AMES GRANT — IN-37-CR

154097

31P.

SIXTH SEMI-ANNUAL REPORT

ON

RESEARCH ON

CONTROL OF FREE-FLYING SPACE ROBOT MANIPULATOR SYSTEMS

(NASA-CR-183104) CONTROL OF FREE-FLYING SPACE ROBOT MANIPULATOR SYSTEMS Semiannual Report No. 6, Sep. 1987 - Feb. 1988 (Stanford Univ.) 31 p CSCL 13I

N88-27563

Unclas

G3/37 0154097

Submitted to

Dr. Henry Lum, Jr., Chief, Information Sciences Division
Ames Research Center, Moffett Field, CA 94035

by

The Stanford University Aerospace Robotics Laboratory
Department of Aeronautics and Astronautics
Stanford University, Stanford, CA 94305

Research Performed Under NASA Contract NCC 2-333
During the period September 1987 through February 1988

Professor Robert H. Cannon Jr.
Principal Investigator

SIXTH SEMI-ANNUAL REPORT

ON

RESEARCH ON

CONTROL OF FREE-FLYING SPACE ROBOT MANIPULATOR SYSTEMS

Submitted to

Dr. Henry Lum, Jr., Chief, Information Sciences Division
Ames Research Center, Moffett Field, CA 94035

by

The Stanford University Aerospace Robotics Laboratory
Department of Aeronautics and Astronautics
Stanford University, Stanford, CA 94305

Research Performed Under NASA Contract NCC 2-333
During the period September 1987 through February 1988

Professor Robert H. Cannon Jr.
Principal Investigator

~

Contents

List of Figures

~

Chapter 1

Introduction

This document reports on the work done under NASA Cooperative Agreement NCC 2-333 during the period September 1987 through February 1988. The research is carried out by a team of graduate students comprising the Stanford University Aerospace Robotics Laboratory under the direction of Professor Robert H. Cannon, Jr. The goal of this research is to develop and test new control techniques for self-contained, autonomous free-flying space robots. Free-flying space robots are envisioned as a key element of any successful long term presence in space. These robots must be capable of performing the assembly, maintenance, and inspection, and repair tasks that currently require astronaut extra-vehicular activity (EVA). Use of robots will provide economic savings as well as improved astronaut safety by reducing and in many cased eliminating the need for human EVA.

The focus of our work is to develop and carry out a set of research projects using laboratory models of satellite robots. These devices use air cushion technology to simulate in two dimensions the drag-free, zero-g conditions of space. Using two large granite surface plates (6' by 12' and 9' by 12') which serve as the platforms for these experiments we are able to reduce gravity induced accelerations to under $10^{-5}g$ with a corresponding drag-to-weight ratio of about 10^{-4}—a very good approximation to the actual conditions in space.

Our current work is divided into five major projects or research areas: Cooperative Manipulation on a Fixed Base, Cooperative Manipulation on a Free-Floating Base, Global Navigation and Control of a Free-Floating Robot, an alternative transport mode called LEAP (Locomotion Enhancement via Arm Push-Off), and Adaptive Control of LEAP.

The fixed-base cooperative manipulation work represents our initial entry into multiple arm cooperation and high-level control with a sophisticated user interface. This experiment is now fully on-line and has already produced several significant new results.

The floating-base cooperative manipulation project strives to migrate some of the new technologies developed in the fixed-base work onto a floating base. This experiment will be using our second generation space-robot model which is still under construction.

The global control and navigation experiment seeks to demonstrate simultaneous control of the robot manipulators and the robot base position so that tasks can be accomplished while the base is undergoing a controlled motion.

The LEAP activity was started about a year ago with the goal of providing a viable alternative to expendable gas thrusters for vehicle propulsion wherein the robot uses its manipulators to throw itself from place to place. This work will be carried out with a slightly revised version of second generation space robot which is currently under construction.

The adaptive LEAP project is a new activity that was started during this report period. Because the successful execution of the LEAP technique requires for an accurate model of the robot and payload mass properties it was deemed as an attractive testbed for adaptive control technology. Initial studies are underway to evaluate various adaptive control algorithms.

The chapters that follow give detailed progress and status reports on a project by project basis. Also included under separate cover is a recently completed thesis by Dr. Harold L. Alexander entitled "Experiments in Control of Satellite Manipulators." This document (SUDAAR 565) represents an in-depth report on the initial work done in satellite robotics at Stanford University.

Chapter 2

Fixed-Base Cooperative Manipulation Experiment

Stan Schneider

2.1 Introduction

To accelerate our development of multi-armed, free-flying satellite manipulators, we have developed a fixed-base cooperative manipulation facility. Although the manipulator arms are fixed, they manipulate free-flying objects. By allowing allow us to quickly experiment with cooperative algorithms, this facility greatly expedites our study of space-based manipulation and assembly. This section describes the progress made to date in our research on cooperative manipulation.

Progress Summary

The major activities completed during the period September, 1987 through February, 1988 were:

- Continued development of the cooperating arms experimental system
- Designed and implemented a complete real-time software environment, including: calibration support, flexible execution control, real-time data collection and display, symbolic debugging, and a user-friendly command interface. The system provides the ARL with an intimate link between the Sun Unix development environment and the real-time control system.
- Implemented automated calibration for all arm sensor systems
- Demonstrated single arm inverse dynamic impedance control
- Demonstrated initial dual-arm coordinated control

Background: research goals

Space construction requires the manipulation of large, delicate objects. Single manipulator arms are incapable of quickly maneuvering these objects without exerting large local torques. Multiple cooperating arms do not suffer from this limitation. Unfortunately, cooperative robotic manipulation technology is not yet well understood. The goal of this project is to study the problem of cooperative manipulation in a weightless environment, and experimentally demonstrate a cooperative robotic assembly.

