designing an adaptive controller, it is necessary to separate out the inertial parameters upon which H* and C* have a linear dependence from the equations of motion of the system. To simplify the derivations of the dynamic equations and control laws, a reduced order planar model was chosen. Using a two link manipulator with prismatic joints greatly simplifies the equations of motion.

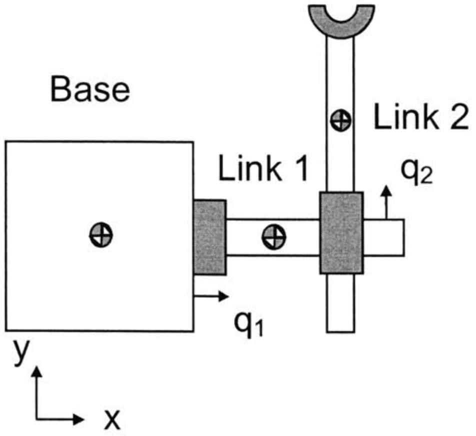


Figure 3.3: Model of Free-floating Space Robot

The test trajectory used in the simulations was a circle in the x-y plane with a radius of 0.25 m traced at 1Hz, so the commanded trajectories in x and y are sinusoidal. The resulting trajectory errors for dynamic simulations with the generalized Jacobian, adaptive, and robust controller are shown in Figure 3.4, Figure 3.5, and Figure 3.6. The parameter estimates produced by the adaptive controller are plotted vs. time in Figure 3.7. The test trajectory is sufficiently exciting that the parameter estimates converge to their actual values with time. The actual values for the parameters $\mathbf{a}$ are

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} 0.1\left(\dfrac{-(m_1+m_2)^2}{m_0+m_1+m_2}+m_1+m_2\right) \\ 0.1\left(\dfrac{-m_2^2}{m_0+m_1+m_2}+m_2\right) \\ \gamma_1 \\ \gamma_2 \end{bmatrix} = \begin{bmatrix} 0.667 \\ 0.417 \\ 0.1 \\ 0.1 \end{bmatrix} \tag{3.25}$$

where m_i is the mass of link i and γ_i is the coefficient of friction in joint i.

Table 3: Robot Model Parameters

Element	Mass (kg)	Length (m)
Base	20	1
Link 1	5	1 (initially)
Link 2	5	1 (initially)

Table 4: Control System Simulation Results

	Absolute Final Position Error (m)	Absolute Final Velocity Error (m/s)
Generalized Jacobian Control	0.022	0.070
Adaptive Control	0.0092	0.018
Robust Control	0.0021	0.0076

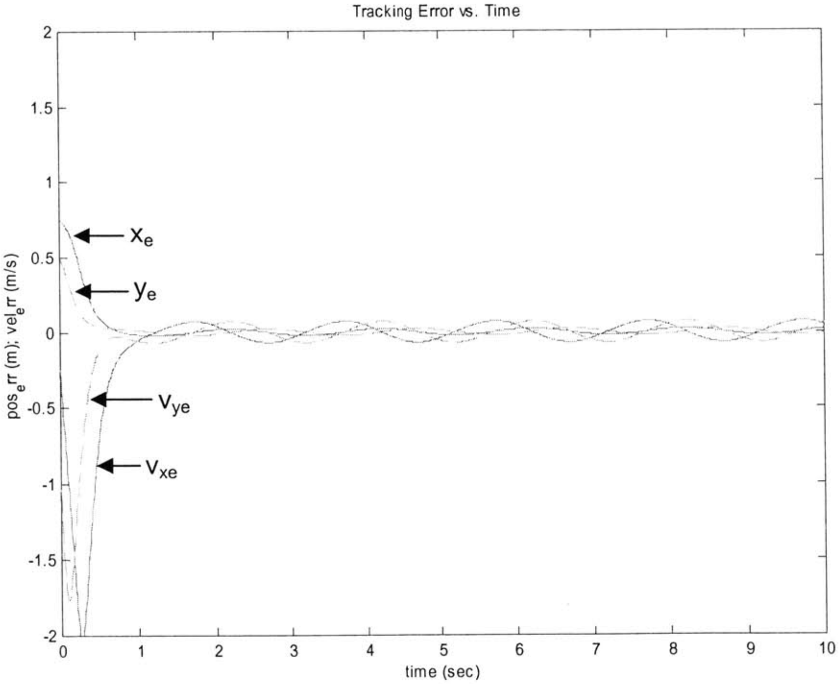


Figure 3.4: Trajectory Errors, Generalized Jacobian Control

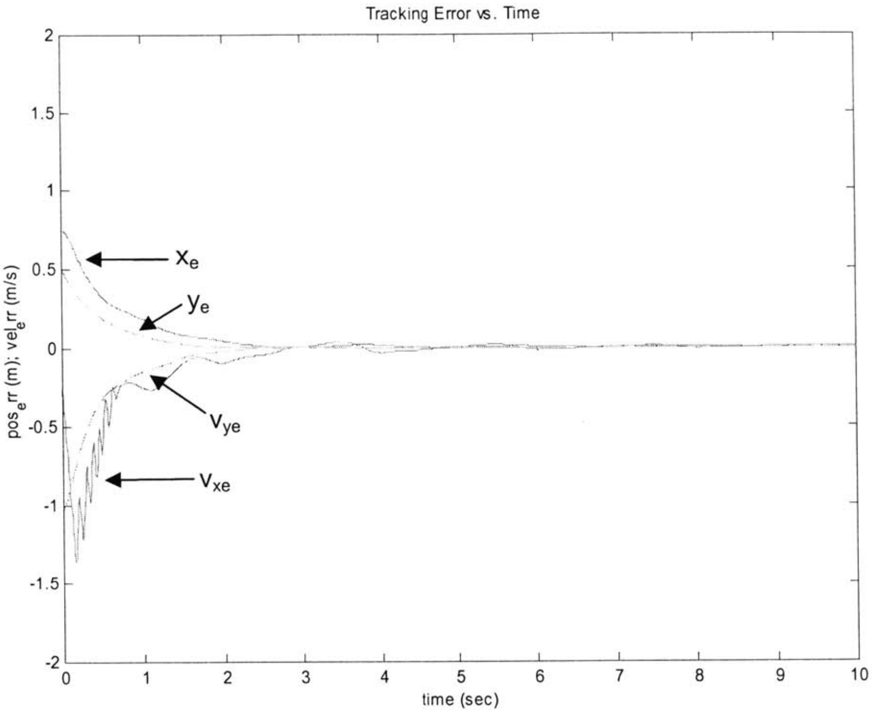


Figure 3.5: Trajectory Errors, Adaptive Control

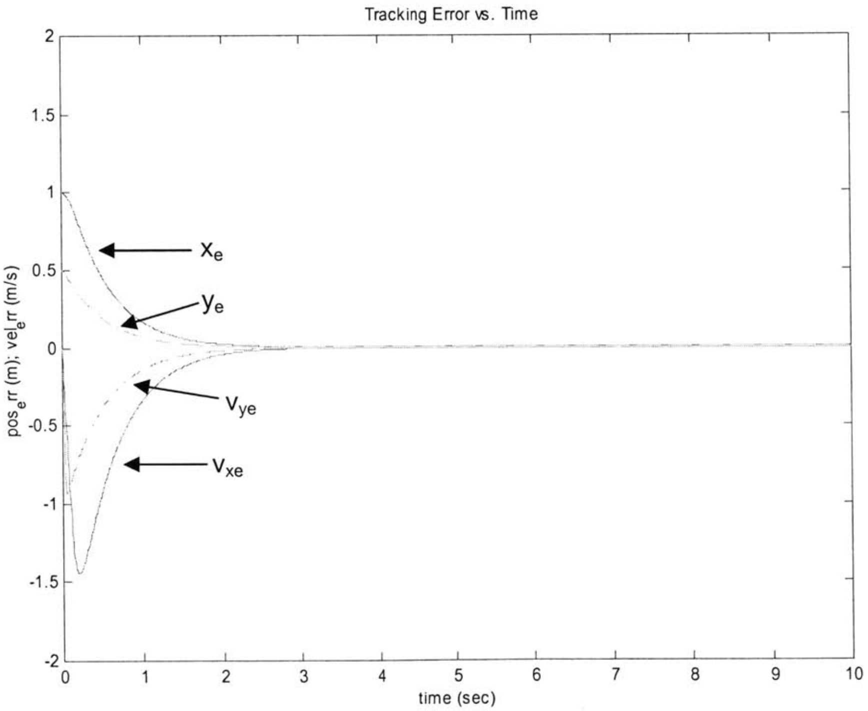


Figure 3.6: Trajectory Errors, Robust Control

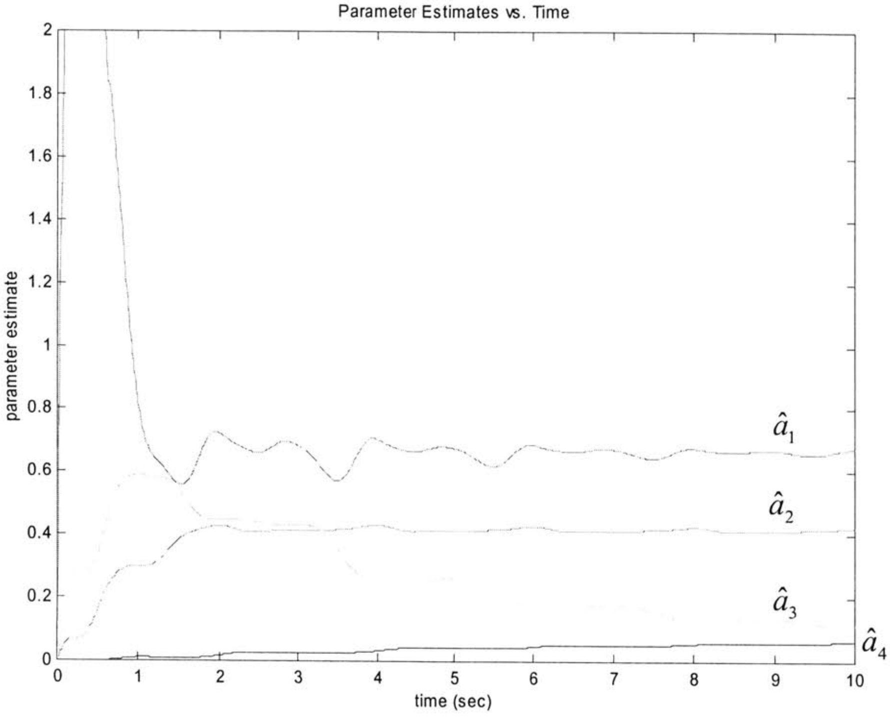


Figure 3.7: Convergence of Parameter Estimates for Adaptive Controller

Spatial Manipulator

To test the manipulator control methods for a more realistic complex system, the spatial manipulator model shown in Figure 3.8 was developed. The manipulator is mounted on a free floating base and has seven rotational degrees of freedom, allowing for control of position and orientation of the end-effecter with redundancy. Due to the complex nature of the system dynamics, the equations of motion were not developed symbolically, so only the standard Jacobian inverse and generalized Jacobian inverse control methods were tested for the spatial manipulator.

