(117)

The Effects of Vehicle Station Keeping and End Effector Disturbance Compensation on Neutral Buoyancy Teleoperation

by

Michael Simon Valdez

S.B. Aeronautics and Astronautics, Massachusetts Institute of Technology (1990)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

Massachusetts Institute of Technology

May 1993

Signature of Author_____
Department of Aeronautics and Astronautics
May 7, 1993

Certified by_____
Professor Harold L. Alexander
Thesis Supervisor

Accepted by_____
Professor Harold Y. Wachman
Chairman, Department Graduate Committee

The Effects of Vehicle Station Keeping and End Effector Disturbance Compensation on Neutral Buoyancy Teleoperation

by

Michael Simon Valdez

Submitted to the Department of Aeronautics and Astronautics
on May 7, 1993, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

Experiments were conducted with a neutral-buoyancy robot to test whether vehicle station keeping and end effector disturbance compensation significantly affect human teleoperation performance. The vehicle used for experiments, called the Submersible for Telerobotic Astronautical Research, or STAR, is a free-flying underwater telerobot equipped with a three degree of freedom arm, a stereo pan/tilt camera platform, and a vision-based navigation system. Using visual feedback from a fixed onboard camera, test subjects performed a Fitts-type tapping task with the arm while the vision navigator and control system held the vehicle steady relative to a visual reference target.

In the absence of other factors, tight station keeping was clearly desirable as test subjects performed the task significantly better with a steady vehicle. However, tight station keeping makes more demands on the propulsion system, increasing fuel requirements substantially. An alternative is to have the robotic arm compensate for vehicle motion, reducing end effector position sensitivity to vehicle motion. With the arm in effect bearing a portion of the station keeping burden, subjects performed essentially as well whether vehicle station keeping was tight or loose. Since end effector disturbance compensation requires little if any extra arm motion, it provides the benefits of tight station-keeping without the associated cost in fuel.

The implications for space-based operations are twofold. First, the use of arm motion compensation may significantly reduce the mass of fuel that must be launched into orbit to resupply the propulsion system. Second, free-flying telerobots do not need to be designed for high-authority thruster control.

Thesis Supervisor: Professor Harold L. Alexander
Title: Assistant Professor of Aeronautics and Astronautics

Acknowledgments

First of all, I would like to thank my thesis supervisor, Sandy Alexander, for his guidance and wisdom in all things practical and not so practical. I learned more from you than in all the classes I took as a grad student. I sincerely wish you all the best!

To Harald, my partner in crime these last three years. We helped build a most excellent vehicle — hope STAR's destined for the glorious existence it deserves. Thanks for all your help and support, and good luck keeping those vortices under control.

To Ali, friend and musician, musician and friend. These last couple of years have been tough for lots of reasons. Thanks for helping me get through it all.

To fellow Class o' 90 Phi Delts. Andy, we made a great team through six years of Aero/Astro! Thanks for coming to almost all my concerts and recitals. It was appreciated much more than you realize. John and Curt, thanks for all the fun times too numerous to mention.

To all the brothers and friends of Phi Delta Theta, and to all the members of the MIT Festival Jazz Ensemble I've come into contact with. You all made these organizations the most enjoyable part of my stay here at MIT. I'll miss it dearly.

To Holly, good friend and purveyor of fine chocolate chip cookies!

To Dean, mechanical genius and UROPer extraordinaire[3]. My experiments simply would not have been possible without all your help. Thanks a million, and good luck at MIT.

To my fellow LSTARites: Mark (fellow Martha's Vineyard cyclist), Eva, Willy, Dewey, Paul, and Beth. For all their help in the lab and at pool tests.

To Ann. Thanks for being in Lobby 10 at just the right time! I know — karma. It's been wonderful.

To Jen and Letitia. For proofreading the manuscript and offering excellent advice for strengthening it. You'll find that I incorporated a great many of your suggestions. Thanks!

To Kurt and Matt, the elder statesmen of LSTAR (sounds like a sci-fi novel). Thanks for making me feel welcome when I first came into the lab, and ever since. OSC's got a couple of great guys.

To my current and former office-mates in 33-407 — you made the room a fun place to be. Thanks to Stephane (your impressions always crack me up), Kevin (fellow Chicagoan and Bulls fan), Sasi, Errol, Didier, Will, Jeff, Alex, Mark, Tom, Fei, Beth and Reuben. Whew! Hope I didn't forget anyone, and no I have *not* been here that long — it's just a big office. Really.

To the members of the MIT Leg Lab. Thanks Marc, Rob, Tony, and Robert. What a cool lab! Keep up the good work. I've had a great time working with all of you.

And finally, to my family. Thanks Mom and Dad, for all the love and support. You are the two most unselfish people I know, and I only hope I can be as good a parent someday. To Ron and Tina. Never figured out how we could all have so little in common and still get along so well. Thanks for being so supportive all these years.

This material is based upon work supported under a National Science Foundation Graduate Research Fellowship. Any opinions, findings, conclusions or recommendations expressed in this publication are those of the author and do not necessarily reflect the views of the National Science Foundation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Teleoperation refers to the real-time remote control of a robot by a human operator who is in some sense separated from the robot's environment. It provides the means for utilizing human intelligence in accomplishing a task without exposing the operator to surroundings that may be too dangerous or inconvenient. Space is one of the most hazardous environments imaginable, but also one of the most interesting to explore.

The ability to work effectively in space is becoming increasingly important. Aboard the planned US Space Station, for instance, tasks such as repair, maintenance, reconfiguration, inspection, and retrieval will inevitably require frequent attention. Currently most orbital tasks are accomplished via extra-vehicular activity (EVA), but given the associated risks it would be more desirable for an astronaut to send out a free-flying remote-controlled vehicle and perform the tasks from the relative safety and comfort of the spacecraft. Easily foreseeable is a scenario where an astronaut flies a vehicle to the worksite, engages an automatic controller to steady the vehicle, then teleoperates the vehicle's manipulator arm to accomplish a task.

In such a circumstance, the vehicle controller must reject disturbances that are either external or induced by the motion of the arm. In the absence of other factors a rock-steady vehicle is clearly most desirable. However, tight station keeping makes large demands on the robot's propulsion system, requiring strong and frequent bursts of thrust in order to completely and immediately cancel any vehicle motion. Such high authority control requires much more fuel than station keeping the vehicle more

loosely. Given the enormous cost per kilogram of sending fuel or any other payload into orbit, reduced fuel consumption would considerably lower operating costs, particularly when spread over an entire fleet of space telerobots. Moreover, if humans can perform teleoperated tasks essentially as well under loose station keeping, the savings can be realized without any loss in productivity.

If the level of station keeping does make a difference, one alternative is to program the arm to counteract variations in the vehicle's position and orientation. By making end effector position independent of vehicle motion, the arm can help reduce the effects of loose station keeping. Generally, end effector disturbance compensation will require little if any extra motion of the robotic arm. Therefore, such a system may provide the advantages of tight station keeping without the large fuel demands and associated cost.