Four aspects are to be studied in detail:

- The dynamic control of multiple arm manipulation systems
- The utilization of video "vision" data for real-time control
- Real-time software structuring for cooperative robotic systems
- User interfacing: the acquisition and utilization of strategic commands

2.2 Facility Development

The fixed-base cooperation facility consists of a pair of two-link manipulators, affixed to the side of a "small" granite table (4 x 8 feet). Each arm is of the popular SCARA configuration—basically anthropomorphic, with vertical-axis, revolute "shoulder" and "elbow" joints. The arms are capable of motion in the plane of the table, and can interact with air-cushion objects floating on the granite surface. During the report period, several components of the fixed-base facility underwent evolutionary development.

Force sensing gripper

The original end-effector design, reported in the last status report, was a very simple 5 inch long 1/4 inch square aluminum beam. When attached to a 1 Kg object, this beam had a first cantilever vibration mode of approximately 16 Hz. This low frequency unmodeled dynamic response caused the high-performance controllers described below to be unstable.

To alleviate this problem, we have developed a new gripper design with a much stiffer beam assembly. Figure 2.1 compares the two beam designs. Instead of the simple long beam, the new design employs a thick (1/2 inch square) upper beam, and a much shorter (3/4 inch long) 1/4 inch square sensitive section. Changing to much more sensitive semiconductor strain gauges allows this design to retain high force sensitivity. A theoretical analysis of the bending characteristics of the new design predicted acceptable force signal levels. Our experimental experience with the new design has confirmed these analyses—The gripper has acceptable sensitivity without the problematic unmodeled dynamics.

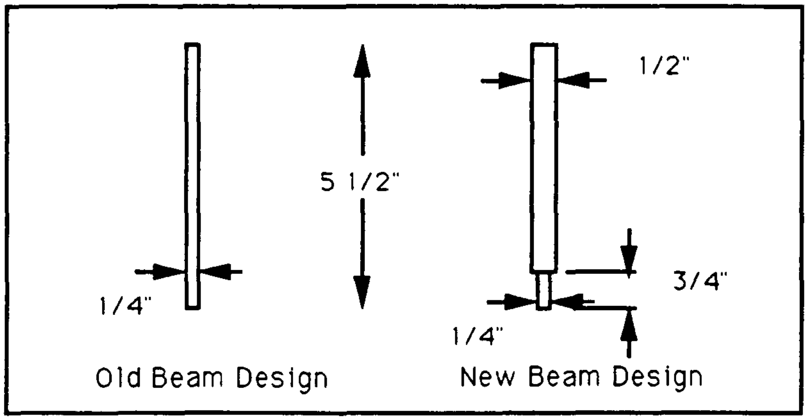

Figure 2.1: Force Sensing Beam Comparison

Vision system

During this report period, we procured and installed the basic components of the vision system. A CCD television camera was purchased, and mounted to provide an overhead view. A Datacube video processing system was also installed. Initialization programs for the system were written and tested. We are now beginning to develop the required software for this system.

Floating object development

Our floating air-cushion objects have also undergone evolutionary development since the last report. Recall that these objects are independent miniature air-cushion vehicles, equipped with a small battery-powered aquarium pump air supply. The arms can manipulate them with the grippers described above, thus providing a two-dimensional simulation of space-based manipulation.

The object design has proven acceptable, and was used for the initial experiments described below. However, there are still two problems to be addressed. First, long term trials have shown that our plexiglass pads suffer from slow warping. This leads to unsatisfactory flotation. New pads, using honeycomb aluminum, are under construction. These should be much more dimensionally stable.

Secondly, the aquarium pump causes considerable vibration. The vibration is due to two effects: the off-center pump drive mechanism rotation, and the pulsed air flow beneath the floating pad. Surprisingly, the latter seems to be the greater effect. Initial experiments with simple plenum chambers in the air flow lines greatly reduced this problem. To reduce the drive mechanism vibration, we also plan to shock-mount the aquarium pump.

2.3 Computer System

Our real-time computer system combines a proven UNIX development environment with high performance real-time processing hardware. Motorola 68020/68881 single board processors running the pSOS real-time kernel provide inexpensive real-time processing power. VME bus shared-memory communications permit efficient multiprocessor operation. The real-time processors are linked, via the VME bus, to our Sun/3 engineering workstations. Thus, we benefit from Sun's superb programming environment, while providing the capacity for relatively cheap, unlimited processing expansion.

Real-time software environment

Considerable effort has already been directed towards real-time software development tools. We have developed a real-time control systems software package that includes extensive calibration support, flexible execution control, real-time data collection, symbolic debugging support, and a user-friendly interface. In addition, companion software running on the Sun provides real-time data monitoring and display, as well as a direct data interface to the Sun's analysis tools, such as Pro-Matlab[3].

We have also developed an integrated simulation module interface. The interface is capable of supporting several types of simulators—we have already ported the iterative simulation package described in the last report, as well as a more traditional simulation package. This allows direct hardware replacement simulation, while providing all the services of the real-time package. Finally, we are adding the ability to run under VxWorks[1], and thus communicate with standard network interfaces. This will allow us to transfer the fixed-base software to the mobile robotic systems.

2.4 Calibration

Automated sensor calibration programs were developed for: joint angular positions, joint pseudo-velocities, probe forces, and motor torque outputs. This section describes the calibration methodologies.