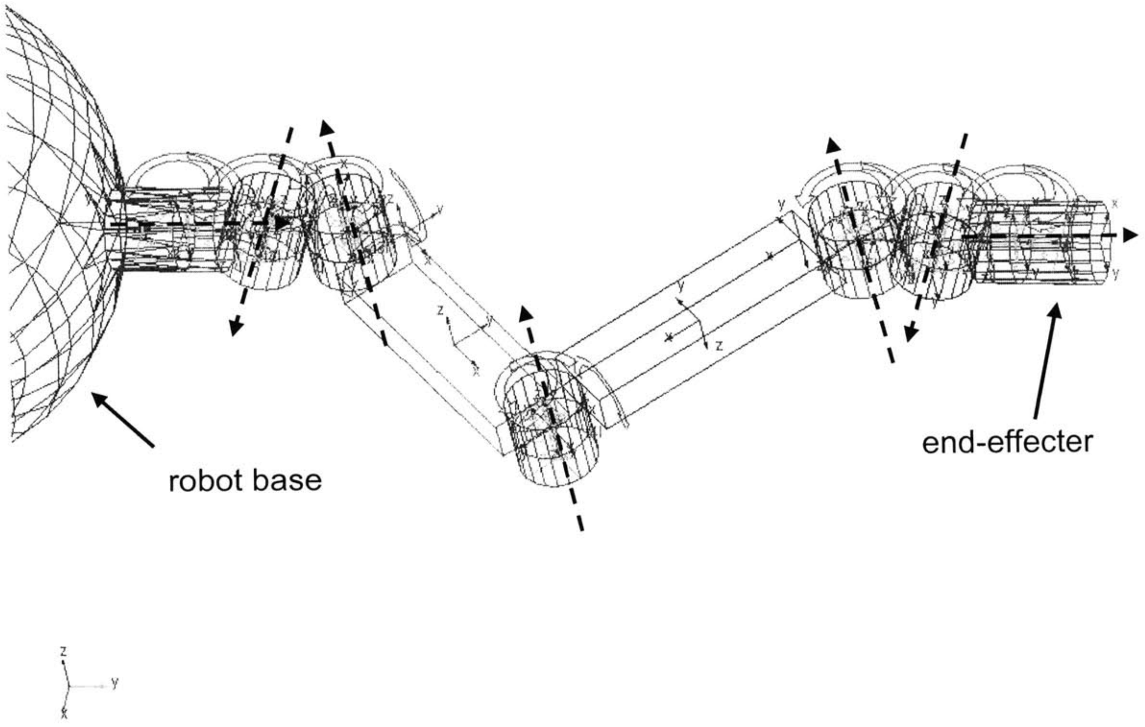


Figure 3.8: Seven Degree of Freedom Manipulator Arm

The spatial manipulator was tested for its ability to track the position and orientation of a grasp point using the different control methods. A screen capture from one of the simulations is shown in Figure 3.9. Due to the relatively short distances and brief duration of the grasp task, the effects of orbital mechanics are assumed to be negligible during the task and are not included in the simulation. The grasp point on the satellite is represented by the point highlighted on a disc rotating at the expected angular velocity of the target satellite. The robot is given an initial linear velocity that matches the tangential velocity of the grasp point when the distance between the grasp point and robot base are at a minimum. The end-effecter has an initial error in position and velocity relative to the grasp point. As the disc rotates and the robot translates, the manipulator synchronizes its position and velocity with the grasp point in both location and orientation until the grasp point leaves the manipulator's workspace.

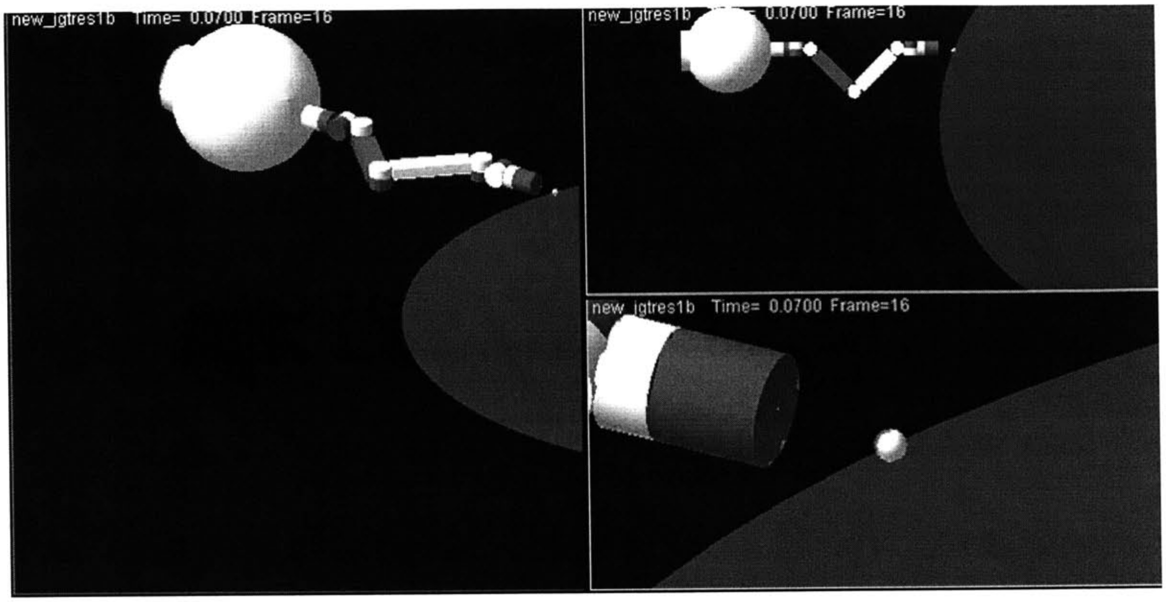
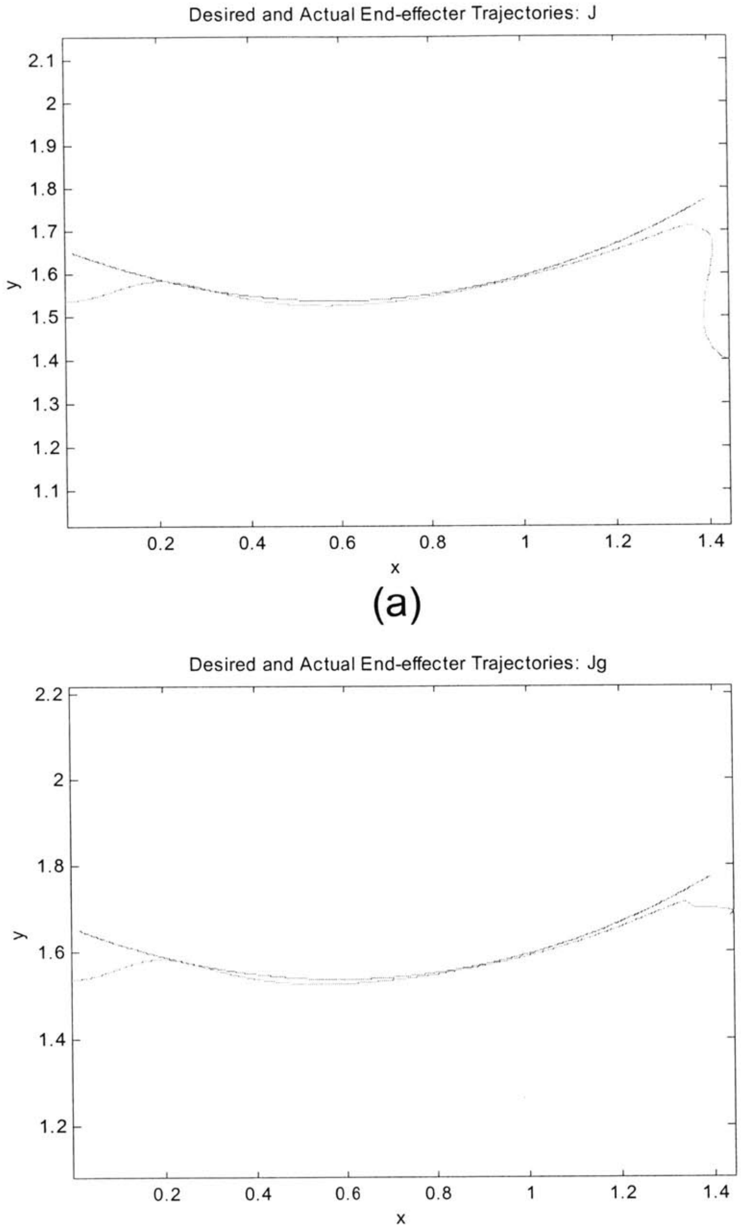

new_jgtres1b Time= 0.0700 Frame=16
new_jgtres1b Time= 0.0700 Frame=16
new_jgtres1b Time= 0.0700 Frame=16

Figure 3.9: Simulation of Spatial Manipulator Tracking Grasp Point

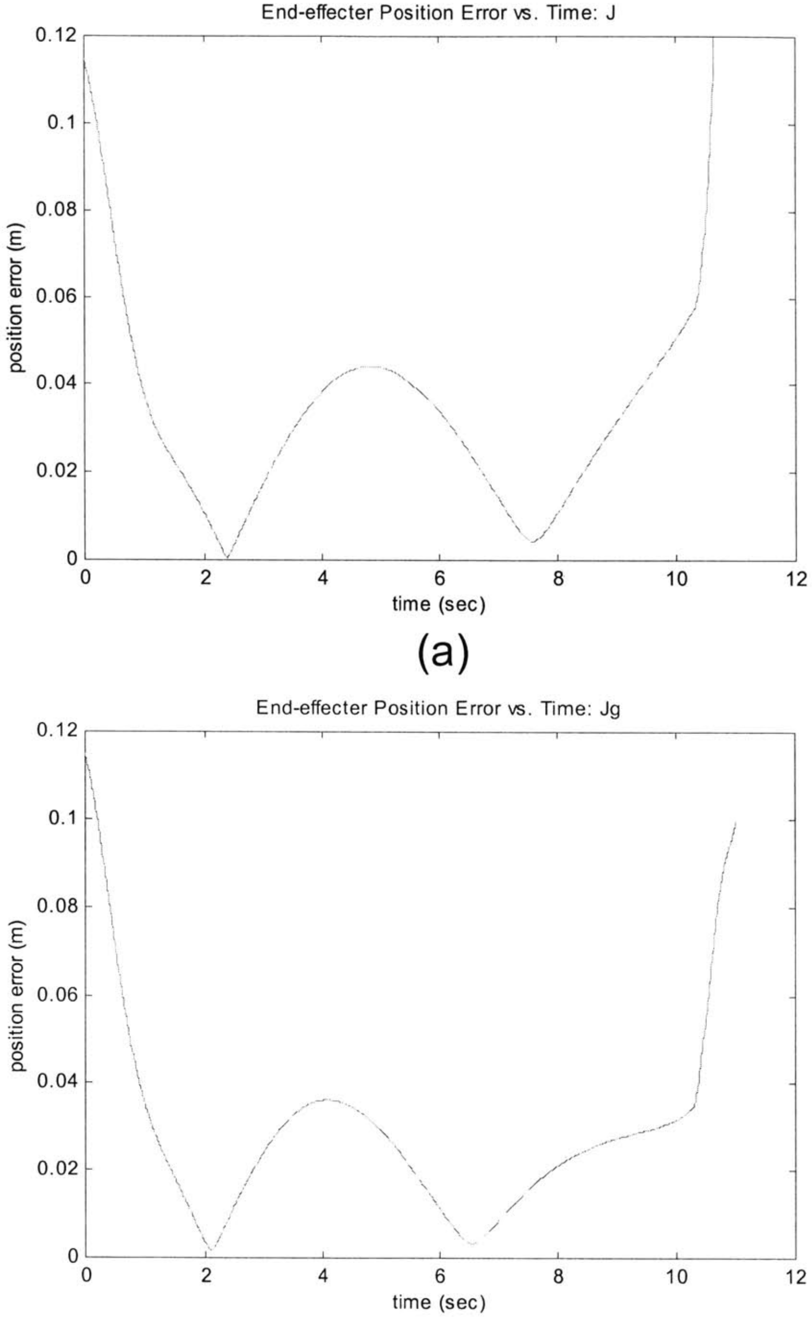
The resulting trajectories in the x-y plane are shown in Figure 3.10 with the end-effecter path is super-imposed on the grasp point trajectory. After the initially large position error, the trajectories converge and the tracking error is very small until the desired position is out of the reach of the manipulator and the trajectories diverge. Both control methods result in position errors of less than 10 cm when the grasp point is within the workspace of the manipulator. However, if the error tolerance of the grasping mechanism is less than 10 cm, these errors may be too large. If the grasping tolerance is 4 cm, for example, the generalized Jacobian controller stays within the acceptable error range for 3 seconds longer than the standard Jacobian controller. Also, in these simulations, the initial velocity conditions of the manipulator and grasping point were relatively well matched, with velocity errors of a few centimeters per second. Larger velocity errors would place a higher demand on the trajectory tracking controllers, and the difference in performance may become more significant.



(a)

(b)

Figure 3.10: End-effecter Trajectories and Trajectory Errors: standard Jacobian (a) and generalized Jacobian (b)



(a)

(b)

Figure 3.11: End-effecter Position Errors vs. Time: standard Jacobian (a) and generalized Jacobian (b)

3.4 Chapter Summary

This chapter evaluated several different methods of inertially referenced manipulator control for the purpose of manipulator during grasp, including generalized Jacobian inverse control, adaptive control, and robust control. The methods were first applied to a simplified planar case, then the generalized Jacobian method was applied to a more complex spatial manipulator. The results of these methods demonstrated that modeling of the dynamic interactions between the base and manipulator of a free-flying space robot can increase the accuracy of endpoint control.

CHAPTER 4 STABILIZATION

4.1 The Stabilization Phase

Once a secure grasp has been made, relative motion between the robot and satellite should be eliminated and the motion of the entire system stabilized. [*] make distinction between relative motion damping and stabilization of system as a whole The objective of the stabilization phase is to reduce excessive spinning or tumbling in the target satellite so that servicing or transporting can be carried out. This motion stabilization can be achieved through a combination of active control techniques and passive energy dissipation mechanisms.