This thesis investigates how these two factors — station keeping and end effector disturbance compensation — affect teleoperation performance. A representative teleoperation task was developed for human test subjects to perform as part of a two-factor experimental design. The vehicle used for the experiment is an underwater telerobot built by the MIT Laboratory for Space Teleoperation and Robotics (LSTAR). Because the vehicle is neutrally buoyant in both position and orientation, it effectively simulates the behavior of a free-flying robot, thus providing an excellent testbed for space teleoperation experiments.

The particular task used in the experiment was based on Fitts' reciprocal tapping test [3] and involved tapping three rectangular targets in a sequence partially determined by a pseudo-random number generator. Fitts used tapping frequency as the measure of performance. For reasons given in Chapter 4, the data collected for this thesis was more amenable to analysis with tapping period as the measure of performance. Since period is inversely related to frequency, it is an equivalent performance measure.

Preliminary test runs involving this task were conducted to help define the specifics for the main experiment on which all the conclusions are based. The intention was to include five test subjects in the main experiment. Due to scheduling constraints and

hardware problems, only two test subjects completed the main experiment. While this reduced the available data, it still permitted a substantial analysis which yielded statistically significant results.

Chapter 2 describes the STAR vehicle in more detail, including its vision navigation system, manipulator arm, control station, and other subsystems. Chapter 3 discusses control of the robotic arm and the implementation of end effector disturbance compensation within STAR's control software. In Chapter 4 are descriptions of the tapping task, the associated target set-up, the software to detect taps and record data, and the specific teleoperation configuration used for the tests. The chapter also elaborates on the preliminary test runs. Presented in Chapter 5 is the main experiment design followed by a detailed statistical data analysis aimed at identifying significant factors. Finally, conclusions and recommendations for further research are given in Chapter 6.

Chapter 2

Submersible for Telerobotic Astronautical Research (STAR)

The underwater vehicle used for the experiments, henceforth referred to as STAR, was built as a platform for space teleoperation and vision navigation experiments. The vehicle is neutrally buoyant in both position and orientation, and hence exhibits the full six degree of freedom (DOF) capability necessary for simulating the behavior of a free-flying space robot. The one limitation of underwater simulation is that the mass (and viscous drag) of the water is very unlike the near vacuum of space. If vehicle motion is kept slow so that drag effects are insignificant, the buoyant water provides an excellent simulation of the zero-gravity space environment.

This chapter describes STAR's main systems, including structure, propulsion, computers, electronics, vision, and manipulator arm. Also discussed are the STAR control station and its multi-tasking network operating system, QNX 4.0. More information, particularly the detailed requirements which drove the design, can be found in [1]. Reference [2] also contains more information, especially regarding the vision-based navigation system in its original implementation.

2.1 Structure

A schematic rear-view drawing of STAR is shown in Figure 2-1. The body of STAR is a waterproof, unpressurized vessel which serves as the primary structural element of the vehicle. The box is made from $\frac{5}{8}''$ aluminum plates and has overall dimensions $3' \times 2' \times 2\frac{1}{4}'$. Two drawer compartments house the lead-acid batteries for the onboard power supply, and a third houses the onboard electronics. Each drawer slides out for convenient access, and the drawer doors are sealed with O-rings.

A frame of $1''$ hollow box tubing is attached to the left, right, and back sides of the main body. The rear frame is omitted in Figure 2-1 so that the drawers can be seen. Intended to provide easy reconfigurability, these three structures serve as attachment points for external components and subsystems, such as the thruster motors.

2.2 Propulsion

Eight bi-directional thrusters provide six DOF propulsion capability. Four thrusters mounted on the side-front of the vehicle are used for fore-and-aft (X) translation, as well as pitch and yaw motion. The other four produce left-right (Y and Z) translation, and roll motion.

Physically, the thrusters are electric DC fishing trolling motors with plastic propellors. Since these motors are not designed for STAR's operating depths of up to 40 feet, each motor housing is pressurized to 10 psi above ambient using a modified scuba regulator [2]. Surrounding each propellor is a foam/fiberglass duct to ensure diver safety during testing. The motor and duct are mounted on a plate which in turn is attached to the vehicle frame.

Mounted on the shaft inside each motor housing is a two-channel quadrature encoder module which allows sensing of motor position. In order to read the encoder signals and control the motor appropriately, the National Semiconductor LM629 motion control chip was employed. The LM629 uses proportional/integral/derivative (PID) compensation and allows the user to set the three control gains (K_p, K_i, and