Joint angle sensor calibration

Joint angular positions are calibrated with the help of a "pegboard" calibration grid. A simple fixture was made to fit into four holes in the pegboard while holding the arm's gripper. Another simple fixture holds the pegboard itself and provides a simple method of positioning the arm at a known location. Although the pegboard positioning system is not highly repeatable—the holes are rather "sloppy"—it is very accurate over large distances; the holes are uniformly spaced on one inch centers over the entire surface.

[1] VxWorks is described in depth in chapter 3

The joint angle calibration program prompts the user to position the arm at many different locations. It then uses the arm inverse kinematics to calculate the actual joint angles at each location. Measured joint angles are also taken at each position. A linear regression least-squares fit then yields scale and offset calibration coefficients for the joint angle sensors.

Joint velocity sensor calibration

After joint angle calibrations are complete, joint velocity calibration is done. Velocity calibration is rather simple: the arm is caused to slew across its workspace while the velocity signals are integrated. Comparison with the difference in joint position measurements then yields velocity calibration coefficients.

Probe force sensor calibration

The probe force sensor calibration is a more involved process. The major complicating factor is cross-talk between the orthogonally mounted strain gauge pairs. Thus, instead of the simple "scale and offset" calibration used above, a 2x2 matrix is needed to relate strain gauge measurements to actual force. Thus, we desire a matrix C, such that for strain gauge measurements v_1 and v_2, the forces f_1 and f_2 are:

$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} c_1 & c_2 \\ c_3 & c_4 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

or

$$\mathbf{f} = \mathbf{C}\mathbf{v}$$

The calibration algorithm is as follows: A simple pulley and weight system is used to exert a known force on the tip of the arm. The known location of the pulley and the arm joint angles are used to calculate the forces actually experienced by the end point force sensor. A measurement of the force sensor response is taken.

Take p such measurements of $\mathbf{v_i}$ with known forces $\mathbf{f_i}$. This yields:

$$\begin{bmatrix} \mathbf{f_1} & \mathbf{f_2} & \dots & \mathbf{f_p} \end{bmatrix} = [\mathbf{C}] \begin{bmatrix} \mathbf{v_1} & \mathbf{v_2} & \dots & \mathbf{v_p} \end{bmatrix}$$

Which we denote:

$$\mathbf{F} = \mathbf{C}\mathbf{V}$$

Let $\mathbf{V}^\dagger = \mathbf{V}^T \left(\mathbf{V}\mathbf{V}^T\right)^{-1}$ be the right pseudo-inverse of $\mathbf{V}$. Then

$$\mathbf{C} = \mathbf{F}\mathbf{V}^\dagger$$

is the least-squares optimal $\mathbf{C}$ calibration matrix.

2.5 Multiprocessor real-time structuring

Several different implementations of the proposed client-server real-time structuring methodology described in the original proposal were studied. According to this paradigm, the real-time software is divided into small, independently executing modules, each with a well-defined function in the controller data flow path. The modules communicate their data to other modules via message passing.

For example, one processor (the server) might be assigned to read and process the analog inputs. The pre-processed data is then sent via a message to a client process running on another CPU. The advantage of this scheme is that changes to the analog (server) environment are well isolated from the client code—the system is highly modular.

Unfortunately, the originally proposed scheme executed too slowly for practical use. Straightforward client-server operation requires two message transactions per client per loop: First, the client sends a message specifying what data is needed, then the server replies with the data. This has two problems: the client has nothing to do while the server processes its request, and the server must process an incoming message (which also forces a context switch) for each client, every sample. This overhead reduced the maximum loop rate from approximately 700 Hz (running a simple loop on one processor) to 250 Hz. This was deemed unacceptable.

To resolve this problem without abandoning the desirable structuring, a simpler scheme was implemented. Under this scheme, a client registers a "recurring" data request with the server at initialization time. The server then repeatedly sends the data at the sampling interval. This only requires one message per loop. In addition, the server can immediately begin reading the next set of data after serving the last client. Thus, the server's data reading operation is overlapped with the client's processing. This simpler scheme worked satisfactorily—the loop rate achievable (for one client) was in the 1000 Hz range.

The biggest disadvantage of this method is the loading on the server. The fixed-base facility has a three-processor computer system. The processing load divides naturally as one processor to calculate each arm's dynamics, etc., and one processor for "system" level tasks, such as vision, object dynamics, etc. If the "system" processor is heavily loaded processing sensor data service requests, its other responsibilities may suffer. Evaluation of this potential problem will have to await the implementation of the vision system software.

2.6 Controllers

We are examining interfaces between the dynamic forces and motions of the robotic manipulators, and higher-level strategic inputs. As a first attempt, we are investigating the application of Nevill Hogan's[2] impedance control concept to multi-arm—and multi-vehicle—cooperative tasks. Impedance control is very attractive for cooperative tasks because it allows direct control of the interaction between the cooperating agents; control of the mechanical power flow from manipulation system to environment. The implementation for multiple arms, however, is not well understood. We have made considerable progress on this front since the last report.

Impedance control overview Impedance control differs from other traditional forms of control in that instead of controlling a state variable—such as position, velocity, or force—to a desired condition, it enforces a relationship between these variables. The simplest example both to understand and implement is a second order linear impedance; a spring-mass-damper system. A manipulator under this type of impedance control would behave to external forces as if it were a simple second order system. Thus, its interactions with the environment are explicitly controlled.