For a satellite servicing mission, it is assumed that the mass properties of the target satellite will be reasonably well known beforehand and that this information can be used to estimate the forces and torques necessary to stabilize the system. However, uncertainty in estimation of system motion or changes in the system mass properties could destabilize the system, so control techniques for stabilization should be robust to these uncertainties.

The use of multiple robots may present advantages during the stabilization phase. Multiple robots connected to various hard points in a symmetric fashion around a spinning satellite could more easily apply a pure torque about the spin axis of satellite and stabilize the system without significantly disturbing its translational motion. If multiple robots are used, their actuators should be well coordinated and the coupling effects of their motion should be accounted for.

This chapter covers the relative motion damping portion of the stabilization phase. The objectives and constraints for relative motion damping are laid out, and several methods for the control of relative motion are presented. The different methods are applied in stabilization simulations and their effectiveness with respect to the objectives and constraints is analyzed. Finally, the implications that these results have for improving grasp planning are presented.

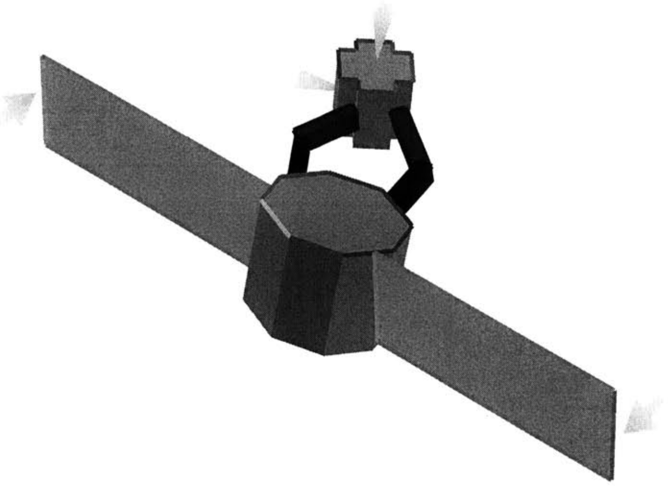


Figure 4.1: The Stabilization Phase

4.2 Control after Grasp

4.2.1 Objectives

There are three main tasks to complete during the stabilization phase, and these correspond to the objectives for control after grasp. The tasks may be performed simultaneously or sequentially, depending on the dynamic state of the target satellite and the capabilities of the space robot control system.

1. Eliminate relative motion between the robot and satellite: $\dot{\mathbf{q}} \rightarrow 0$ If the links of the manipulator connecting the robot and satellite are sufficiently rigid, the first objective consists of essentially driving the joint velocities to zero.

2. Reconfigure the robot and satellite system: $\mathbf{q} \to \mathbf{q}^*$ It is assumed that not all manipulator configurations are equally well suited for stabilizing the system as a whole. The optimal configuration may be one that minimizes steady state joint forces due to centripetal acceleration, minimizes plume impingement, or allows the robot to apply a maximum moment about the system center of mass during the third stabilization task.
3. Eliminate the spinning or tumbling motion of the system: $\boldsymbol{\omega} \to 0$ When the system is in its optimal configuration for stabilization, the robot fires its attitude control thrusters, uses its reaction wheels, or employs another momentum transferring device to reduce the overall motion of the system.

4.2.2 Constraints

The space robot's control systems must operate within a number of physical constraints while performing the stabilization tasks. The constraints can be grouped into the following general categories.

1. Actuator saturation limits: $|\boldsymbol{\tau}| \le \boldsymbol{\tau}_{\max}$ The maximum torques that can be produced by the joint motors in the robot manipulator will limit the rate at which relative motion between the robot and satellite can be eliminated. The joint torque limits should be at least high enough that the robot can overcome centripetal accelerations and reconfigure itself while spinning together with the target satellite after capture.
2. Maximum interactive forces: $|\mathbf{F}| \le \mathbf{F}_{\max}$ The maximum forces and torques that the space robot can safely apply may be limited by factors other than actuator saturation. The grasping mechanism at the end-effecter may not be able to maintain a rigid grasp under large stresses. High interactive forces could cause structural damage to the space robot's manipulator or delicate appendages of the target satellite.
3. Workspace and joint limits: $|\mathbf{q}| \le \mathbf{q}_{\max}$ and $|\mathbf{p}| \le \mathbf{p}_{\max}$ To prevent damage to manipulator joints, the motion of the robot after grasp should avoid joint limits and manipulator singularities at workspace limits. Also, the range of motion of the

manipulator will be limited by the need to avoid collision between the two bodies after grasp.

Once the robot is connected to the satellite, the system can be viewed as a single manipulator chain. The satellite is much more massive than the robot and can, thus, be treated as the base of the manipulator chain. The robot base can be treated as the end effecter of the chain, and the workspace of the manipulator chain is defined in a coordinate frame moving with the satellite.

The workspace for relative motion stabilization can be subdivided into the reachable and dexterous workspaces. The reachable workspace represents the set of possible end-effecter positions, while the dexterous workspace is the subset of points which the end-effecter can reach with any orientation. Boundaries of the reachable workspace are clearly restrictions which can cause high forces to develop in the manipulator joints. High interaction forces can also occur at the transition from the dexterous to reachable workspaces, where the manipulator links may be subject to large rotational accelerations to satisfy the constraints on end-effecter orientation.

The manipulator end-effecter on the space robot should be rigidly connected to the target grasp point before relative motion stabilization begins. The grasping mechanism may take a number of seconds to secure the grasp and cannot begin before the grasp point is within the workspace of the manipulator. Therefore, the workspace available for relative motion stabilization will be reduced due to the motion of the robot while the grasp is being secured. This reduced workspace represents the set of possible initial positions of the robot base relative to the satellite at the start of the relative motion damping process.

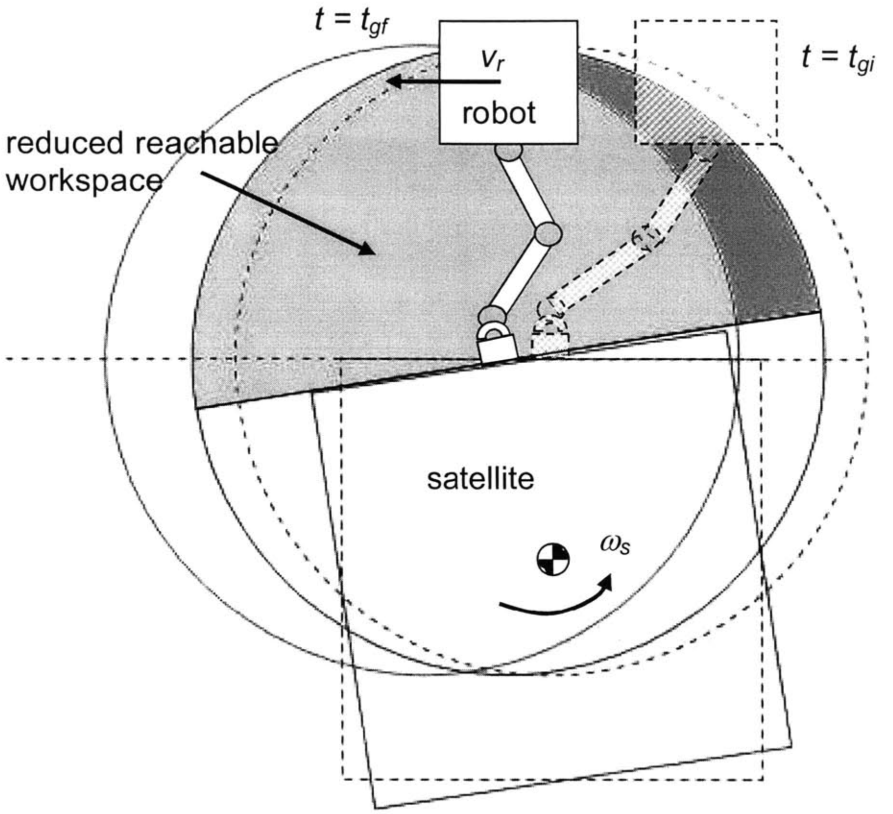


Figure 4.2: Reduced Reachable Workspace of Manipulator

4.2.3 Joint Control

The first controller tested was a simple PD controller applied to position and velocity in joint space.

$$\boldsymbol{\tau} = \mathbf{K}_D\left(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}\right) + \mathbf{K}_P\left(\mathbf{q}_d - \mathbf{q}\right) \tag{4.1}$$

The desired joint velocities are set to zero and the desired joint positions are chosen as the initial configuration immediately after grasp. To reduce the impulse due to a sudden increase in joint torques after grasp, the desired velocities are initially equal to the actual joint velocities at grasp time and reduced to zero exponentially. The time constant for exponential decay is sufficiently small so that the manipulator does not drift significantly toward its workspace limits before the joint motion is eliminated.

The properties of the space robot and satellite used for control simulations are given in Table 5 and shown in Figure 4.3. For the representative case used to test the control method, the initial conditions were ω_s = 5 deg/sec, q_1 = 57 deg, q_2 = -114 deg, and

$v_r = 0.5$ m/sec. The proportional and derivative gains K_p and K_d were set to 100. Figure 4.4 shows the results of simulation using joint control to eliminate relative motion. The joint controller achieves the control objectives reasonably well, with a settling time of 8 seconds, a maximum control torque of 28 Nm, and a maximum joint force of 37 N.

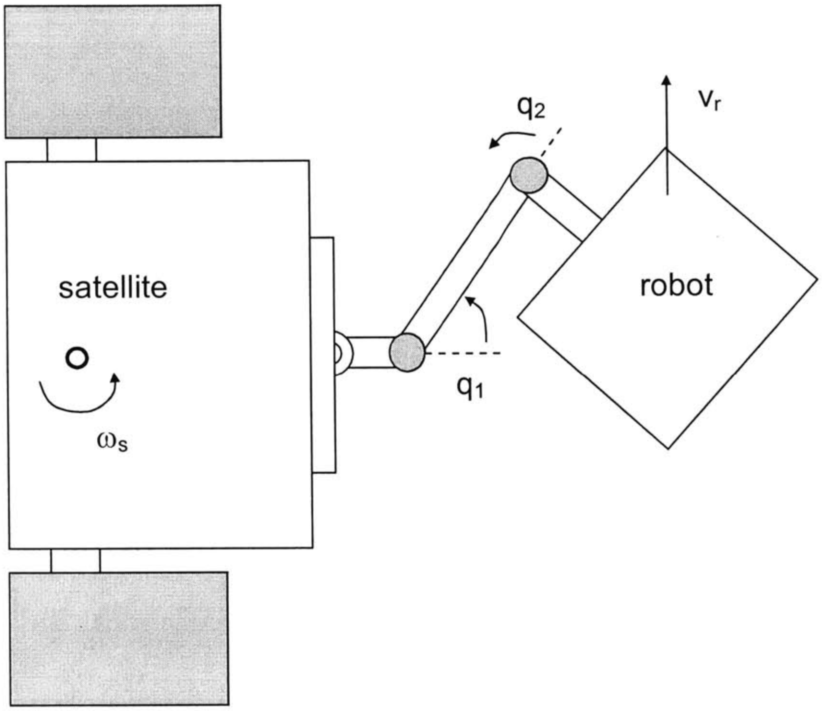


Figure 4.3: Initial Configuration of Satellite and Robot System

Table 5: Properties of Robot and Satellite

	Mass (kg)	**Length (m)**	**Width (m)**
Satellite	2400	2	2
Robot Base	100	1	1
Link 1	4	1	--
Link 2	end-effecter treated as part of satellite		