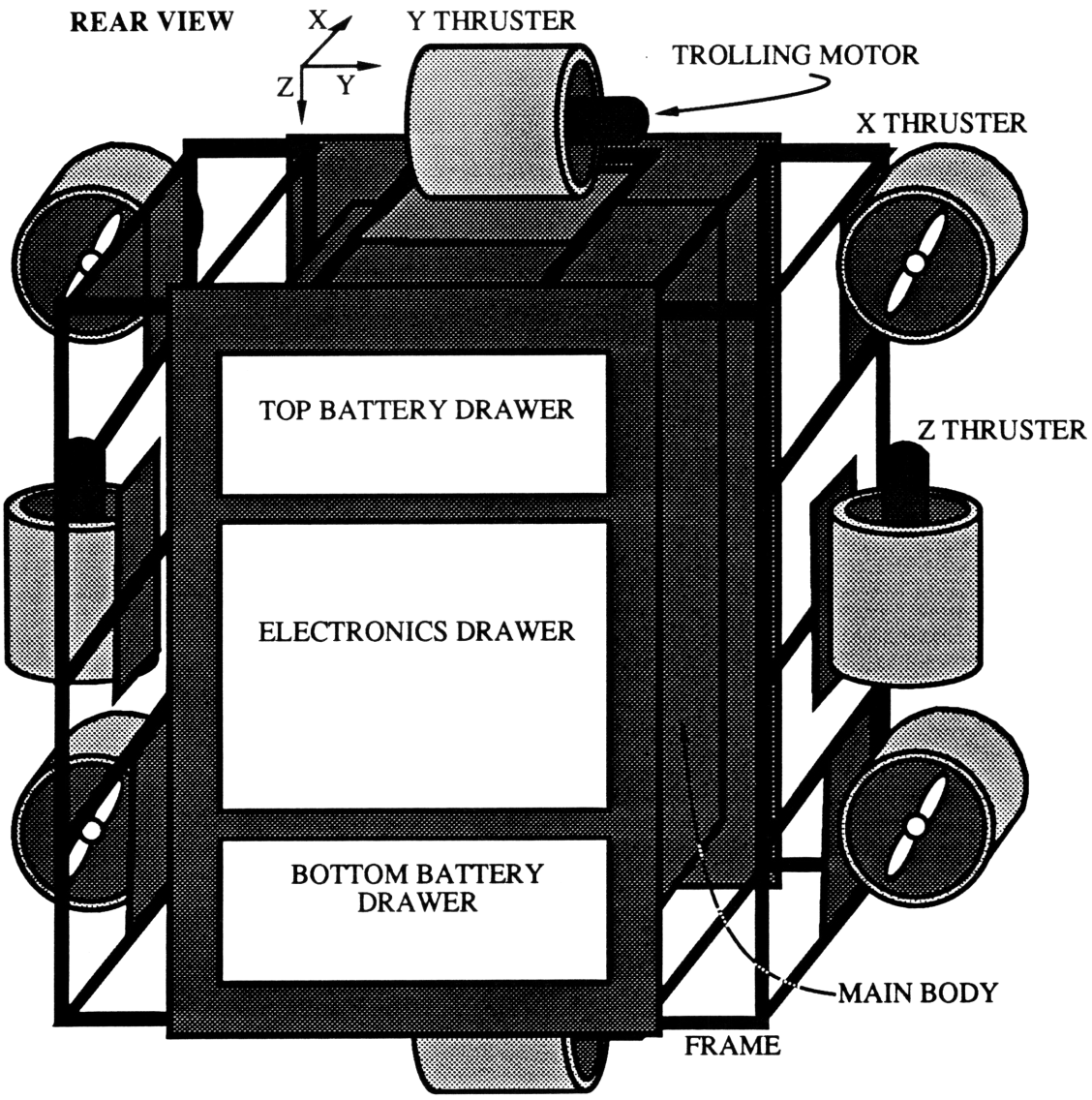

Figure 2-1: **STAR: Schematic Rearview Drawing** *Shown are the vehicle main body, eight propellor thrusters, the left and right frame structures, and the three drawer compartments. Drawing is not meant to be pictorially correct, but rather, accurate in a qualitative sense. Overall dimensions of the vehicle are approximately 6'x5.75'x3', and total weight is 1175 lbs. The fixed onboard camera is mounted at the top-front of the vehicle and is not visible in the drawing. Also not visible is the robotic arm, which is mounted on the front panel 26" below the camera.*

K_d) and command the motor to a specific angular position. For thrusters, the angular velocity rather than position of the motor is of interest, and the LM629 can also regulate the motor to a specific velocity. The STAR software uses this velocity control mode to command the thrusters.

Since the chip outputs only TTL-level sign and pulse-width-modulation (PWM) signals, H-bridge circuits were built to read the signals and drive the motor at the duty-cycle specified by the LM629's PWM output signal. The actual driver circuit assembly is attached just beneath the mounting plate for each motor, while the controller board containing the LM629s resides within the electronics drawer.

2.3 Onboard Electronics

With the exception of the motor drivers, all of STAR's electronics are housed in the large middle drawer of the main body. Currently, the electronics drawer contains:

- The onboard computer, which is a single-board 80386 PC from Ampro Computers, Inc. There is room for two more similar computers.
- An STD bus, in the standard 19″ rack form, containing all the motor controller boards and A/D converter boards for the onboard sensors. Also in the rack is a medium-power driver card which is used to open and close the relays in the battery drawers. The driver card in conjunction with the relays allows power to the thrusters and the arm to be switched on and off from the control station computer described below.
- A set of DC-DC power converters, which convert raw +24V from the batteries into conditioned power for the computer, the STD rack, and the onboard camera.
- An attitude sensor package, consisting of three angular rate sensors from Watson Industries Inc. and three accelerometers from NovaSensor.

- A patchboard through which all electronic signals are routed, allowing more convenient access to relevant signals and easier reconfiguration of connections.

2.4 Control Station

The control station provides the necessary interface between STAR and its operator. This section describes the control station computer, the various control input devices, and the QNX operating system, which enables seamless networking with the onboard computer.

2.4.1 Computer

The control station (CS) computer is a Gateway 20MHz 80386 PC equipped with a SCSI hard drive running the QNX 4.0 operating system. A PC74 digital I/O board from Industrial Computer Source allows reading external signals, such as potentiometer outputs, as well as computer control of external devices, such as the target-light apparatus described in the next chapter.

The CS computer also does the image input processing for the vision navigation system. The Overlay Frame Grabber (OFG) board from Imaging Technologies digitizes the analog signal from the onboard camera and places it in memory, making the image available to the computer as a two-dimensional array of pixel brightness values. The board also routes the picture to a video monitor so the operator can see the "grabbed frame". As discussed in the next chapter, the test subjects were not provided with video feedback directly from the camera, but were instead provided with the same sequence of frames grabbed by the OFG.

The one other expansion card is the CorNet network card. Both the CS computer and the onboard computer are equipped with a CorNet card as required by the QNX operating system. Connecting the two CorNet cards, and therefore the two computers, is a 93Ω coax umbilical which runs out from the electronics drawer to the CS computer.

2.4.2 QNX Operating System

Developed by Quantum Software, QNX 4.0 is a real-time, multi-tasking, message-passing network operating system for PC-type computers. It is a kernel-based operating system where every running task — from a shell to a device handler to a user-written program — is simply a process which can communicate with other processes by sending and receiving messages. From any given computer, or "node", in a QNX network, the entire network looks like a single computer with storage devices and I/O devices on every node readily available. Users on one node can run programs and access files or devices on any other node without having to login to the remote node. They can also assume control of another node, using their monitor and keyboard as if they were sitting at that remote node.

With regard to STAR operation, QNX allows the CS computer and the onboard computer to cooperate seamlessly in running the two interacting programs which control STAR. A description of the control software is given in Section 2.6.1.

2.4.3 Control Input Devices

The STAR control station has four input devices which allow an operator to control the vehicle.

Two large displacement-type joysticks provide control for six DOF motion — one joystick for translation and the other for rotation. In the context of this thesis, the joysticks were used only to fly the vehicle to the testing site in order to engage the vision-based station keeping.

A three DOF master arm provides the interface for real-time control of STAR's robotic arm, described in Section 2.7. The master arm is relatively specific to this thesis and is therefore described in the next chapter on robotic arm control.

A six DOF head tracker is used to sense the orientation of the operator's head. The resulting readings are used to command the stereo pan/tilt camera platform to mirror the head movements of the operator. In conjunction with a helmet-mounted display, this system allows the operator to "look around" in the vehicle's environment.

The pan/tilt platform, the head-tracker, and the helmet-mounted display were not used in the experiment.

2.5 Vision-Based Navigation System

A major goal for STAR since its inception was the implementation of a real-time vision-based navigator. The basic purpose behind a vision navigation system is to determine the motion of the vehicle from video images of the vehicle's environment. For instance, given that a vehicle's onboard camera is pointed at an object, movement of the object left in the video image indicates that the vehicle itself is moving right relative to that object.

Weigl describes STAR's original vision system in [2]. Henceforth, the "position and orientation" of the vehicle relative to some fixed origin will be referred to as the vehicle "state". Given a nominal vehicle state, the system uses images from STAR's onboard camera to detect any deviations from that state. The system is outlined here for the reader's benefit, but the details are beyond the scope of this thesis and are found in reference [2]. The actual vision navigator used for the experiments is an improved version of Weigl's system and is briefly described later. The reader should also note that it is the tightness or looseness of the vehicle station keeping, as opposed to the specific navigation method, that was important to the experiment.

The two basic steps in vision-based navigation are feature tracking and vehicle state computation. Any navigation system requires a fixed reference to determine vehicle state. Feature tracking involves locating visual features in the environment and keeping track of where they appear in the video image. Vehicle state computation involves using these feature locations to calculate the position and orientation of the vehicle relative to the fixed reference.