This type of control has many desirable attributes. Chief among them is the ability to come into contact with a hard surface without losing stability. Impedance control is thus ideal for tasks requiring assembly or other contact with external systems.

Notation In the discussions below, $\mathbf{p}$ refers to the (x,y) tip position of an arm, and $\mathbf{q}$ refers to the vector of joint angles. The vector of applied joint torques is τ. $\mathbf{J}$ refers to the arm's Jacobian matrix, defined by $\dot{\mathbf{p}} = \mathbf{J}\dot{\mathbf{q}}$. The force applied to the arm tip is denoted $\mathbf{f}$. The arm's equations of motion are:

$$\tau = \mathbf{M}\ddot{\mathbf{q}} + \mathbf{C} + \mathbf{J}^T\mathbf{f}$$

Where $\mathbf{M}$ is the manipulator mass matrix, and $\mathbf{C}$ is a catch-all vector of coriolis, friction, etc. forces. If appropriate, subscripts will be appended to all of these symbols to distinguish the arms. Thus, for example, $\mathbf{p_i}$ denotes the (x, y) tip position of the i^{th} arm.

Single arm controllers

Position-Derivative control Collocated Position-Derivative control implements the algorithm:

$$\tau = K_p(\mathbf{q}_{desired} - \mathbf{q}) + K_v(\dot{\mathbf{q}}_{desired} - \dot{\mathbf{q}})$$

for each joint. This controller was the first used to test the arm's operation. It offers no control of the interaction forces between the arm and it's environment. It also does no dynamic compensation. It has two advantages: for "reasonable" gains, a PD controller is guaranteed stable, and it is very simple to implement.

Kinematic impedance control The kinematic impedance control algorithm is quite simple. From the principle of virtual work [Khatib], a force $\mathbf{f}$ applied to the arm tip is produced by a torque τ, where:

$$\tau = \mathbf{J}^T\mathbf{f}$$

If we use for $\mathbf{f}$ a simple (static and massless) spring / damper relationship:

$$\mathbf{f} = K_p(\mathbf{p}_{desired} - \mathbf{p}) + K_v(\dot{\mathbf{p}}_{desired} - \dot{\mathbf{p}})$$

then the arm will behave with a controlled impedance. This controller completely ignores the arm dynamics; it is based on a purely static analysis.

Implementation of this controller on our robotic arm was relatively straight-forward. As expected, small slow motions were executed well, but rapid slews suffer from dynamic errors.

Dynamic impedance control To compensate for the arm dynamics, differentiate the arm kinematic relation $\dot{\mathbf{p}} = \mathbf{J}\dot{\mathbf{q}}$ and solve for $\ddot{\mathbf{q}}$ to yield:

$$\ddot{\mathbf{q}} = \mathbf{J}^{-1}\left[\ddot{\mathbf{p}} - \dot{\mathbf{J}}\dot{\mathbf{q}}\right]$$

Now use the dynamic spring relationship:

$$m_{desired}\ddot{\mathbf{p}} = K_p(\mathbf{p}_{desired} - \mathbf{p}) + K_v(\dot{\mathbf{p}}_{desired} - \dot{\mathbf{p}})$$

Substitution into the arm equations of motion:

$$\tau = \mathbf{M}\ddot{\mathbf{q}} + \mathbf{C} + \mathbf{J}^T\mathbf{f}$$

yields a dynamically compensated "computed torque" impedance controller. In this relationship, $\mathbf{f}$ is the measured external force on the manipulator tip. This controller thus incorporates force feedback to enforce the impedance relationship.

This controller successfully compensated for the dynamics of our experimental manipulator. Large straight-line slews were executed with no appreciable deviation from the desired trajectory. Disturbances were handled as desired—if a critically damped spring-mass-damper system was specified, the actual response was essentially critically damped.

Coordinated arm controllers

We define any algorithm that controls the arms independently—but possibly on coordinated trajectories—as coordinated arm control. The arms are "coordinated" rather than "cooperative" because they respond neither to each other's actions or inputs, nor to the object dynamics. The three controllers described above were used to control two coordinated arms moving a single object. This section discusses the performance of these controllers.

The coordinated PD controllers were successful in moving the object, but no control of the forces of interaction was possible. Object trajectory control was similarly poor. This is obviously not an acceptable cooperation algorithm.

The kinematic impedance controller allowed cooperation for very slow motions, but for slews at even relatively moderate speeds, the dynamics of the arms became very significant, and the control deteriorated.

When this project was begun, we were attracted to impedance control for its promise of allowing cooperative motions without imposing dynamic over-constraint. We had expected to perform all our cooperative tasks using coordinated dynamic impedance control. Coordinated impedance control can be visualized as attaching two "springs" (actually spring-damper systems) to the object—one at each arm's attachment point. The system can then conceptually manipulate the object by moving the other end of these springs. Small errors would not result in large forces on the object, they would simply stretch the virtual springs a little, putting a very small additional force on the object.

In most cases, the dynamic impedance controller provided acceptable control of the interaction forces. It also insulated the object motion from the object dynamics. For linear slews without rotation, critically damped response to disturbances was achievable. However, as the next section outlines, coordinated control suffers from many limitations.

Cooperative controllers

Limitations of coordinated control Coordinated controllers, by their nature, suffer from several limitations. All of these limitations stem from the same problem: the failure of the control algorithm to consider the full dynamics of the object.

First, object control in all three pertinent degrees of freedom (x, y, and rotation) is very difficult to do simultaneously. For instance, using the coordinated impedance control described above, if the virtual spring-damper gains are selected to provide critically damped disturbance rejection in both the x and y directions, then the rotation dimension response—the response to disturbance torques—will not be critically damped.