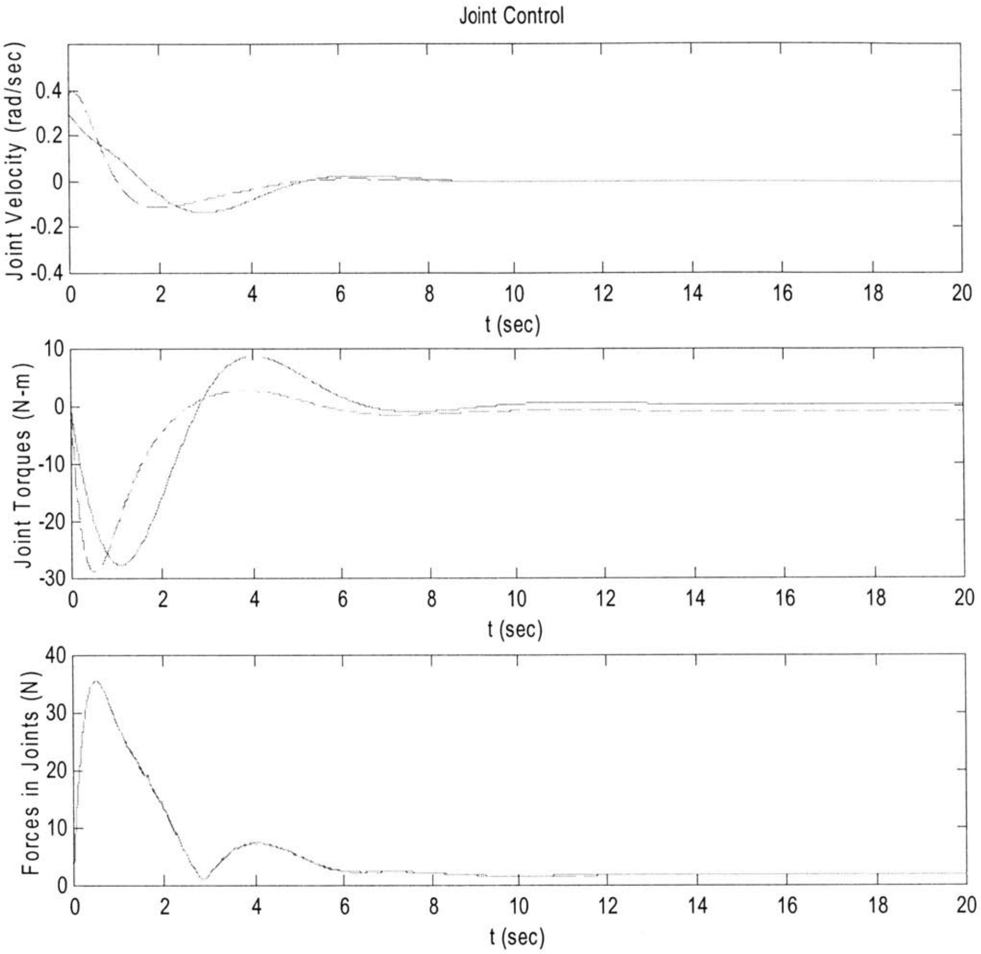


Figure 4.4: Joint Control Simulation Results

4.2.4 Control in Relative Motion Frame

If the mass of the robot base is significantly greater than the manipulator, it may be more effective to determine the control torques based on the motion of the robot base with respect to the satellite rather than simply the joint motions. This controller applies a force to the robot base that directly opposes its motion in a frame attached to the satellite center of mass. For the given system parameters, the resulting maximum forces in the joints are somewhat lower than for control in joint space.

$$\boldsymbol{\tau} = \mathbf{J}^T \left(\mathbf{K}_D \left(\dot{\mathbf{x}}_d - \mathbf{J}\dot{\mathbf{q}} \right) + \mathbf{K}_P \left(\mathbf{x}_d - \mathbf{x} \right) \right) \tag{4.2}$$

Similar to the case with joint control, the desired velocity is set to zero and the desired position is chosen as the initial position immediately after grasp. Figure 4.5

shows simulation results for control in the relative motion frame. This controller shows improvements compared to the joint controller with regards to maximum control torque and joint forces, but requires a slightly longer period of time to eliminate relative motion.

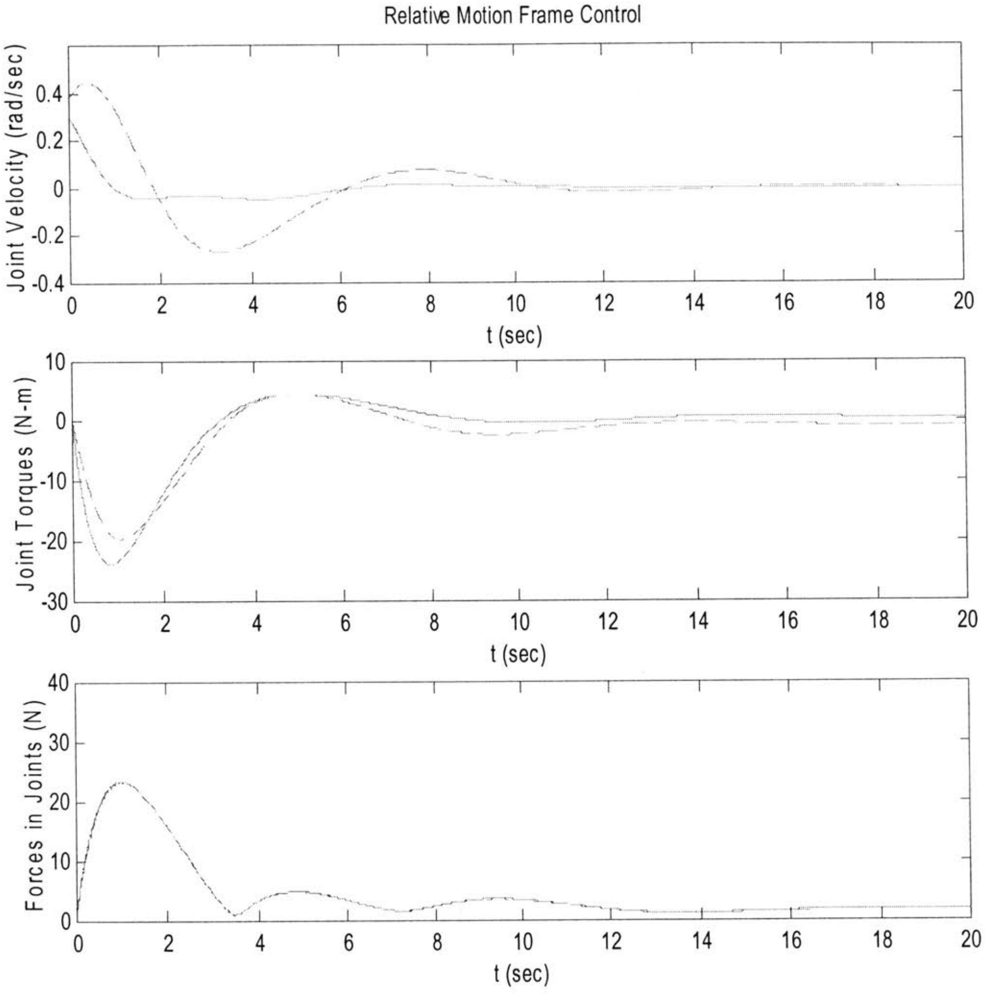


Figure 4.5: Control in Relative Motion Frame

4.2.5 Control Based on Manipulability Metric

If the manipulator connecting the space robot to the satellite reaches a singular configuration, the resulting impulse can cause a spike in joint and end-effecter forces that may damage the robot or satellite. When the manipulator is in a singular configuration, the value of the determinant of its Jacobian goes to zero.

$$\det\left(\mathbf{J}(\mathbf{q})\mathbf{J}^{T}(\mathbf{q})\right)=0 \tag{4.3}$$

Therefore, the value of this determinant can be used to measure the suitability of a given configuration to permit motion in an arbitrary direction. A measure of manipulability proposed by [Asada, 1985] is given by

$$w = \sqrt{\det\left(\mathbf{J}\mathbf{J}^T\right)} \tag{4.4}$$

For a manipulator without any redundant links, **J** is symmetric and the expression for manipulability simplifies to

$$w = \left|\det\left(\mathbf{J}\right)\right| \tag{4.5}$$

The manipulability index in inertial space for the two-link manipulator shown in Figure 4.3 and described in Table 3 is plotted over its workspace in the x-y plane in Figure 4.6.

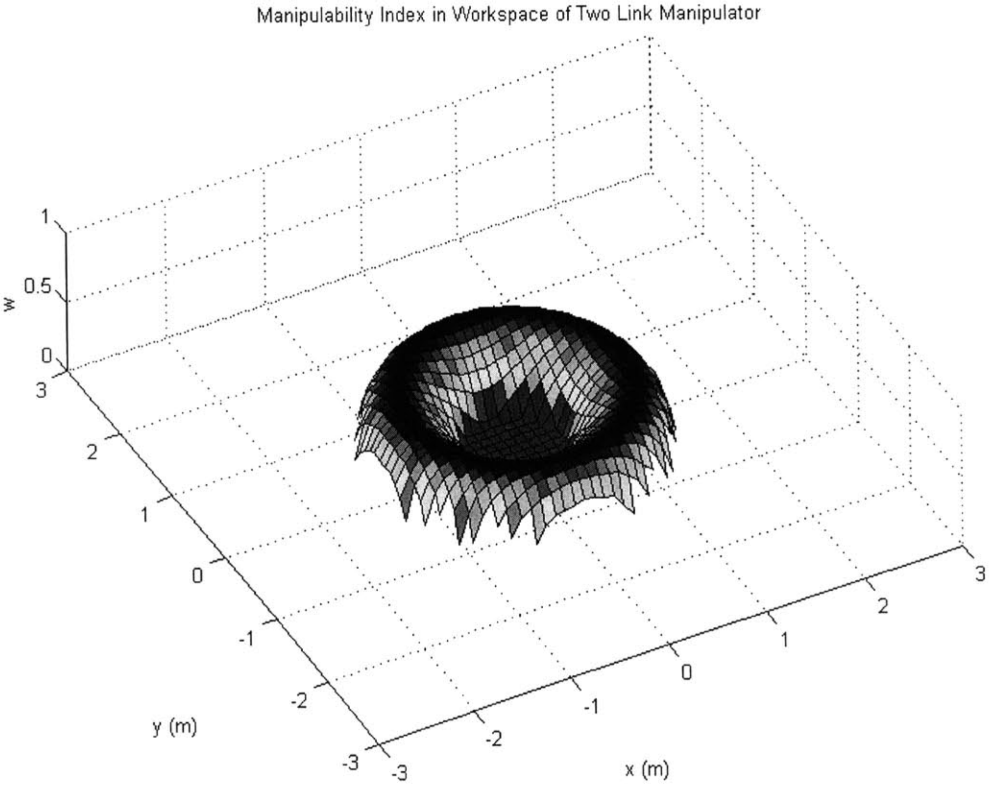


Figure 4.6: Manipulability Index in Workspace of Two Link Manipulator

The expression for a manipulability ellipse [Craig, 1989] can be written as

$$\mathbf{x}^T\mathbf{J}\mathbf{J}^T\mathbf{x} = 0 \tag{4.6}$$

This equation gives an ellipse that shows the ability of the manipulator end-effecter to move in each direction from its current position. The orientation of the major axis of the ellipse represents the direction in which the end-effecter has the greatest freedom of movement, and the minor axis represents the direction in which movement is most restricted. Manipulability ellipses at several configurations of the same two link manipulator are plotted in Figure 4.7.

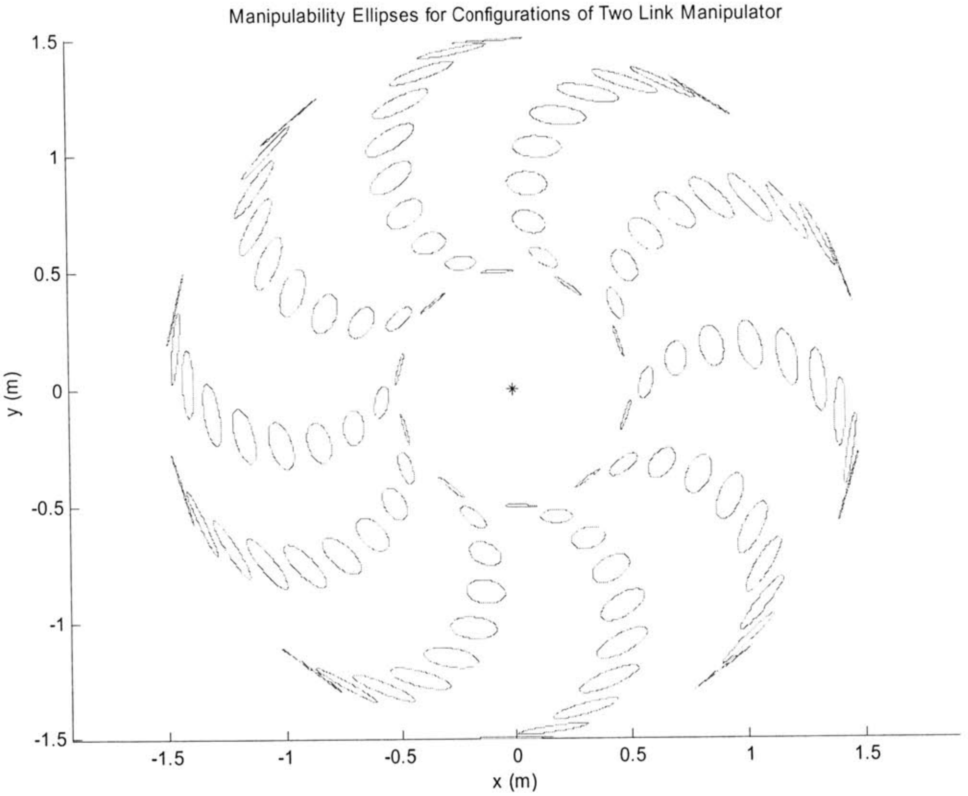


Figure 4.7: Manipulability Ellipses for Two Link Manipulator: RH Configurations

A control method that seeks to maximize the manipulability of a manipulator could be used to avoid singularities and prevent dangerously high interaction forces. Calculating the gradient of the manipulability metric described previously and incorporating the gradient into the manipulator control law is one possible method of maximizing manipulability during relative motion stabilization.