2.5.1 Feature Tracking

The "object" used as a fixed reference by STAR's vision system is a specially designed target consisting of three white squares on a black background, as shown in Figure 4-5.

Visual features to be tracked on this particular target are corners or edges of the squares.

The high contrast between black and white on the target allows the edges to be found by a simple thresholding technique. If the brightness value of a particular pixel is above a certain threshold, then the pixel is assumed to be white, otherwise it is assumed to be black. The system scans from pixel to pixel until it detects a transition from a white pixel to a black pixel, indicating the location of an edge.

2.5.2 Vehicle State Deviation Computation

For any specific vehicle state relative to the vision target, a given incremental change in the vehicle state from the nominal causes an incremental change in the location of a square's edge in the video image. For example, if STAR moves slightly to the right, vertical edges on the target will appear to move left in the image provided by STAR's vehicle-fixed camera. These relationships may be consolidated into the equation:

$$\delta\vec{E} = \mathbf{H}\,\delta\vec{V} \tag{2.1}$$

where $\delta\vec{E}$ is a vector of the incremental changes for each edge-point being tracked, and $\delta\vec{V}$ is the vector of deviations from the nominal state $[\delta x\ \delta y\ \delta z\ \delta\theta_{roll}\ \delta\theta_{pitch}\ \delta\theta_{yaw}]^T$. Henceforth, the deviation of the vehicle from the nominal state will be referred to as vehicle state deviation. The **H** matrix, or sensitivity matrix, relates $\delta\vec{E}$ and $\delta\vec{V}$ to first order about a specific nominal state. In short, the elements of **H** are partial derivatives of feature point image plane coordinates $(X_{feature}, Y_{feature})$ with respect to the six vehicle states x, y, z, θ_{roll}, θ_{pitch}, and θ_{yaw}. They are derived from geometry and are functions of the camera parameters (e.g. focal length) and the nominal state. Given the six values which define a specific nominal state, the numerical **H** matrix for that state can be calculated.

The functional details of the sensitivity matrix are presented in [2]. Here we assume **H** is calculated for some nominal state. Using **H**, we can estimate the vehicle state deviation using the edge deviations as detected by the vision system. From

Equation 2.1:

$$\delta\vec{V}_{est} = \mathbf{H}^{\dagger}\,\delta\vec{E}$$

$$\mathbf{H}^{\dagger} = (\mathbf{H}^{T}\mathbf{H})^{-1}\mathbf{H}^{T} \tag{2.2}$$

where $\mathbf{H}^{\dagger}$ is the so-called pseudo-inverse of $\mathbf{H}$. $\mathbf{H}^{\dagger}$ allows calculation of the least-squares solution for $\delta\vec{V}_{est}$ when the edge deviations $\delta\vec{E}$ are "known". In other words, Equation 2.1 can be thought of as an overdetermined system of linear equations in the six unknown variables, $\delta\vec{V}$.

The vehicle state deviation estimates $\delta\vec{V}_{est}$ are used, along with readings from the angular rate sensors, as feedback to the station keeping control code which implements a proportional/derivative (PD) compensator for regulating vehicle motion. STAR's control system implementation is described in Section 2.6.2.

2.5.3 Actual Vision System

The vision system used for the experiment in this thesis is more sophisticated version of the system desribed above. Programmed by Professor Harold Alexander, director of LSTAR, the new system differs in two major ways. First, to find edges, the new system chooses its scan lines based on where the edge was last found, making it much more robust to large vehicle motions. Weigl's system used scans with fixed starting and ending points in which case large motions were more likely to cause an edge to move out of the scan range. Second, the new system does not directly use the edge deviations ($\delta\vec{E}$) to estimate vehicle state deviation. Instead, it uses the edge deviations to locate the sides of each square, then finds the intersection of those lines to get the locations of the corners. In other words, the new system tracks the corner as a feature, while the old system tracked the edge as a feature. Thus, unlike the $\mathbf{H}$ matrix in the previous formulation (Equation 2.2), the new $\mathbf{H}$ relates corner deviations to the state deviation, and we can write

$$\delta\vec{V}_{est} = \mathbf{H}^{\dagger}\,\delta\vec{P} \tag{2.3}$$

where $\delta\vec{P}$ is the vector of corner location deviations and $\delta\vec{V}_{est}$ is again the estimated vehicle state deviation[1]. In summary, the location of target feature points (corners of the white squares) in the image plane depends on the specific position and orientation of the STAR relative to the target, and the sensitivity matrix **H** describes this geometric relationship for a given nominal state. The vision system detects small changes in the image-plane position of the feature points (feature tracking) and works backwards through the sensitivity matrix to estimate small changes $\delta\vec{V}_{est}$ in vehicle state relative to the nominal (state deviation computation).

2.6 Vehicle Control

The vehicle control system implements a digital compensator using vehicle position and orientation feedback from the vision system, and vehicle angular rate feedback from the onboard sensors. This section discusses the relevant elements of the control software and the implementation of the digital compensator.

2.6.1 Software

The control software consists of two "C" programs running concurrently: the control station process (CSP) runs on the control station computer and the onboard process (OBP) runs on STAR's computer. Figure 2-2 illustrates the role of each process and how they interact. The CSP is responsible for:

- Various initialization routines
- Starting up the OBP on STAR's computer
- Reading the master arm inputs
- Executing the function which detects taps, records tapping times, and lights the target indicator bulbs (see Chapter 4)

[1]The estimated state vector would actually be for the state of *camera*. A simple translation is done to get the state of the vehicle. This step was omitted for simplicity.

- Running the vision algorithm, which grabs video frames from the onboard camera, searches for the feature points on the vision target, and calculates the estimated state deviation $\delta\vec{V}_{est}$ of the vehicle
- Sending these estimates from the vision system, as well as the commands from the master arm $\vec{\Theta}_{ma}$, to the OBP

The OBP is responsible for

- Various initialization routines
- Receiving the vehicle state estimates $\delta\vec{V}_{est}$ and master arm commands $\vec{\Theta}_{ma}$ from the CSP
- Computing the thruster commands, from the vehicle state deviation estimates, based on a proportional/derivative compensator scheme
- If arm motion compensation is engaged, calculating the compensation terms $\delta\vec{\Theta}_{comp}$ to be added to the master arm commands (see Chapter 3)
- Commanding the thrusters and the robotic arm

For simplicity, only the modules that are relevant to this thesis are shown in Figure 2-2. Both programs perform other functions that may be important but are not crucial to understanding the experiment. For example, the CSP is responsible for reading the joysticks and sending the commands to the onboard process, but during testing, the joystick commands are ignored since the vision system assumes full control of the vehicle except for the robotic arm.