Second, acceleration feed-forward is very difficult to do. While the arm controllers may be provided with desired accelerations, without a dynamic model of the object the arms cannot also be provided with the expected forces that the motion will produce. When the impedance controllers respond to the applied "external" forces, trajectory tracking performance deteriorates.

Third, the inter-arm forces, i.e. the "tension" or "compression" forces, are not explicitly controlled. Ideally the arms cooperate to produce the forces on the object required to produce the desired accelerations, while the object internal forces—those that do not produce motion—are explicitly controlled. With coordinated impedance control, the forces between the arms at rest is determined by the ratio of the distances between the actual grip points and the virtual spring endpoints. However, the lack of dynamic information about the object prevents dynamic control of the inter-arm forces.

Finally, and perhaps most importantly, this type of control is not very intuitive to the user. To adjust the control, the user must select "spring" gains in x and y for both arms. Automatic gain selection is feasible, but this does not address the rotational degree of freedom problem, nor the internal force problem. Determination of the gains required to effect an assembly—which most likely requires non-symmetric action of the arms—is even more cloudy.

Object impedance control To remedy this situation, we propose to develop a controller that attaches the impedance virtual spring not to each arm, but to the object itself. Thus the object will behave as if it were attached to its environment by linear spring-damper systems in x and y, and also by a torsional spring to control rotational orientation.

This controller should alleviate all of the problems outlined above. The user interface to this controller is quite simple and straight-forward. The user is presented with only the selection of the object's behavior—the arms are very much an abstract manipulation system. Gains can be automatically set to provide critical damping in all degrees of freedom simultaneously. It should also be capable of remote center of compliance (RCC) operation, thus permitting simple and efficient part mating and insertion operations.

This type of controller requires consideration of the dynamics of the entire system; both arms and manipulated object. Utilization of the full system equations of motion is one method of performing this control, but we have developed a candidate controller that instead is able to utilize the force-sensing grippers to isolate the system into three systems: arm1, arm2, and the common object. This vastly simplifies the computations required at runtime.

2.7 User interface

To begin our study of high-level strategic inputs, we have developed a simple mouse-based user interface. This program simply displays several "objects" on the screen, and allows the user to drag them about with the mouse. The user's trajectory is transmitted to the real-time system, and the arms then move the object along the directed path. This system is quite simple at present, but demonstrates the concept of a simple graphical interface that we intend to develop.

As noted above, this approach of developing simple, working implementations of all the control "layers" simultaneously is quite beneficial. Even this simple user interface pointed out some of the difficulties with our initial attempt at cooperative control.

2.8 Future Work

During the next period, we will develop and implement the object impedance control concept outlined above. This should allow us to complete a cooperative arm assembly demonstration. We also expect to utilize vision system input data for the first time, and perform object acquisition tasks. The user interface work is progressing—we plan to expand the simple mouse-based command input presently operational to a full user control panel. By the end of the next period, we expect to have completed the fixed-base cooperative manipulation experiment, and will begin the technology documentation (thesis preparation) process.

Chapter 3

Multiple Arm Cooperation on a Free-Flying Robot

Ross Koningstein

3.1 Introduction

This chapter summarizes the work performed on multiple arm cooperation on a free-flying robot. This work represents one of the basic technologies required for space based manipulation. The continuing work in the design and construction of the two armed free-flying robot experiment will be discussed in the following section, in particular, the power system for the free-flying experiment will be discussed.

3.2 Motivation

Free-flying satellites rely on solar cells, rechargable batteries, and perhaps nuclear power sources in order to function. Our laboratory space robot simulator vehicle, although not a real satellite, also requires an autonomous power system. The space robot simulator vehicle relies on its physical disconnection from the laboratory in order to faithfully simulate a two-dimensional free-flying robot. The experiment should be able to function for extended periods of free-flight. Due to the finite lifetime of rechargable battery packs, these batteries will need to be replaced by fresh batteries during this period. To ensure that vital onboard systems such as the computer remain functional, the power delivery to onboard systems should not be interrupted when onboard energy sources are replaced, or external power sources engaged or disengaged. It is desirable that the vehicle be operable when connected to an external power source (eg. in order to do computer software development) without draining battery packs. When connected to an external power source, the vehicle should be able to recharge its batteries.

3.3 Power System Components

3.3.1 Power Supply

Power to the onboard systems is provided either via an external power cable, or by two on-board nickle-cadmium rechargable battery packs. External power is used when the system is at rest, and the computer is required for software experimentation. When the system is free-floating, external power is not connected and power is provided by the batteries. Batteries may be replaced one at a time, without interruption of power to the experiment's subsystems, thus allowing discharged batteries to be replaced with fully charged batteries. When connected to external power, onboard battery chargers can recharge batteries onboard the experiment.

3.3.2 Power Management and Distribution

Power is controlled either by the experimentalist or by computer control via the power control unit (PCU). It has the capability of switching power onto the system's power bus from the power sources. Multiple sources can be switched on at the same time thereby ensuring continuous power supply in the event of the removal of a source such as unplugging external power or a battery pack. The PCU has sensors which allow the computer to read the status of the power system, as well as override functions, which allow the computer to automatically switch batteries on and off the power bus, allowing a dead battery to be switched out while a new battery is being switched in. The power provided by the PCU is unregulated ±12VDC, which is used for the motor drivers and two sets of power converters/conditioners. One set of power converters is used to provide power to the computer and the other for the analog electronics section (e.g. sensor electronics, etc.), allowing the computer to have an isolated power supply. Load fluctuations and power spikes on the analog system's power busses will not be seen by the computer.