The manipulability gradient is simply the gradient of the manipulability function calculated over its inertial or joint parameter space, given by

$$\overset{\mathrm{r}}{\nabla} w(q_1,q_2)=\left[\frac{\partial}{\partial q_1}w(q_1,q_2)\quad \frac{\partial}{\partial q_2}w(q_1,q_2)\right]^T \tag{4.7}$$

where $\frac{\partial w}{\partial q_i}$ is found by the absolute value differentiation rule

$$\frac{\partial}{\partial x}\left(\left|u(x)\right|\right)=\frac{u}{|u|}\frac{\partial u}{\partial x}=\operatorname{sgn}(u)\frac{\partial u}{\partial x} \tag{4.8}$$

The gradient vector field for a two link manipulator is shown in Figure 4.8, which indicates a maximum in the manipulability metric at $q_2=\frac{\pi}{2}$ radians. $\frac{\partial w}{\partial q_1}$ is zero for all values of q_1, because the manipulator has polar symmetry about its first joint.

For the two link manipulator, obtaining an analytical expression for the manipulability gradient is not difficult. However, for a redundant spatial manipulator with a high number of degrees of freedom, it may be more practical to numerically estimate the gradient of manipulability locally by calculating the value for w at a few different nearby points in the configuration space.

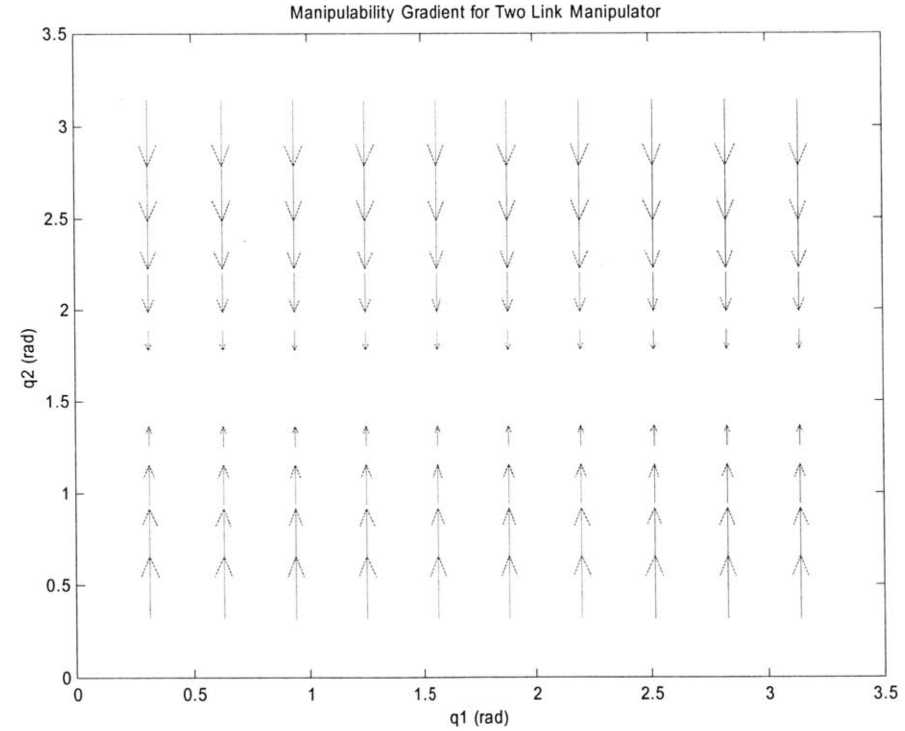


Figure 4.8: Manipulability Gradient in Joint Space for Two Link Manipulator

The gradient of the manipulability metric is added to the control law and functions as a type of flexible position control. The new term drives the manipulator in the direction that provides the maximum increase in manipulability while the manipulator simultaneously approaches the desired position and velocity. For the best system performance, the desired position should also be a point of maximum manipulability. Alternatively, the manipulability gradient term can be used instead of position control if eliminating relative velocity while avoiding singularities are higher priorities than position control during relative motion stabilization.

$$\boldsymbol{\tau} = \mathbf{J}^T \left(\mathbf{K}_D \left(\dot{\mathbf{x}}_d - \mathbf{J}\dot{\mathbf{q}} \right) + \mathbf{K}_P \left(\mathbf{x}_d - \mathbf{x} \right) \right) + \mathbf{K}_M \vec{\nabla} w \left(q_1, q_2 \right) \tag{4.9}$$

The results of applying this control law with a manipulability gain of $K_m = 10$ are shown in Figure 4.9. The maximum control torques and joint forces are further reduced to approximately half of their values with the original joint controller, without a significant change in settling time compared to the relative motion frame controller. The path of the center of mass of the space robot in the x-y plane of a coordinate system moving with the satellite is super-imposed on a plot of manipulability in Figure 4.10. As shown in the figure, the final position of the robot relative to the satellite is such that the manipulability of its manipulator is maximized.

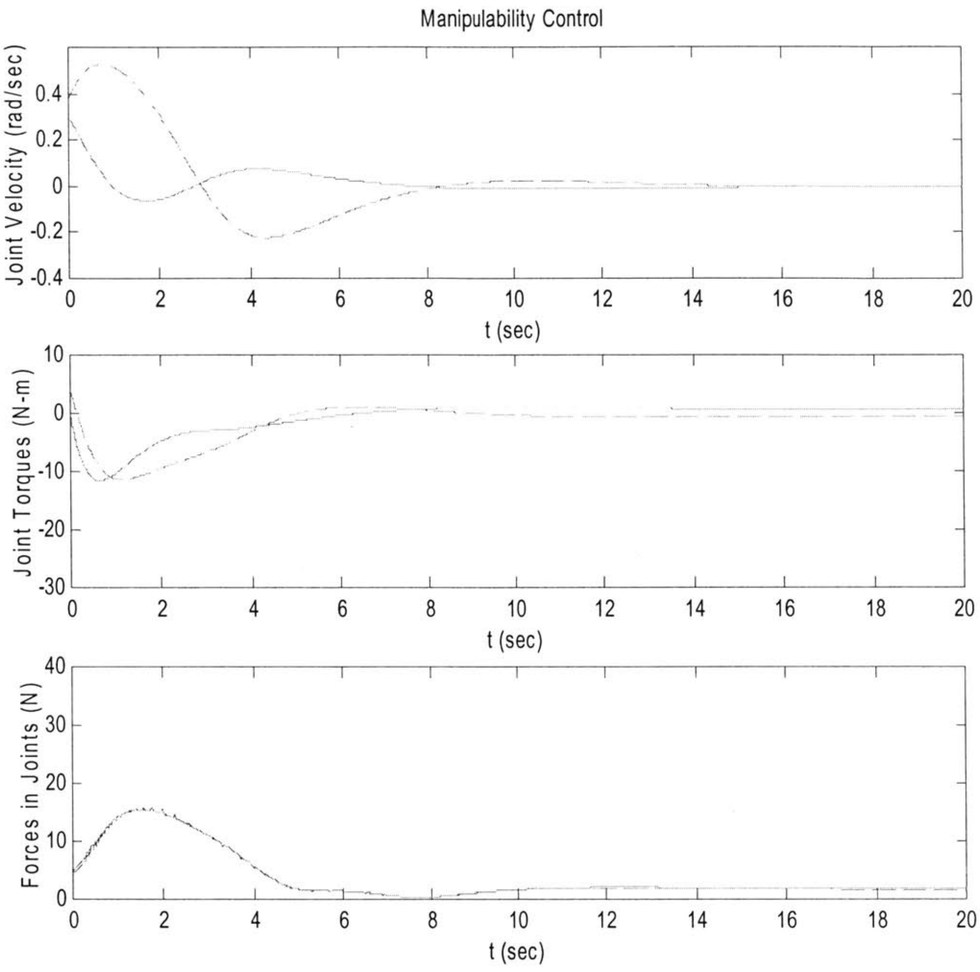


Figure 4.9: Manipulability Control Simulation Results

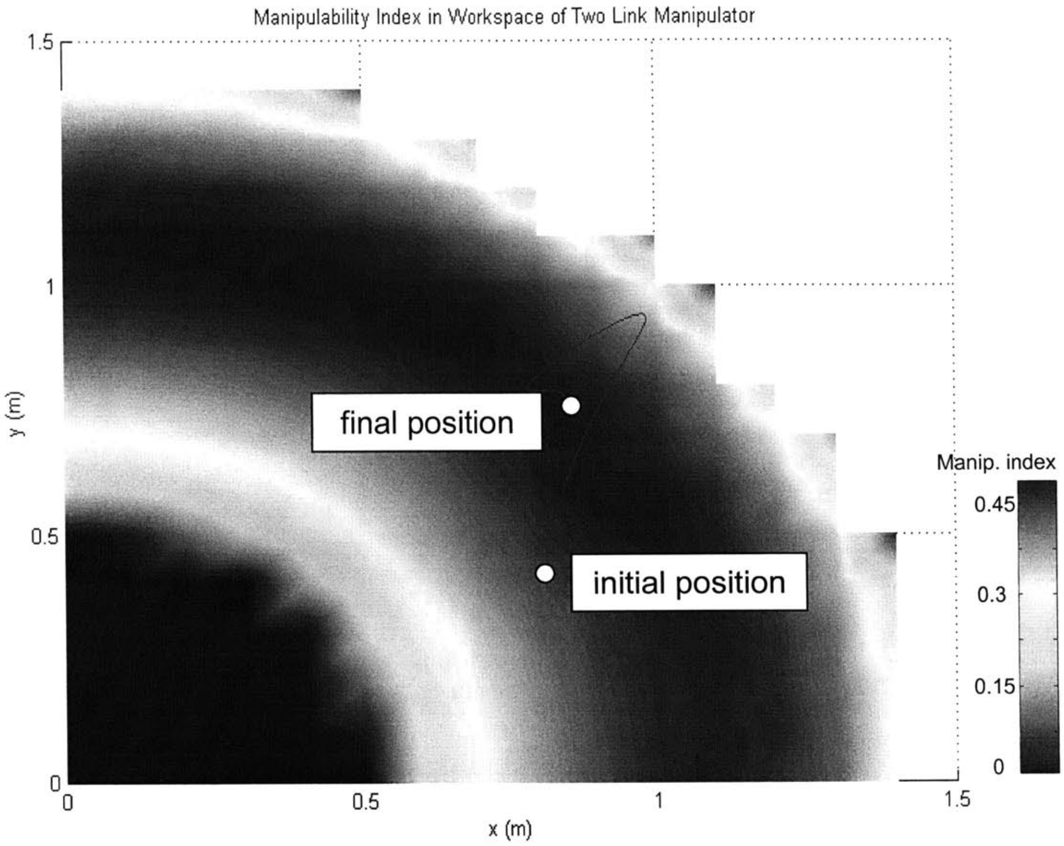


Figure 4.10: Trajectory for Manipulability Control

4.2.6 Minimization of Steady-State Joint Forces

After relative motion between the robot and satellite is eliminated, the system rotates at an angular velocity ω_{sys}, which is a function of the initial angular momentum of the system and its moment of inertia about its axis of rotation. This moment of inertia and the location of the system center of mass are dependent upon the final system configuration. The velocity of the center of mass of link 0 relative to the system center of mass is

$$\mathbf{v}_0 = \boldsymbol{\omega}_{sys} \times \mathbf{r}_0 \tag{4.10}$$

where $\mathbf{r}_0$ is the position of link 0 in a coordinate frame with its origin at the system center of mass. The centripetal force acting on link 0 is given by

$$\mathbf{F}_0 = m_0 \omega_{sys} v_0 \left(\frac{-\mathbf{r}_0}{|\mathbf{r}_0|} \right) \tag{4.11}$$

and the moment acting at joint 0 is

$$\mathbf{M}_0 = -\mathbf{b}_0 \times \mathbf{F}_0 \tag{4.12}$$

where $\mathbf{b}_0$ is the vector from the center of mass of link 0 to joint 0.