2.6.2 Control System Implementation

Designing a controller for a six DOF plant (STAR) with eight inputs (thrusters) is a very difficult multivariable control problem. It requires a good model of plant dynamics which takes into account the inherent nonlinearities of the system, as well as coupling between the DOFs. Fortunately, STAR's operating envelope encompasses

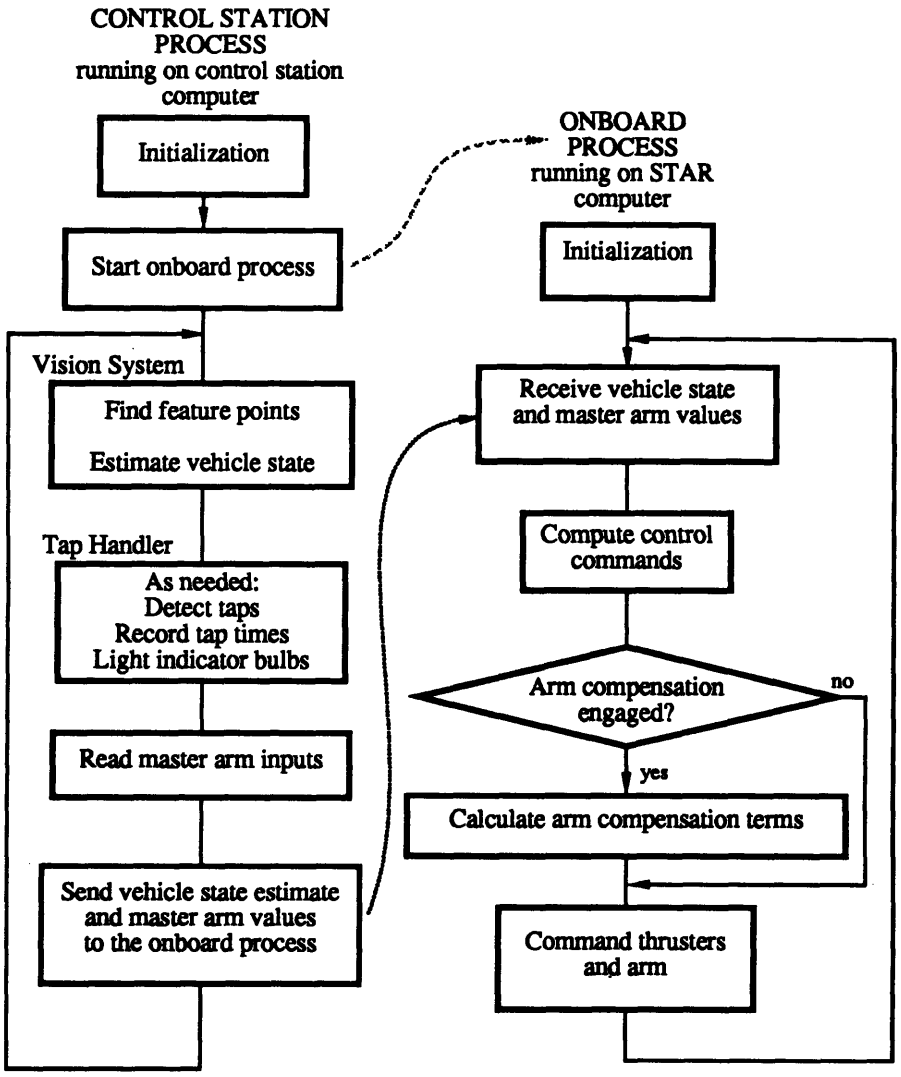

Figure 2-2: **Flow Diagram for STAR Control Software** *The two processes run concurrently and communicate via the message-passing mechanism built into the QNX 4.0 operating system*

mostly small and slow motions so that the system can be treated as linear without significantly compromising control system performance. Another useful simplification is to consider each DOF as decoupled from the others. Experience flying STAR from the control station joysticks suggested that this is a viable first-order approximation [2].

With these two simplifications, a separate compensator can be implemented for each of the six linear one-DOF "plants" — namely, vehicle x, y, z, roll, pitch, and yaw. The proportional/derivative (PD) compensator was chosen over other alternatives for three reasons: it is computationally efficient, it allows for intuitive optimization of control gains, and it does not require an accurate plant transfer function model [2].

Proportional/Derivative Control

In a digital PD compensator design, the control command (u) for the current time step (k) is calculated from the current output (w) and its derivative ($\dot{w}$):

$$u_k = K_p w_k + K_d \dot{w}_k \tag{2.4}$$

where w is a generic variable that could be any of the six vehicle states.

In the case of STAR's control system, the compensator gains (K_p and K_d) for each DOF were found experimentally. Two gain sets were used in the tests: a "high-gain" set that kept the vehicle very steady, and a "low-gain" set that held the vehicle in place very loosely. The "low-gains" were made as small as possible without making the vehicle control so loose that the square targets would go off the edge of the screen, thereby disabling the vision system and terminating the station keeping.

Vehicle Orientation Control

The vision system supplies the three vehicle orientation state deviation estimates ($\delta\theta_{roll}$, $\delta\theta_{pitch}$, and $\delta\theta_{yaw}$ angles), while the the angular rate sensors supply their derivatives ($\dot{\theta}_{roll}$, $\dot{\theta}_{pitch}$, and $\dot{\theta}_{yaw}$). Since there is a direct measurement of both the output variable and its derivative, the PD compensator for each orientation angle is

straightforward:

$$u_k = K_p \delta\theta_k + K_d \dot{\theta}_k \tag{2.5}$$

As mentioned earlier, K_p and K_d for each of the three angles were found experimentally using an iterative process.

Vehicle Translation Control

Unlike vehicle orientation control, there is no direct feedback available for translational velocity. The vision system does not provide linear velocity estimates ($\dot{x}$, $\dot{y}$, and $\dot{z}$), nor is there a separate sensor to measure these quantities. For example, there is no direct measurement of $\dot{y}$ for calculating the control (u_k) for y-translation:

$$u_k = K_p \delta y_k + K_d \dot{y}_k.$$

One option is to calculate the derivative of y from the measurements of δy, as provided by the vision system. However, differentiation of a signal, δy in this example, tends to amplify the measurement noise so that the resulting $\dot{y}$ and calculated control u_k are also noisy, causing the thrusters to twitch rapidly in response to the violent commands.

Therefore, a pole was added to the PD compensator to roll-off K_d, attenuating high frequency noise. The details of this so-called "lead" compensator are presented in [2]. In short, the digital implementation of this compensator results in a more complex control law. For our example,

$$u_k = (K_p + K_d)\delta y_k - (K_p a' + K_d)\delta y_{k-1} + u_{k-1}.$$

where a' is chosen to give the desired roll-off rate. With lead compensation implemented, motor response was considerably smoother.

2.7 Robotic Arm

STAR's robotic arm.is called the Submersible Articulated Manipulator (SAM) and was designed and built in the summer of 1992 by Paul Stach and Dean Franck, then undergraduate assistants at LSTAR. Intended to give STAR basic manipulation capability, SAM has three degrees of freedom, referred to as arm yaw, arm pitch, and arm elbow. In the context of the experiment for this thesis, the robotic arm is the means by which a test subject can accomplish a representative teleoperated task. This section gives some general background on the arm hardware and associated electronics. More details are found in Appendix A.