3.4 Status

The system power bus has been wired, as has the regulated power distribution system. The power control unit (PCU) has been constructed and tested in the circuit. Uninterrupted power delivery to the computer system has been demonstrated when removing battery packs and switching from external to on-board power.

3.5 Further work

The PCU requires modification in order to reduce its sensitivity to the power surges seen by the system when power is initially turned on. This changes will not affect the operation of the system as described in this document. The completion of the power system allows us to place a computer, sensor electronics, and motor drivers on board the free-floating experiment without the need for a power cable which could affect its motion. More sensor and actuator electronics needs to be assembled and tested for the experiment prior to the commencement of experimentation.

~

Chapter 4

Navigation and Control of Free-Flying Space Robots

Marc Ullman

4.1 Introduction

This chapter summarizes the progress to date in our research on global navigation and control of free-flying space robots. This work represents one of the key aspects of our comprehensive approach to developing new technology for space automation. Ultimately, we envision groups of fully-self contained mobile robots making up the core work force in space.

4.1.1 Motivation

Although space presents us with an exciting new frontier for science and manufacturing, it has proven to be a costly and dangerous place for people. Space is therefore an ideal environment for sophisticated robots capable of performing tasks that currently require the active participation of astronauts.

While earth based robots have not always proved to be cost effective solutions to manufacturing inefficiencies (due to the abundance of cheap labor), the tremendous cost associated with putting men in space, especially when EVA is required, makes the economics of robots in space particularly attractive.

4.1.2 Research Goals

The immediate goals of this project are to:

- demonstrate the ability to simultaneously control rol.ot base position and arm orientation so that a free-flying robot can navigate to a specified location in space while manipulating its arm(s).

- demonstrate the ability to capture a (possibly moving) free-floating target "on-the-fly" using the manipulator arm while the base is in transit.
- provide a suitable platform for the eventual addition of A.I. based path planning and obstacle avoidance algorithms which will enhance the robustness of task execution.

4.1.3 Background

This work emphasizes the modeling of robot dynamics and the development of new control strategies for dealing with problems of:

- a non-inertially fixed base (i.e. free-floating base)
- redundancy with dissimilar actuators
- combined linear and non-linear actuators
- highly non-linear dynamics
- unstructured environments

Our laboratory work involves the use of a model satellite robot which operates in two-dimensions using air-cushion technology. We have developed a series of satellite robots which, in two dimensions, experience the drag-free and zero-g characteristics of space. These robots are fully self-contained vehicles with onboard gas supplies, propulsion, electrical power, computers, and vision systems. The latest generation of robots is also equipped with a pair of two-link arms for acquiring and manipulating target objects.

4.2 Summary of Progress

The analog subsystem hardware described in our previous report is now nearly complete. We have designed and fabricated several custom printed circuit boards to implement battery monitoring, switching, and charging; safety cutout[1]; sensor conditioning and interfacing; and motor drivers. A more detailed description of the onboard power system can be found in Chapter 3.

During the past report period we made an important change in our laboratory and onboard computer system strategy and implementation. This change is reflected in the sections on Real-Time Development System and Experimental Hardware that follow.

4.3 Real-Time Development System

Following an industry-wide (as well as a NASA wide) trend, our entire laboratory has elected to migrate to a network of Sun Microsystems diskless workstations for system design and analysis as well as for software development, testing, downloading, and debugging. The

[1]Although the safety card has been designed, it is still awaiting fabrication

network environment, based on an Ethernet LAN, couples our computers together so that all software and data can be shared transparently. This facilitates technology transfer and enhances information sharing among the many different projects in our laboratory.

One of the key factors that motivated this decision was the availability of a new realtime development system know as VxWorks.[2] VxWorks is a software package that allows UNIX-based host systems (e.g. Sun Workstations) to be coupled to standalone target systems via an Ethernet or fiber optic LAN. It offers a complete TCP/IP implementation and will ultimately support Sun's Network File System (NFS) so that each target realtime system simply appears as an additional node or client on the host network.

This capability makes several new operations possible. Realtime software can be written, tested, and debugged using the powerful set of development tools available under Sun UNIX and then executed on the realtime system. An optional debugging tool, dbxWorks, provides real-time windowed source level debugging of code executing on the target system via a remote ptrace facility. Data can be sent to/from the host system using UNIX sockets, Sun's Remote Procedure Call (RPC) and eXternal Data Representation (XDR) mechanisms, or transparently via remote file I/O.

Our previously proposed software and hardware solution (QNX on 80386 based platforms) also gave us many of these desirable capabilities, but did so at the cost of compatibility with the "UNIX world." In finding a UNIX alternative, it was essential for our satellite robot work to be able to couple a remote target system to a host with nothing more than a thin fiber optic cable. This technology provides us with a very high bandwidth communications path without the inherent problems of radio link systems while introducing minimal disturbances to the drag-free, gravity free dynamics we are trying to simulate.

4.4 Experimental Hardware

4.4.1 Real-Time Computer

In light of the decision to switch to a UNIX environment, we will be using onboard computer systems employing the VME bus architecture and 68020 or 68030 microprocessors. We have selected a new microcomputer from Motorola (introduced in January), the MVME 147, as our basic onboard computer. This single-board, dual-height VME card features a 68030 microprocessor running at 20 MHz (25 MHz versions will be available later this year) and an associated 68882 Floating Point Coprocessor running at the same clock rate. In addition, it features 4 MB of dynamic RAM, an Ethernet controller, a SCSI bus interface, four serial communications ports, a Centronics parallel interface, and a complete VME bus controller. We have been quoted delivery in late May and look forward to receiving this impressive piece of hardware. With a performance rating of four to six VAX mips it should provide ample onboard computing power.