For each successive link, the velocity with respect to the system center of mass can be expressed as

$$\mathbf{v}_i = \boldsymbol{\omega}_{sys} \times \mathbf{r}_i \tag{4.13}$$

The force acting at the i^{th} joint is then the sum of the centripetal force acting at the center of mass of link i and the force at joint i-1.

$$\mathbf{F}_i = \mathbf{F}_{i-1} + m_i \omega_{sys} v_i \left(\frac{-\mathbf{r}_i}{|\mathbf{r}_i|} \right) \tag{4.14}$$

Likewise, the moment at the i^{th} joint is the torque created by $\mathbf{F}_i$ added to the moment at joint i-1.

$$\mathbf{M}_i = \mathbf{M}_{i-1} - \mathbf{b}_i \times \mathbf{F}_i \tag{4.15}$$

These steady-state forces and moments are a highly nonlinear function of the final system configuration. For a given set of system dimensions and inertial parameters, the force or moment in a particular joint can be numerically minimized as a function of the final joint angles. Using the same two link system used in the relative motion stabilization simulations, the magnitude of the steady-state force at the grasping point was numerically minimized. Figure 4.11 shows the magnitude of the steady-state force at the grasping point for several different final configuration as well as the configuration that produces the lowest force magnitude. The minimum steady state force sets the minimum forces and torques that the system must be able to tolerate. Also, it may be useful to know which configuration puts the smallest load on the manipulator after relative motion has been eliminated so that interactive forces between the bodies can be minimized during stabilization of the entire system.

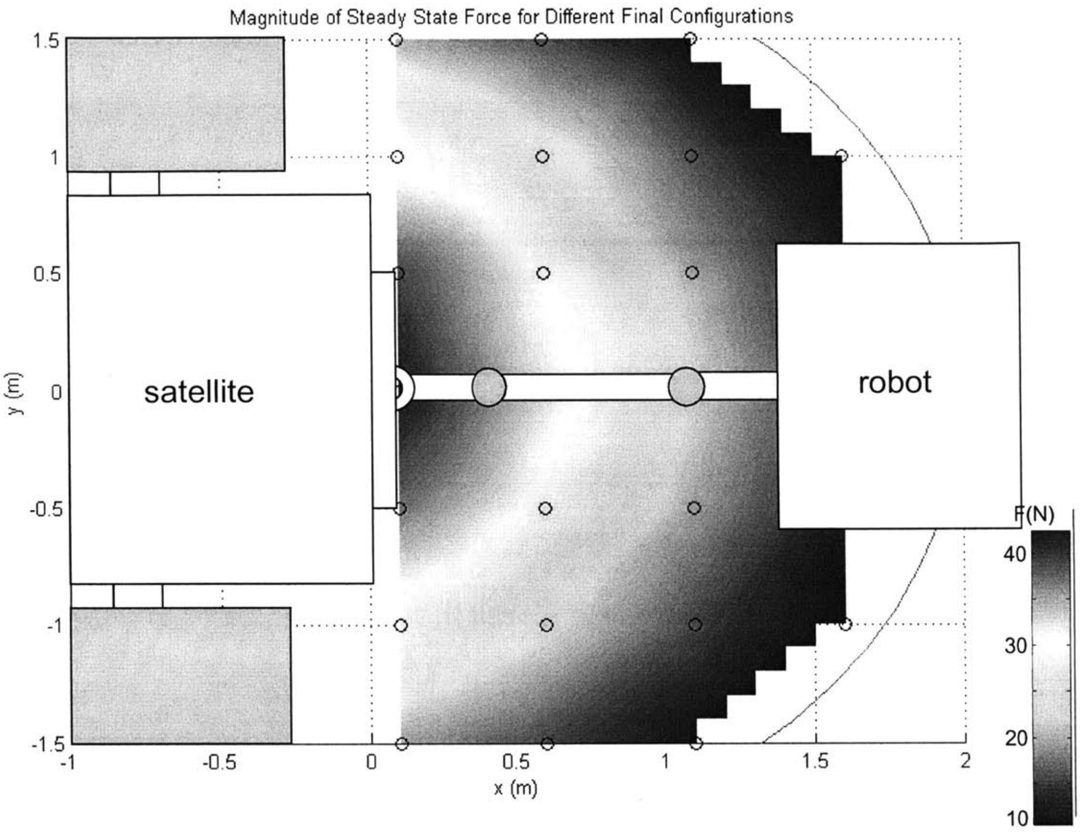


Figure 4.11: Steady-state Force at End-effecter for Different System Configurations

4.3 Implications for Grasp Planning

4.3.1 Optimal Initial Configuration

The performance of the manipulator controller during relative motion stabilization is highly dependent upon the initial conditions at the grasp time. In order to choose the optimal initial configuration for the system, there are a number of performance metrics that can be used. Maximum interactive force at the end-effecter is an important concern during satellite capture. The relationship between this maximum force and the initial configuration is complicated, so numerical optimization of a dynamic simulation may be used to find the initial configuration which minimizes the forces at the manipulator end-effecter. Simulations were run for the same system used to evaluate the controllers for several different initial system configurations. The maximum force corresponding to each location of the robot base in the satellite frame is shown in Figure 4.12 along with the configuration that produces the lowest maximum force for the initial velocity conditions used in the previous simulations.

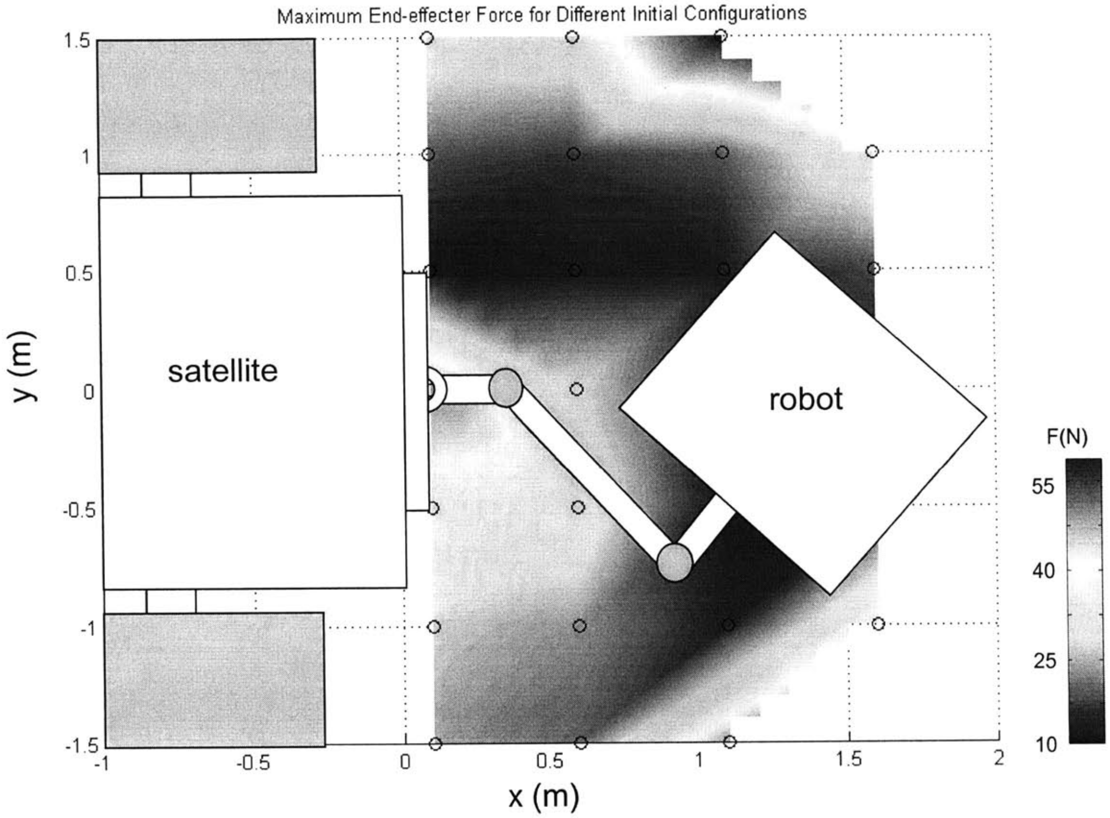


Figure 4.12: Maximum End-effecter Force for Different Initial Configurations

4.4 Chapter Summary

In this chapter, the relative motion damping task during the stabilization phase was analyzed in detail. The objectives and constraints for relative motion damping were presented, and several methods for the control of motion between the satellite and robot were compared. A control method based on the maximization of a manipulability metric was developed as a way to avoid singular system configurations. Dynamic simulations run within a numerical optimization routine were used to find the optimal initial configuration of a robot during the grasp task.

CHAPTER 5

CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORK

5.1 Conclusions

This thesis presented an overview of the technical challenges involved in the robotic capture of an uncontrolled satellite. The mission was divided in phases and a number of planning and control problems from each phase were addressed. Several planning and control algorithms were proposed as solutions to these problems, and the effectiveness of each algorithm was evaluated through dynamic simulation.

In Chapter 2, an approach trajectory planning method based on optimization techniques was presented and applied to two representative cases. The method accounts for important considerations such as maximizing safety, mitigating plume impingement, and minimizing fuel usage by minimizing a cost function made up of performance metrics. A trajectory tracking control method was developed and simulated using a free-flying robot with realistic models of sensor and actuator performance to demonstrate that the planned trajectory was physically realizable.

Chapter 3 evaluated several different methods of inertially referenced manipulator control for the purpose of manipulator during grasp, including generalized Jacobian inverse control, adaptive control, and robust control. The methods were first applied to a simplified planar case, and then the generalized Jacobian method was applied to a more complex spatial manipulator. The results of these methods demonstrated that modeling of the dynamic interactions between the base and manipulator of a free-flying space robot can increase the accuracy of endpoint control.

In Chapter 4, the relative motion damping task during the stabilization phase is analyzed in detail. The objectives and constraints for relative motion damping were presented, and several methods for the control of motion between the satellite and robot were compared. A control method based on the maximization of a manipulability metric was presented as a way of avoiding singular system configurations. It was shown that numerical optimization along with dynamic simulation can be used to find the optimal initial configuration of a robot during grasp.

5.2 Suggestions for Future Work

Robotic satellite capture is a relatively new mission concept, and this project began with a problem definition study to identify the important technical challenges of the mission. Accordingly, in addition to the solutions presented in this report, several questions were raised during the course of the study. A number of the questions are logical extensions of the current research topics, and are candidates for further research in the future.

To test the effectiveness of the proposed planning and control techniques, they could be experimentally validated for a simplified satellite capture case. While accurately reproducing weightless behavior in a terrestrial laboratory is difficult, free-floating or free-flying motion could be emulated for at least a planar case using redundant manipulators currently available at the FSRL. Further experimental validation could be performed using facilities available at NASDA.

Simple models of robots, sensors, and actuators were assumed in this study since the primary focus was to evaluate planning and control algorithms for robotic systems that have not yet been designed and constructed. Though the performance of the models was based on realistic components, space ready systems are subject to additional constraints. These constraints could be added to the models to test the sensitivity and robustness of the algorithms for more realistic systems. The objectives of the planning and control algorithms could be incorporated into the design of systems that are optimized for the tasks in a satellite capture mission.

A more demanding satellite capture mission involving a satellite with high inertial parameters or a high spin rate could require the use of multiple robots for capture and stabilization. The optimal approach planning method was applied to a case where two robots approached a target simultaneously, but the dynamic control methods were evaluated for the single robot case due to the complexities of running dynamic simulations with several bodies. A possible extension of this work could be to apply the proposed control methods to multiple cooperating robots, or robots equipped with multiple manipulators.

This thesis addressed only the first part of motion stabilization after grasp, which is relative motion elimination. Re-configuration to the optimal pose after relative motion has been eliminated is an important task in preparation for motion stabilization of the system as a whole. Metrics for choosing the optimal pose should be determined, such as equivalent stiffness of the manipulator chain or the effectiveness of attitude control actuators in a given configuration.

The most appropriate control method for stabilizing or de-spinning the entire satellite and robot system is dependent upon the nature of the system motion, and the best attitude control method may be different if the system is spinning, nutating, or tumbling, for example. This critical step in the stabilization process that will be necessary before further satellite servicing, so planning and control algorithms to deal with this problem should be studied. Also, alternative methods and novel devices for dissipating energy from a spinning system could be investigated to determine if passive dissipation could be used to stabilize the system.