A schematic of the arm in a nominal configuration is shown at the bottom of Figure 2-3. Also shown is the kinematically identical master arm described in the next chapter. As with all of STAR's external subsystems, the first requirement was to insure full underwater functionality. The motors which actuate the three joints are housed in three identical water-tight aluminum cans, each $3\frac{1}{2}''$ in diameter and $8\frac{1}{2}''$ in length. Interfaces between the cylinder and the endcaps, and between the endcaps and the motor shaft, are sealed by O-rings. The upper-arm and forearm links are made from square aluminum tube stock and both effective link lengths were intended to be 24″. The actual lengths are slightly different and are given in Appendix B, which presents the derivation of the arm's Jacobian. The remaining link, between the yaw and pitch joints, is about $4\frac{1}{2}''$ long. Routed through a pipe fitting on one end of each can, a cable carries six signals to and from the motor: positive and negative motor terminals, +5V, ground, and two shaft-angle encoder signals.

Built into each motor is a quadrature two-channel shaft encoder which measures motor position. As with the thruster motors, the LM629 motion control chip was employed. An H-bridge circuit, of slightly different design than for the thrusters, drives each arm motor. A diagram of this circuit appears in Appendix A.

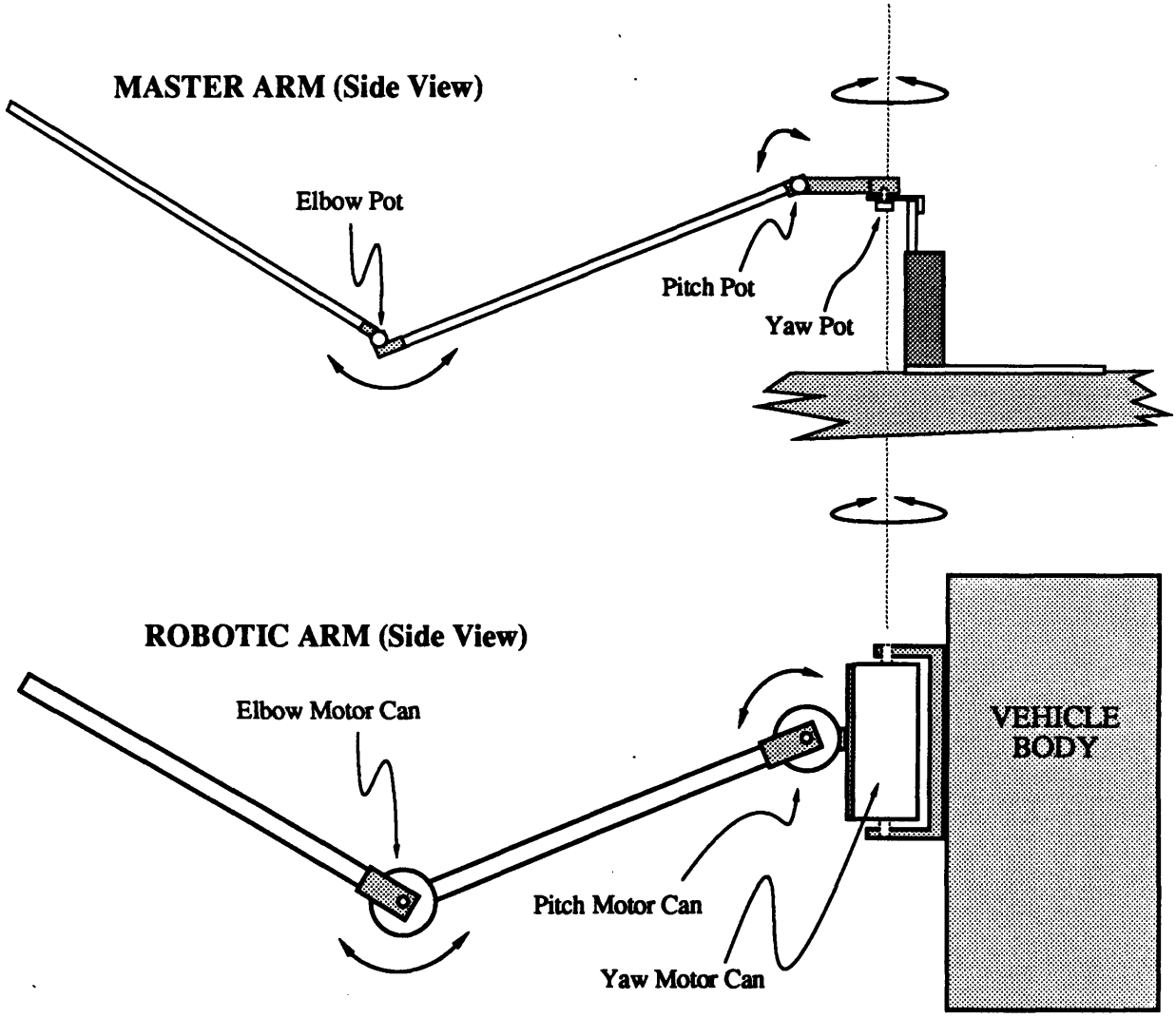

Figure 2-3: **Master Arm and Robotic Arm** *Shown at 1/10th scale. Arms are kinematically identical. Upper arm and forearm links are 24" in length*

2.8 Summary

This chapter described STAR, the free-flying underwater telerobot used in the experiment. Discussed were STAR's structure, propulsion, electronics, control software, vision navigation system, and robotic arm. The intention was to give the reader an overall understanding of the vehicle and its capabilities.

Having provided the necessary background on the STAR, we move to discussion specific to the experiment. Chapter 3 describes how STAR's robotic arm was controlled and how end effector disturbance compensation was implemented. Chapter 4 describes the task performed by the test subjects using STAR as a teleoperated vehicle. Data collected from the experiment is analyzed in Chapter 5.

Chapter 3

Robotic Arm Control

The central focus of the experiment in this thesis is to investigate end effector disturbance compensation, or simply arm motion compensation, as a fuel-saving alternative to tight station keeping. This chapter presents the idea behind this technology and its implementation within STAR's control station programming. Before the concept of arm motion compensation can be understood, the basic method of controlling the arm must first be introduced.

3.1 Standard Control and Master Arm

The variable of interest when controlling a robotic arm is typically the position of the arm's end effector (in our case, the tip of the arm), and the joints are actuated to produce the desired motion of the end effector. In order to teleoperate the arm on STAR, a real-time input device was needed. Called the "master arm", this input device is kinematically identical to the robotic arm, meaning that it has the same link dimensions and joint relationships as the robotic arm.

A schematic of the master arm is shown at the top of Figure 2-3. To match the robotic arm, it has three degrees of freedom: a yaw and pitch joint at the shoulder, and a pitch joint at the elbow. Physically, the master arm consists of three potentiometers joined by aluminum links equal in length to those on the real arm. The yaw potentiometer is attached to a base which sits on the floor beside the operator. By

grasping and moving the tip of the master arm, as shown in Figure 4-4, the operator controls the robotic arm in the familiar master-slave fashion.