[2] VxWorks™ is a product of Wind River Systems, Emeryville, CA

4.4.2 I/O Interface Modules

We have ordered the following three VME boards for interfacing the control computer to the vast array of sensors and actuators that the robot is equipped with.

- Xycom XVME 290/1 Digital I/O Module: This board has 32 digital I/O lines and will be used for controlling the thruster solenoids as well as for enabling and disabling batteries, safety cutout circuits, and other devices.

- Xycom XVME 590/3 Analog Input Module: This board supports 16 channels (expandable to 32) of 12 bit analog input with a acquisition time of $25\mu s$. It will be used to read the position and velocity informatio' 'rom our RVDT sensors as well as other analog sensors the vehicle has (e.g. rate gyros, gas pressures, battery voltages, etc.).

- Xycom XVME 595/1 Analog Output Module: This board is equipped with four output channels each of which is controlled by a 12 bit DAC. It will be used to issue torque commands to the manipulator arm motor drivers.

4.4.3 Onboard Computer System Packaging

The computer layer or upper layer of our robot design which was briefly described in past reports has be modified to accommodate a pair of 5 slot VME card cages. These card racks will be mounted horizontally facing opposite directions. The two backplanes feature external termination which allows them to be coupled via a 96 conductor ribbon cable with DIN connectors on each end. The CPU and its associated I/O transition module will be placed in one card cage while all of the analog and digital I/O interface boards will be placed in the other card cage.

4.5 Summary

Decisions that should have a major impact on the long term success of this project have been made during this report period. The selection of a very promising development environment along with a neatly packaged state-of-the-art real-time computer system is a significant step toward a fully operational system that is easy to use for software development and testing. At the same time, work on the robot has be progressing smoothly to the point where it is starting to look like a complete system with fully integrated electrical and mechanical subsystems. The modular design philosophy which has been a guiding principle for this project since its inception is proving to be a very successful one. It is very easy to disassemble the robot module by module for construction and servicing.

4.6 Future Work

We look forward to receiving the VxWorks development system and all of the real-time computer equipment described above. By the end of the next report period, the robot should have an operational onboard computer. With the Ethernet communication link between the onboard computer and our Sun workstations and we will be able to begin downloading and testing software on the robot itself.

$\sim$

Chapter 5

Locomotion Enhancement via Arm Pushoff (LEAP)

Warren J. Jasper

To perform complex assembly tasks, an autonomous vehicle needs to move from one place to another. The use of propellants may not be ideal because of cost and safety factors. Also, the use of thrusters may disturb the environment by impacting a target which the robot is trying to grasp. Our alternative approach is called LEAP: Locomotion Enhancement via Arm Pushoff. In LEAP, the vehicle pushes itself off from a large space object and "leaps" to the desired resting place or simply "crawls" along an object. This is the common mode of locomotion used by the astronauts while in the Space Shuttle. This new project was added to investigate the problems and issues involved in autonomous space locomotion. The first phase of the project involves: devising the experiment, deriving the equations of motion and candidate control laws, and then simulating the model to size physical parameters for the actual experiment. The second phase encompasses design and fabrication of the vehicle, while the third phase experimentally verifies the theoretical development. The following paragraphs describe the progress on phase two.

5.1 The Experiment

A new vehicle is being designed to study LEAP. This vehicle should simulate the motions that an autonomous space robot would perform while in the space station or maneuvering out in space. The experiment will consist of the vehicle pushing off a bar located on one side of the granite table, rotating 180 degs, and catching itself by grasping a bar located at the other end of the table. Ideally, one would like to complete this task without the use of thrusters. However, at the point of initial release from the bar, errors in the velocity of the center of mass of the vehicle can only be corrected using thrusters. To enhance the robustness of this approach, thrusters will be incorporated into the control laws for midcourse correction. The following figure shows the robot in three configurations: pushing off the bar, rotating, and catching itself at the other end. By incorporating crawling and

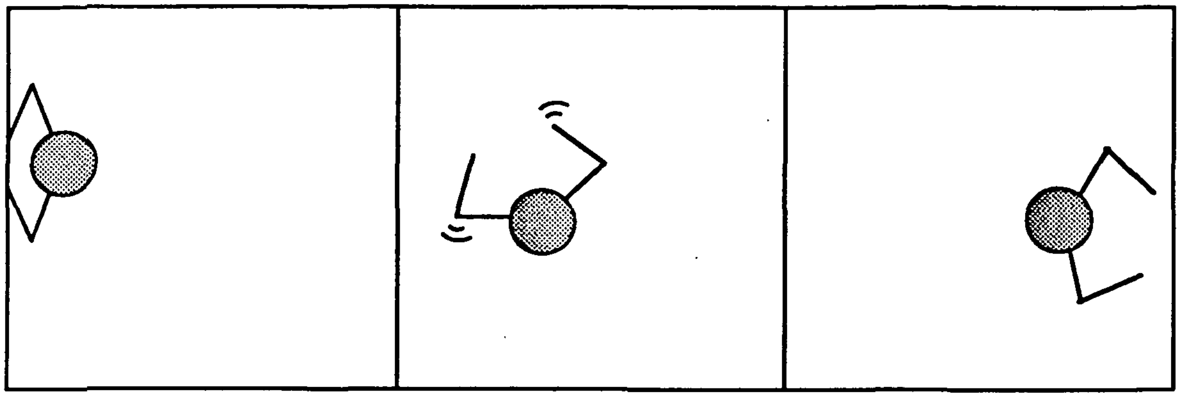

Figure 5.1: The LEAP Demonstration

leaping, the robot can position itself anywhere on the table with a minimum amount of propellant. This investigation complements current work done at the Stanford Aerospace Robotic Laboratory by incorporating global navigation and object manipulation into a general study of locomotion.