The manipulability metric used to help avoid singular configurations in Chapter 4 was a function of manipulator joint angles, but did not account for the mass distribution of or velocities of the bodies in the system. The dynamic effects of the link masses and velocities should be considered when determining the ability of the manipulator to move freely in an arbitrary direction. A dynamic manipulability metric has been suggested to account for these effects, and the metric could possibly be applied to planning safe trajectories during relative motion damping.

REFERENCES

Clohessy, W., H., and Wiltshire, R., S., "Terminal Guidance System for Satellite Rendezvous" *Journal of the Aerospace Sciences*, Vol.15, September, 1960, pp.653-658

Dubowsky, S., Durfee, W., Kuklinski, A., Muller, U., Paul, I., and Pennington, J. "The Design and Implementation of a Laboratory Test Bed for Space Robotics: The VES MOD II" *Proceedings of the 23rd ASME Mechanisms Conference*, Minneapolis, MN, September 1994

Dubowsky, S. and Torres, M., "Path Planning for Space Manipulators to Minimize the Use of Attitude Control Fuel" *Proc. of the International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Kobe, Japan, November 1990

Dubowsky, S., and Papadopoulos, E., "The Kinematics and Control of Multi-Manipulator Free-flying and Free-floating Space Robotics System" *IEEE Transactions on Robotics and Automation*, vol. 9, no. 5, 1993, pp. 1-13

Ejiri, A. et al., "Satellite Berthing Experiment with a Two-armed Space Robot" *Proceedings of the 1994 IEEE International Conference on Robotics and Automation: ICRA94*,

Jacobsen, S., et. al., "Planning of Safe Kinematics Trajectories for Free-Flying Robots Approaching an Uncontrolled Spinning Satellite" *Proc. of ASME DETC*, Montreal, Canada, September 2002

Kaplan, M., H., "Modern Spacecraft Dynamics & Control" John Wiley & Sons Inc., 1976

Kawano, I., et al., "First Results of Autonomous Rendezvous Docking Technology Experiments on NASDA's ETS-VII Satellite" *IAF-98-A.3.09, 49th International Astronautical Congress*, 1998

Kawatmoto, S., Matsumoto, K., and Wakabayashi, S., "Ground Experiment of Mechanical Impulse Method for Uncontrollable Satellite Capturing" *Proceedings of the*

6^{th} International Symposium on Artificial Intelligence and Robotics & Automation in Space: i-SAIRAS 2001, Montreal, Canada 2001

Kindel, R., Hsu, D., Latombe, J., and Rock, S., "Kinodynamic Motion Planning Amidst Moving Obstacles" *IEEE International Conference on Robotics and Automation,* April 2000

King, D., 2001, "Space Servicing: Past, Present and Future", *Proceedings of the 6^{th} International Symposium on Artificial Intelligence and Robotics & Automation in Space: i-SAIRAS 2001*, Montreal, Canada 2001

Matsumoto, S. et. al., "Fly-by Approach and Guidance for Uncontrolled Rotating Satellite Capture" *AIAA Guidance Navigation and Control Conference*, Austin, TX, August 2003

Matsumoto, S., et. al., "Approach Planning and Guidance for Uncontrolled Rotating Satellite Capture Considering Collision Avoidance" *Proceedings of the 7th International Symposium on Artificial Intelligence and Robotics & Automation in Space: i-SAIRAS 2003*, Nara, Japan, May 2003

Matsumoto, S., et. al., "Satellite Capturing Strategy Using Agile Orbital Servicing Vehicle, Hyper-OSV" *Proc. of IEEE ICRA 2002*

Nakamura, Y. and Mukherjee, R., "Exploiting Nonholonomic Redundancy of Free-flying Space Robots", *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 4, August 1993

Nelder, J. A. and Mead, R., "A Simplex Method for Function Minimization." *Computing Journal* Volume 7, 308-313, 1965

Oda, M., "Space Robot Experiments on NASDA's ETS-VII Satellite – Preliminary Overview of the Experiment Results", *Proceedings of the 1999 International Conference on Robotics and Automation: ICRA 1999*, Detroit, MI 1999

Oda, M., “Experiences and Lessons Learned from the ETS-VII Robot Satellite” *Proceedings of the 2000 IEEE International Conference on Robotics and Automation: ICRA 2000*, San Francisco, CA, April 2000

Papadopoulos, E. and Dubowsky, S., "On the Dynamic Singularities in the Control of Free-Floating Space Manipulators," *Proc. of the Winter Annual Meeting of the ASME*, ASME DSC-Vol. 15, San Fransisco, CA December 1989

Papadopoulos, E. and Dubowsky, S., "On the Nature of Control Algorithms for Space Manipulators," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Cincinnati, OH May 1990

Papadopoulos, E. and Gonthier, Y., "Large Force Task Planning for Mobile and Redundant Robots," *Proc. of Int. Conf. on Intelligent Robots and Systems: IROS '95*, Pittsburgh, PA

Person, D. J., “The Glideslope Approach”, *Adcences in the Astronautical Sciences*, Vol.69, American Astronautical Society, San Diego, CA, 1989, pp.109-123; also AAS paper 89-162

Richards, A. et. al., “Spacecraft Trajectory Planning With Collision and Plume Avoidance Using Mixed-Integer Linear Programming” *AIAA Journal of Guidance, Control and Dynamics*, July 2002

Sakawa, Y., “Trajectory planning of a free-flying robot by using the optimal control.” *Optimal Control Applications and Methods*, 20, 235-248, 1999

Sidi, M., J., “Spacecraft dynamics and control: a practical engineering approach,” Cambridge University Press, 1997

Ullman, M., “Experiments in Autonomous Navigation and Control of a Multi-Manipulator Free-Flying Space Robot” *PhD thesis*, March 1993, Stanford University, Stanford, CA

Umetani, Y. and Yoshida, K., "Continuous Path Control of Space Manipulators Mounted on OMV" *Acta Astronautica*, Vol. 15, No. 12, 1987

Vafa, Z. and Dubowsky, S., "On the Dynamics of Manipulators in Space Using the Virtual Manipulator Approach", *Proceedings of the 1987 IEEE International Conference on Robotics and Automation: ICRA87*, Raleigh, NC

Vafa, Z. and Dubowsky, S., "On the Dynamics of Space Manipulators Using the Virtual Manipulator, with Applications to Path Planning" *Journal of Astronautical Sciences*, Vol. 38, No. 4, 1990

Wertz, J. R., "Spacecraft Attitude Determination and Control", Kluwer Academic Publishiers, 1978

Xu, Y. and Kanade, T., "Space Robotics: Dynamics and Control" Kluwer Academic Publishers, ISBN 0-7923-9265-5., November 1992

Yoshida, K., "Space Robot Dynamics and Control: To Orbit, From Orbit, and Future", *Robotics Research, The Ninth International Symposium, Eds, Hollerbach, J.M, and Koditschek, D.E*, pp.449-456, Springer. 2000

Yoshida, K. and Umetani, Y., "Control of a Space Free-flying Robot", *Proceedings of the 29th IEEE Conference on Decision and Control*, vol. 1, pp.97-102, 1990

Yoshida, K., et al., "Impact Dynamics of Space Long Reach Manipulators", *Proceedings of the 1996 IEEE International Conference on Robotics and Automation, ICRA 1996*, Minneapolis, MN

Zimpfer, D., and Spehar, P., "STS-71 Shuttle/Mir GNC Mission Overview", *Adcences in the Astronautical Sciences,* Vol.93, American Astronautical Society, San Diego, CA, 1996, pp.441-460; also AAS paper 96-129

APPENDIX A

PLUME IMPINGEMENT ANALYSIS

If a free-flying robot makes attitude or trajectory adjustment maneuvers while in close proximity to a satellite, the plume gases ejected from the robot's thrusters can impinge on the surface of the satellite or its panels with harmful effects. It is important to assess the extent of the effects of plume impingement, because avoiding plume impingement may require altering an approach trajectory in a way that is not optimal with respect to safety and capture mechanics, and the correct balance between these concerns should be found. While docking with the ISS, the Shuttle decelerates using indirect thruster firings, at a certain cost to fuel efficiency. The ETS-VII chaser satellite shut off its thrusters in close proximity of the target satellite, requiring an extremely slow docking speed and, hence, a long time period to complete the docking maneuver.

The detrimental effects of plume impingement can be divided into the categories of chemical, thermal, and physical effects. Harmful chemical effects include contamination of solar panel surfaces and fouling of optical lenses by ejected material, most commonly products of combustion. Thermal effects are any unwanted heating or uneven temperature distribution on a surface due to hot gases in the thruster plume. Physical effects may include structural damage to satellite appendages or dynamic disturbance of the satellite immediately prior to docking.

Chemical and thermal effects are a critical concern when using "hot gas" thrusters that burn hydrazine for propulsion. "Cold gas" thrusters, relying on compressed inert gas such as nitrogen, do not cause the same chemical or thermal effects and may, therefore, be preferable for use in satellite capture. However, dynamic disturbances are still a concern, and the magnitude of these potential disturbances should be determined.

Physical effects of plume impingement are difficult to predict accurately, due to the complex nature of plume dispersion in the vacuum of space. The probabilistic density distribution of a plume for a given thruster (Figure A.1) can be written as [Mayer and Prickett, 1986]

$$\rho(r,\theta) = \alpha \cdot \frac{\cos\left(\frac{\theta}{2}\right)^{\beta}}{V_{\infty} \cdot r^2} \qquad \text{(A.1)}$$

The parameters α, β, and V_∞ vary for different types of thrusters. For a typical 1N hydrazine thruster, these values are

$\alpha = 7.314\times10^{-4}$ kg/sec

$\beta = 39.098$

$V_\infty = 2403.6$ m/sec

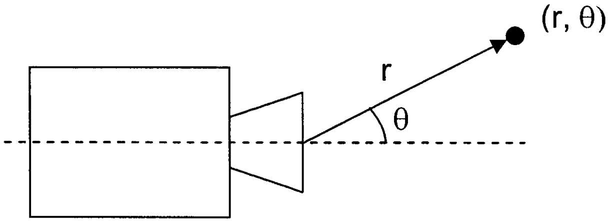


Figure A.1: Parameters in Thruster Plume Model

The pressure due to the plume at a point (r,θ) from the thruster is given by

$$P(r,\theta) = \frac{F}{A} = \frac{\dot{m} \cdot v}{A} = \rho \cdot v^2 \qquad \text{(A.2)}$$

The pressure distribution on a 1 m^2 plane locate 1 m from the thruster is shown in Figure A.2. Integrating pressure over the surface area yields total impingement force

$$F_{tot} = \int_A \rho v^2 dA \qquad \text{(A.3)}$$

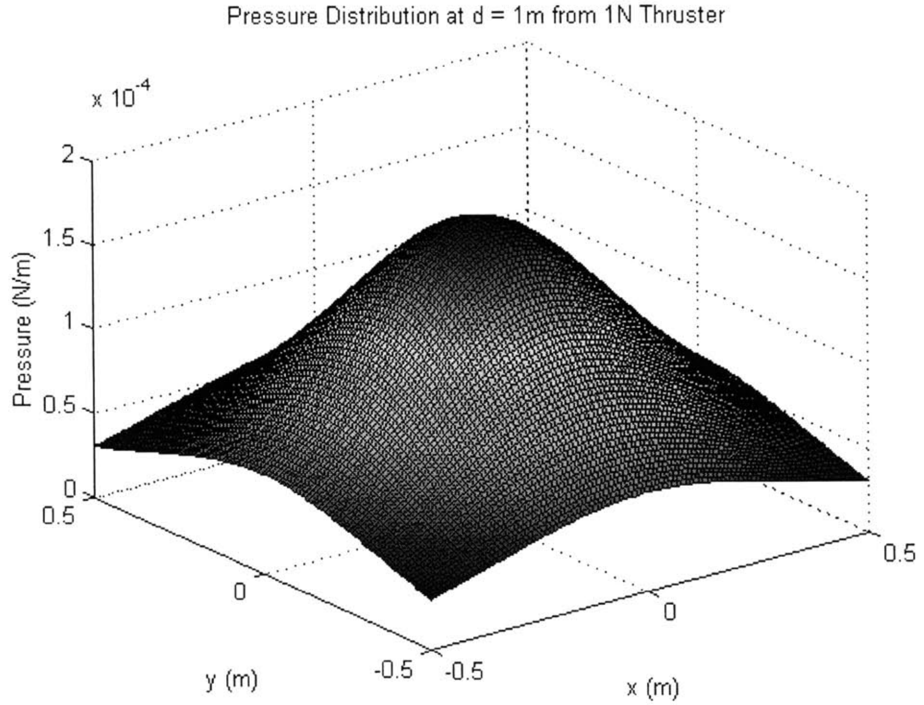


Figure A.2: Pressure Distribution at 1m from 1N Thruster

The force on 1 m^2 panel from a 1 N thruster is plotted vs. distance from the thruster in Figure A.3. This indicates that the physical effects of plume impingement decrease rapidly with distance, such that the effects are negligible at distances greater than a few meters. When the targets satellite has a mass on the order of 2,000 kg with a spin rate of more than 3 deg/sec, the dynamic disturbances due to plume impingement are on the order of state estimation errors. If larger combustion driven thrusters are required, plume impingement can present a problem and measures should be taken during approach trajectory planning to limit the amount of plume impingement that occurs.