The advantage of a kinematically identical controlling device is that the joint angles read from the device need only be mirrored by the corresponding joints on the real arm. If the arms were not kinematically identical, then this simple control scheme would not be possible. The joint angles on the master arm would have to be converted into a corresponding tip position. The set of joint angles on the real arm required to produce the desired tip position would then have to be calculated through inverse kinematics. In addition, since inverse kinematics equations often lack a unique solution, there might be more than one set of joint angles that would produce the desired end effector position, leaving the problem of choosing among them. The master arm described here eliminates these complications, but has the disadvantage of being a bit long and unwieldy due to the size requirements imposed by kinematic identity. However, the size of the master arm did not present any problems with regard to accomplishing the tapping task, which is described in the next chapter.

3.2 End Effector Disturbance Compensation

An alternative to tight station keeping is end effector disturbance compensation, or arm motion compensation. The idea is to program the arm to counteract the vehicle motion so that, for a limited range of vehicle disturbance, the arm tip position in a fixed reference frame remains constant. In order to visualize arm motion compensation, we consider the situation where the arm is locked in some configuration so that arm and vehicle move as a rigid body. The vehicle is at a given position and orientation, then moves to a slightly different position and orientation. Clearly, the new arm tip position depends on how much the vehicle translated and rotated. To get the arm tip back to its original position without moving the vehicle, the joint angles must change in just the right manner. As an intuitive example, if the vehicle moves left, the arm must yaw right and extend slightly to compensate.

This section describes the mathematics and implementation of end effector dis-

turbance compensation. We must first find the relationship between a given vehicle incremental motion and the resulting change in arm tip position. We must also know arm tip position as a function of the arm joint angles. Once we have derived these two relationships, we can use the vehicle state deviation estimates $\delta\vec{V}_{est}$ from the vision system to calculate the new arm tip position, then calculate the change in joint angles necessary to return the arm tip to its original position.

3.2.1 Mathematics

As discussed in the previous section, the joint angles commanded to the arm $\vec{\Theta}_{cmd}$ are ordinarily set equal to the joint angles read from the master arm $\vec{\Theta}_{ma}$:

$$\vec{\Theta}_{cmd} = \vec{\Theta}_{ma}$$

where a vector $\vec{\Theta}$ contains three elements corresponding to yaw, pitch and elbow angles, $\vec{\Theta} = [\theta_{yaw}\ \theta_{pitch}\ \theta_{elbow}]^T$. Engaging arm motion compensation adds a term $\delta\vec{\Theta}_{comp}$ to the above equation that adjusts the commanded arm angles to compensate for the current vehicle deviation from nominal:

$$\vec{\Theta}_{cmd} = \vec{\Theta}_{ma} + \delta\vec{\Theta}_{comp} \tag{3.1}$$

where $\delta\vec{\Theta}_{comp} = [\delta\theta_{yaw}\ \delta\theta_{pitch}\ \delta\theta_{elbow}]^T$. If the vehicle has moved from its nominal state by an amount $\delta\vec{V}$:

$$\delta\vec{V} = \begin{bmatrix} \delta x \\ \delta y \\ \delta z \\ \delta\theta_{roll} \\ \delta\theta_{pitch} \\ \delta\theta_{yaw} \end{bmatrix}$$

then the resulting movement of the arm tip, relative to its original position, is described by the Cartesian vector $\delta\vec{P}_{tip}$:

$$\delta\vec{P}_{tip} = \begin{bmatrix} \delta x - r_{1z}\,\delta\theta_{pitch} + r_{1y}\,\delta\theta_{yaw} \\ \delta y + r_{1z}\,\delta\theta_{roll} - r_{1x}\,\delta\theta_{yaw} \\ \delta z - r_{1y}\,\delta\theta_{roll} + r_{1x}\,\delta\theta_{pitch} \end{bmatrix} \tag{3.2}$$

where $\vec{r}_1=[r_{1x}\,r_{1y}\,r_{1z}]^T$ is a vector from the yaw joint to the arm tip (see Appendix B). The δx, δy, and δz contribution is clear — the arm tip has a translation component equal to vehicle translation. The other terms are due to the rotation of the vehicle, and are derived from geometry using small-angle approximations. To compensate for this motion, the arm tip must move by an amount $(-\delta\vec{P}_{tip})$.

In order to calculate the joint adjustment $\delta\vec{\Theta}_{comp}$ required to produce the arm tip position described by $(-\delta\vec{P}_{tip})$, a formulation based on the Jacobian matrix is used. The Jacobian **J** relates the arm tip velocity in Cartesian coordinates to the arm joint angular velocities:

$$\frac{d\vec{R}_{tip}}{dt} = \mathbf{J}\frac{d\vec{\Theta}_{arm}}{dt} \tag{3.3}$$

where $\vec{R}_{tip}$ is the Cartesian tip position and $\vec{\Theta}_{arm}$ is the vector of joint angles. A detailed functional description and derivation of the Jacobian is given in Appendix B. From Equation 3.3 the following first-order approximation can be made:

$$\delta\vec{R}_{tip} \approx \mathbf{J}\delta\vec{\Theta}_{arm}, \tag{3.4}$$

which describes how small changes in arm joint angles affect the tip position. Therefore, we can invert the equation and calculate the joint angle adjustment as,

$$\delta\vec{\Theta}_{comp} \approx \mathbf{J}^{-1}(-\delta\vec{P}_{tip}). \tag{3.5}$$

In summary, our hypothetical vehicle has moved as described by the state deviation $\delta\vec{V}$ causing movement of the arm tip $\delta\vec{P}_{tip}$. Using the Jacobian and Equation 3.5, we can calculate the joint angle changes $\delta\vec{\Theta}_{comp}$ needed to move the arm tip back to its

original position.

3.2.2 Implementation

Ideally, the Jacobian J should be calculated at each time step, that is, linearized around the latest arm position. For real-time practicality J was instead pre-calculated for a nominal position of the arm. When the arm was far from the nominal position, this formulation was less precise, but still well-behaved and effective. The arm motion compensation software uses vehicle position and orientation data $\delta\vec{V}_{est}$ provided by the vision system to calculate $\delta\vec{\Theta}_{comp}$ from Equations 3.5 and 3.2. When the system is engaged, the control program adds $\delta\vec{\Theta}_{comp}$ to the joint commands from the master arm, as shown in Equation 3.1. The arm motion compensation code was written in the Fall of 1992 by Harald Weigl, then a research assistant at LSTAR. There are three principle C functions which make up the system: `initKin()`, `invKin()`, and `compensate()`. A code listing is given in Appendix B.

The system was checked qualitatively by moving STAR and observing the response of the robotic arm. The commanded joint angles $\vec{\Theta}_{cmd}$ were held constant so that only the compensation term $\delta\vec{\Theta}_{comp}$ affected the arm motion. Various vehicle translations and rotations were commanded from the joysticks and the arm reacted correctly to all of them, successfully keeping the tip position constant.

3.3 Summary

This chapter described STAR robotic arm control using the master arm, and how the end effector disturbance compensation system helps minimize vehicle-induced arm tip motion. The next chapter describes the representative task used to test the effectiveness of this system for space-based teleoperation.