5.2 Design of Third Vehicle

The following design modifications were completed on the third vehicle:

- Momentum Wheel Subsystem – A momentum wheel was designed to minimize mass and maximize moment of inertia. A brush DC motor with peak torque of 75 oz-inch drives the momentum wheel, while a turning fork rate sensor measures the angular rate and position of the base body with respect to the inertial frame. The rate and position information will be directly incorporated into the control law. This system is integrated in with the thrusters to make the second layer of the robot the propulsion/momentum management subsystem layer.
- Gripper – The new gripper incorporates an optical triggering mechanism as well as new solid state force sensors. These sensors allow for 0-4 Newton force measurements. Three sensors will be located on each hand to allow for the option of force control. Two of the sensors will be embedded in the palm of the gripper, while the third force sensor will be located in the tip of the gripping finger. The gripper is actuated pneumatically at 50 psi. The gripper is covered with Sorbothane, and energy absorbing material, to reduce the "bounce" problem on impact and increase compliance.

5.3 Future Work

With the design of the vehicle mostly done, the next nine months will involve fabrication and test of the vehicle. A bar needs to be constructed and attached to the granite table to allow the robot to push off the table. In addition to the bar, all the parts for a third vehicle need to be purchased and assembled.

~

Chapter 6

Adaptive Control of LEAP

Roberto Ernesto Zanutta

6.1 Introduction

A major task of free-flying robots is to aid in the construction, maintenance and repair of space structures (e.g. the space station and satellites) while in orbit. Because of the high costs of placing mass into orbit it is desirable to reduce the amount of propellant which the free-flying robots will need in carrying out their tasks. The typical tasks performed by the robots will require them to transport and retrieve various objects. To minimize the amount of propellant required in moving about the robots will have to accurately know the inertia properties of the objects which they carry. Since it is not practical to specify the object mass properties each time a robot performs a task some method of identification is necessary. This can be done through the use of adaptive control.

The investigation of adaptive control for a two-armed free-flying robot is a new project which was begun in July of 1987. The preliminary work has been in the investigation of adaptive control schemes and vehicle modeling and simulation. The preliminary modeling and simulation of the LEAP system was reported in the fifth semi-annual report. The following is a summary of the work which was done on this project during the last six months. Also included is a brief description of future work.

6.2 Literature Search

The literature is rich in adaptive control topics for various kinds of plants including robots. During the past six months much of this literature has been reviewed. Two similar approaches stood out as having the most promise for the aforementioned problem. These were presented by Slotine and Li of MIT[4] and Wen and Bayard[1] of JPL. Their approaches appear both computationally and experimentally desirable.

The two approaches have desirable characteristics for real-time applications. Neither requires estimates or measurements of accelerations as others do. This acceleration information comes at the expense of increased complexity and/or increased cost of the control hardware and software. In addition, both approachs avoid inversion of the mass matrix. The inversion itself is computationally expensive for a complex system. In fact the matrix, when parameter adaptation is being performed, may be singular and therefore non-invertible.

Slotine and Li have experimentally demonstrated their controller on a fixed-base two-link arm. They have shown that in the presence of unknown mass and inertia properties the controller enabled the robot to track a desired trajectory with a small error. Presently there are no simulation or experimental results for the work of Wen and Bayard.

In both cases the control law design emphasis is on trajectory following, not parameter identification. For space applications it is necessary to have both good trajectory following in the presence of unknown object parameters *and* good identification of these parameters. In addition, both approaches assume there are no closed-loop kinematic chains and there is one actuator per degree of freedom. Neither of these assumptions is valid for a two-armed robot holding an object or holding on to the side of a structure. Typically a two-armed two-link robot will have an actuator for each link. If the robot is holding an object with both arms there will be only three controllable degrees of freedom, but there are four actuators. This is an important consideration in the case of the LEAP project when the robot has to push-off.

6.3 Future Work

The approaches which have been mentioned appear to have much promise, but need modifications because of the kinematic and actuator assumptions. In addition, the parameter identification issue must be addressed more directly. These issues will be investigated in the upcoming months. It will also be necessary to study the real-time implementation aspects of the algorithm. This is an over-riding limitation because of the speed of the real-time system and the complexity of the equations of motion. Ongoing work in the hardware development will continue.

Bibliography

[1] David S. Bayard and John T. Wen. Simple adaptive control laws for robotic manipulators. In *Proceedings of the Fifth Yale Workshop on the Applications of Adaptive Systems Theory*, pages 244–251, 1987.

[2] Nevill Hogan. Impedance control: An approach to manipulation. *Trans of the ASME, Journal of Dyanmic Systems, Measurement, and Control*, 107:1–24, March 1985.

[3] John N. Little and Cleve Moler. *PRO-MATLAB User's Guide.* Sherborn, Massachusetts.

[4] Jean-Jacques E. Slotine and Weiping Li. On the adaptive control of robot manipulators. In *Proceedings of 1986 ASME Winter Annual Meeting, Anaheim, CA*, pages 51–56, 1986.