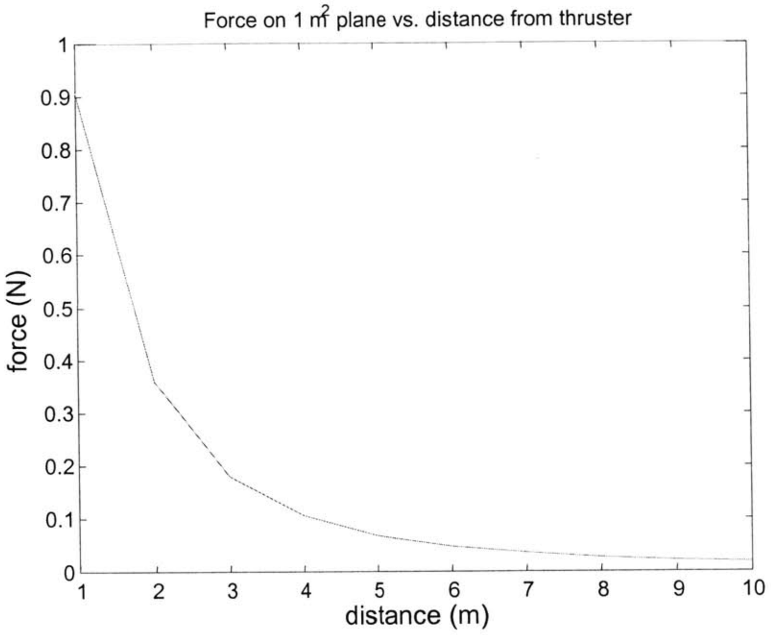


Figure A.3: Force on 1m^2 Plane vs. Distance from Thruster

APPENDIX

B

KINEMATICS AND DYNAMICS OF A TWO LINK SPACE ROBOT

The manipulator has two revolute joints and can control the position of the end-effecter in the x-y plane. The base is a square $2*b_0$ wide and each link is a_i+b_i in length, as shown in Figure B.1.

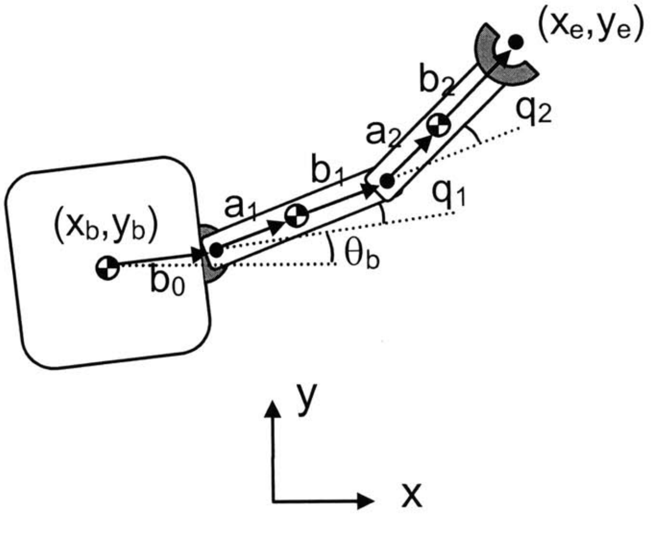


Figure B.1: Model of Free-floating Space Robot

The vectors between each joint and link center of mass can be written as

$$\mathbf{a}_i = \mathbf{R}_{0i} a_i \hat{\mathbf{i}} \tag{B.1}$$

$$\mathbf{b}_i = \mathbf{R}_{0i} b_i \hat{\mathbf{i}} \tag{B.2}$$

where the rotation matrices describing their orientation are defined below.

$$\mathbf{R}_0 = \begin{bmatrix} \cos(\theta_b) & -\sin(\theta_b) & 0 \\ \sin(\theta_b) & \cos(\theta_b) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{B.3}$$

$$\mathbf{R}_{01} = \begin{bmatrix} \cos(\theta_b + q_1) & -\sin(\theta_b + q_1) & 0 \\ \sin(\theta_b + q_1) & \cos(\theta_b + q_1) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{B.4}$$

$$\mathbf{R}_{02} = \begin{bmatrix} \cos(\theta_b + q_1 + q_2) & -\sin(\theta_b + q_1 + q_2) & 0 \\ \sin(\theta_b + q_1 + q_2) & \cos(\theta_b + q_1 + q_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{B.5}$$

Link vectors along each link are

$$\mathbf{l}_i = \mathbf{a}_i + \mathbf{b}_i \tag{B.6}$$

Then the rotational velocities of each link about its center of mass are

$$\boldsymbol{\omega}_0 = \dot{\theta}_b \hat{\mathbf{k}} \tag{B.7}$$

$$\boldsymbol{\omega}_1 = \boldsymbol{\omega}_0 + \dot{q}_1 \hat{\mathbf{k}} \tag{B.8}$$

$$\boldsymbol{\omega}_2 = \boldsymbol{\omega}_0 + \dot{q}_1 \hat{\mathbf{k}} + \dot{q}_2 \hat{\mathbf{k}} \tag{B.9}$$

The translational velocities in the inertial coordinate frame are

$$\mathbf{v}_0 = \dot{x}_b \hat{\mathbf{i}} + \dot{y}_b \hat{\mathbf{j}} \tag{B.10}$$

$$\mathbf{v}_1 = \mathbf{v}_0 + \boldsymbol{\omega}_0 \times \mathbf{r}_{01} + \dot{q}_1 \left(\hat{\mathbf{k}} \times \mathbf{l}_1 \right) \tag{B.11}$$

$$\mathbf{v}_2 = \mathbf{v}_0 + \boldsymbol{\omega}_0 \times \mathbf{r}_{02} + \dot{q}_1 \left(\hat{\mathbf{k}} \times \mathbf{l}_1 \right) + \dot{q}_2 \left(\hat{\mathbf{k}} \times \mathbf{l}_2 \right) \tag{B.12}$$

The kinetic energy of the system can then be written as

$$K = \frac{1}{2} \left(I_0 \boldsymbol{\omega}_0^T \boldsymbol{\omega}_0 + m_0 \mathbf{v}_0^T \mathbf{v}_0 + I_2 \boldsymbol{\omega}_1^T \boldsymbol{\omega}_1 + m_1 \mathbf{v}_1^T \mathbf{v}_1 + I_2 \boldsymbol{\omega}_2^T \boldsymbol{\omega}_2 + m_2 \mathbf{v}_2^T \mathbf{v}_2 \right) \tag{B.13}$$

Without any potential energy, K = T in the Lagrange equations

$$\frac{d}{dt} \left\{ \frac{\partial T}{\partial \dot{q}_i} \right\} - \frac{\partial T}{\partial q_i} = \tau_i \tag{B.14}$$

Combining the resulting equations of motion with the conservation of momentum equation

$$\sum_{i=0}^{2} I_i\boldsymbol{\omega}_i + m_i\mathbf{v}_i = 0 \tag{B.15}$$

as described in section 3.2, produces the reduced order model of the system.

$$\mathbf{H}^*(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}^*(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} = \boldsymbol{\tau} \tag{B.16}$$

The resulting equations are quite involved, even for the simple manipulator with two revolute joints. For example, the terms in the generalized Jacobian for the system, J*, are given below.

$J^* = [J^*_{11}\ J^*_{12};\ J^*_{21}\ J^*_{22}]$

where

J^*_{11}

```
-l1*sin(q1)*(m0*l1+m1*c1)/M-l2*sin(q2)*((m0+m1)*l2+m2*c2)/M-(-
m0*c0/M*l0*sin(q0)-l1*sin(q1)*(m0*l1+m1*c1)/M-
l2*sin(q2)*((m0+m1)*l2+m2*c2)/M)*(I1+m0*(m1+m2)/M*b1^2+(m0+m1)*m2/M*b1^2+m0*m2/
M*b1*c1+I2+(m0+m1)*m2/M*b2^2+(m0*(m1+m2)/M*c0*b1+m0*m2/M*c0*c1)*cos(q1)+2*(m0*m
2/M*b1*b2+(m0+m1)*m2/M*c1*b2)*cos(q2)+m0*m2/M*c0*b2*cos(q1+q2))/(I0+m0*(m1+m2)/
M*c0^2+I1+m0*(m1+m2)/M*b1^2+(m0+m1)*m2/M*b1^2+m0*m2/M*b1*c1+I2+(m0+m1)*m2/M*b2^
2+2*(m0*(m1+m2)/M*c0*b1+m0*m2/M*c0*c1)*cos(q1)+2*(m0*m2/M*b1*b2+(m0+m1)*m2/M*c1
*b2)*cos(q2)+2*m0*m2/M*c0*b2*cos(q1+q2))
```

J^*_{12}

```
-l2*sin(q2)*((m0+m1)*l2+m2*c2)/M-(-m0*c0/M*l0*sin(q0)-
l1*sin(q1)*(m0*l1+m1*c1)/M-
l2*sin(q2)*((m0+m1)*l2+m2*c2)/M)*(I2+(m0+m1)*m2/M*b2^2+(m0*m2/M*b1*b2+(m0+m1)*m
2/M*c1*b2)*cos(q2)+m0*m2/M*c0*b2*cos(q1+q2))/(I0+m0*(m1+m2)/M*c0^2+I1+m0*(m1+m2
)/M*b1^2+(m0+m1)*m2/M*b1^2+m0*m2/M*b1*c1+I2+(m0+m1)*m2/M*b2^2+2*(m0*(m1+m2)/M*c
0*b1+m0*m2/M*c0*c1)*cos(q1)+2*(m0*m2/M*b1*b2+(m0+m1)*m2/M*c1*b2)*cos(q2)+2*m0*m
2/M*c0*b2*cos(q1+q2))
```

J^*_{21}

```
l1*cos(q1)*(m0*l1+m1*c1)/M+l2*cos(q2)*((m0+m1)*l2+m2*c2)/M-
(m0*c0/M*l0*cos(q0)+l1*cos(q1)*(m0*l1+m1*c1)/M+l2*cos(q2)*((m0+m1)*l2+m2*c2)/M)
*(I1+m0*(m1+m2)/M*b1^2+(m0+m1)*m2/M*b1^2+m0*m2/M*b1*c1+I2+(m0+m1)*m2/M*b2^2+(m0
*(m1+m2)/M*c0*b1+m0*m2/M*c0*c1)*cos(q1)+2*(m0*m2/M*b1*b2+(m0+m1)*m2/M*c1*b2)*co
s(q2)+m0*m2/M*c0*b2*cos(q1+q2))/(I0+m0*(m1+m2)/M*c0^2+I1+m0*(m1+m2)/M*b1^2+(m0+
m1)*m2/M*b1^2+m0*m2/M*b1*c1+I2+(m0+m1)*m2/M*b2^2+2*(m0*(m1+m2)/M*c0*b1+m0*m2/M*
c0*c1)*cos(q1)+2*(m0*m2/M*b1*b2+(m0+m1)*m2/M*c1*b2)*cos(q2)+2*m0*m2/M*c0*b2*cos
(q1+q2))
```

J^*_{22}

```
l2*cos(q2)*((m0+m1)*l2+m2*c2)/M-
(m0*c0/M*l0*cos(q0)+l1*cos(q1)*(m0*l1+m1*c1)/M+l2*cos(q2)*((m0+m1)*l2+m2*c2)/M)
*(I2+(m0+m1)*m2/M*b2^2+(m0*m2/M*b1*b2+(m0+m1)*m2/M*c1*b2)*cos(q2)+m0*m2/M*c0*b2
*cos(q1+q2))/(I0+m0*(m1+m2)/M*c0^2+I1+m0*(m1+m2)/M*b1^2+(m0+m1)*m2/M*b1^2+m0*m2
/M*b1*c1+I2+(m0+m1)*m2/M*b2^2+2*(m0*(m1+m2)/M*c0*b1+m0*m2/M*c0*c1)*cos(q1)+2*(m
0*m2/M*b1*b2+(m0+m1)*m2/M*c1*b2)*cos(q2)+2*m0*m2/M*c0*b2*cos(q1+q2))
```