Chapter 4

Teleoperation Task

End effector disturbance compensation, as described in the preceding chapter, is proposed as an alternative to tight station keeping. A very steady vehicle is clearly most desirable for efficient arm teleoperation, but a loose station keeping requires much less fuel. Arm motion compensation may be a way to provide the teleoperation performance benefits of tight station keeping without the associated cost in fuel. To test this technology, a representative teleoperation task was needed for human test subjects to perform as part of a designed experiment. This chapter describes the task and its implementation.

4.1 Task Development and Description

In a 1954 paper, Fitts [3] described an experiment where test subjects performed what he called a reciprocal tapping task. On a table in front of the subject were two narrow targets spaced a certain horizontal distance apart as shown in Figure 4-1. The object was to alternately tap the two targets as quickly as possible, using a pen-like metal stylus. Fitts was interested in how the tapping frequency was affected by the width (w) of the targets and the distance (D) between them, and the relationship he found came to be known as Fitts' Law.

Fitts' task is a generally well-accepted test for human performance evaluation. Given this fact, along with the limitations on STAR's robotic arm, it was natural

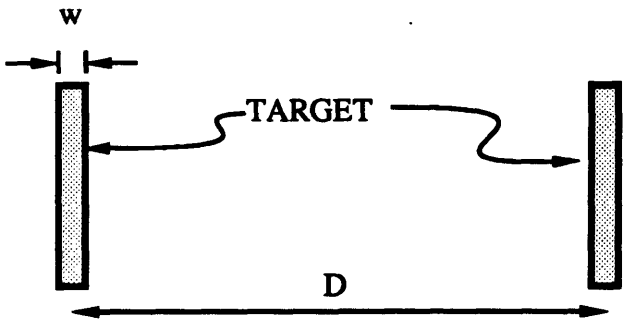

Figure 4-1: **Set-up for Fitts' Reciprocal Tapping Experiment**
Targets were laid out on a desk as shown.

and appropriate to consider the same tapping task for these experiments. However, informal experiments showed that with such a cyclic task, vehicle movement under loose station keeping was too predictable. The steady back-and-forth arm motion induced similar vehicle motion, which was not the erratic disturbance motion that was desired.

It was therefore decided that including some element of uncertainty would be more representative of real space teleoperation. The task devised for the experiment required three rather than two targets, placed in a horizontal row. The subject would first tap the center target and a light bulb would light behind one of the two outer targets. The subject would then tap the indicated target, and return to tap the center target again. Once again a bulb would light to indicate the next target to tap, and the process continued, with the subject always returning to tap the center target to find out which side target to tap next. The computer provided an audible "beep" to inform the test subject that a tap had been detected. Henceforth, a tap of the center target is referred to as a "center tap", and a tap of either side target is referred to as a "side tap".

The number of taps in the task was arbitrarily set to ten so that there were five center taps and five side taps in a given task run — a center tap followed by a side tap followed by a center tap, and so on, ending with a side tap. The task can also be thought of as five consecutive center-tap/side-tap combinations. The time of each

individual tap occurrence was recorded, yielding ten time values per run. As explained in the next chapter, the value of interest was the time interval between each center tap so that a run produced four data measurements, each representing the time it took to tap the center target, tap a side target, then tap the center target again.

The remainder of this chapter describes the tapping target assembly, its associated tap detection circuitry, and the data acquisition software. Also discussed are the exact testing configuration and procedure, and an informal preliminary testing session which helped to certify the hardware, software and procedures, as well as identify some problems to be fixed prior to the main experiment.

4.2 Target Apparatus

The target apparatus used for the experiment was mounted at the bottom of the vision target as shown in Figure 4-2. The black portion of each target is a spring-hinged panel that can swing backward and snap back to its original vertical position. Two bulbs are mounted as shown to indicate which target to tap.

A schematic of the electronic set-up is shown in Figure 4-3. A digital I/O card allows the control station computer to light the bulbs and detect taps. When a target is pushed back, the associated magnetic reed switch closes and the corresponding digital input to the computer is pulled low. To light a bulb, the corresponding digital output is set low, triggering a transistor amplifier to send current through the bulb. Once during each loop of the control program, a function is executed which checks for taps, records the times, and lights the bulbs. A listing of this code and a diagram of the bulb driver are found in Appendix C.

4.3 Experimental Configuration

The teleoperation configuration used for testing is illustrated in Figure 4-4 along with the corresponding underwater set-up. The master arm rests on the floor beside the subject's chair and the video monitor is placed roughly at head height (sitting

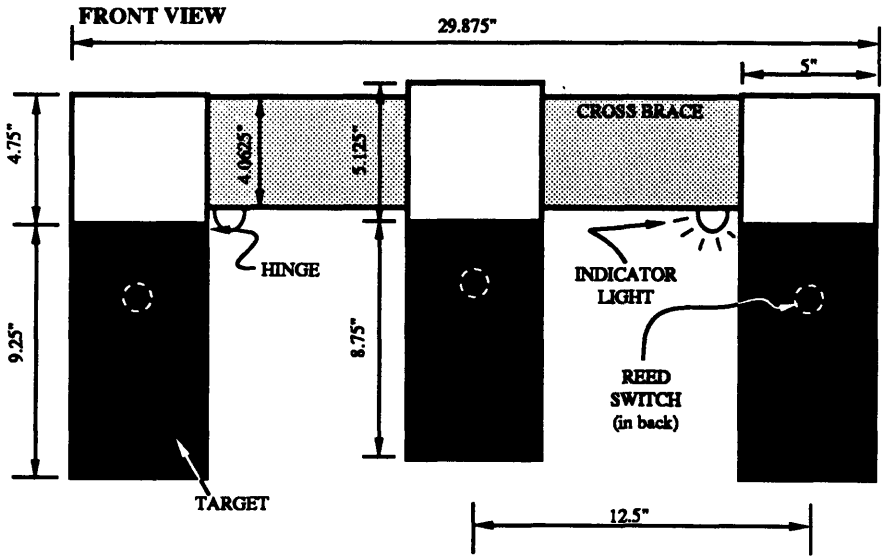

Figure 4-2: **Target Tapping Apparatus** *Shown at 1/10th scale.*

down). It is important to note that the same vehicle-fixed camera is used for both the vision system and teleoperation feedback, so that the image seen by the test subject is updated only as fast as the vision system runs (about ten frames per second [2]). Feedback frequency is certainly a potential factor, but is not considered in this thesis. In any case, the same image frequency was used for all testing runs, thus preserving the statistical integrity of the experiment. Also, the "infrequent" updates did not hamper the subjects' ability to accomplish the task. A typical feedback view as seen by the test subject is shown in Figure 4-5. The camera providing this view is mounted at the top-front of the vehicle and the robotic arm is mounted about 26″ below it.

4.4 Preliminary Testing and Test Subjects

Since testing was conducted at the MIT Alumni Pool, where scheduling was severely limited, it was important to get an estimate of how long the task typically took to complete. This would give an idea of how many test subjects could be used and how many runs they could perform in the available time. It was also important that any

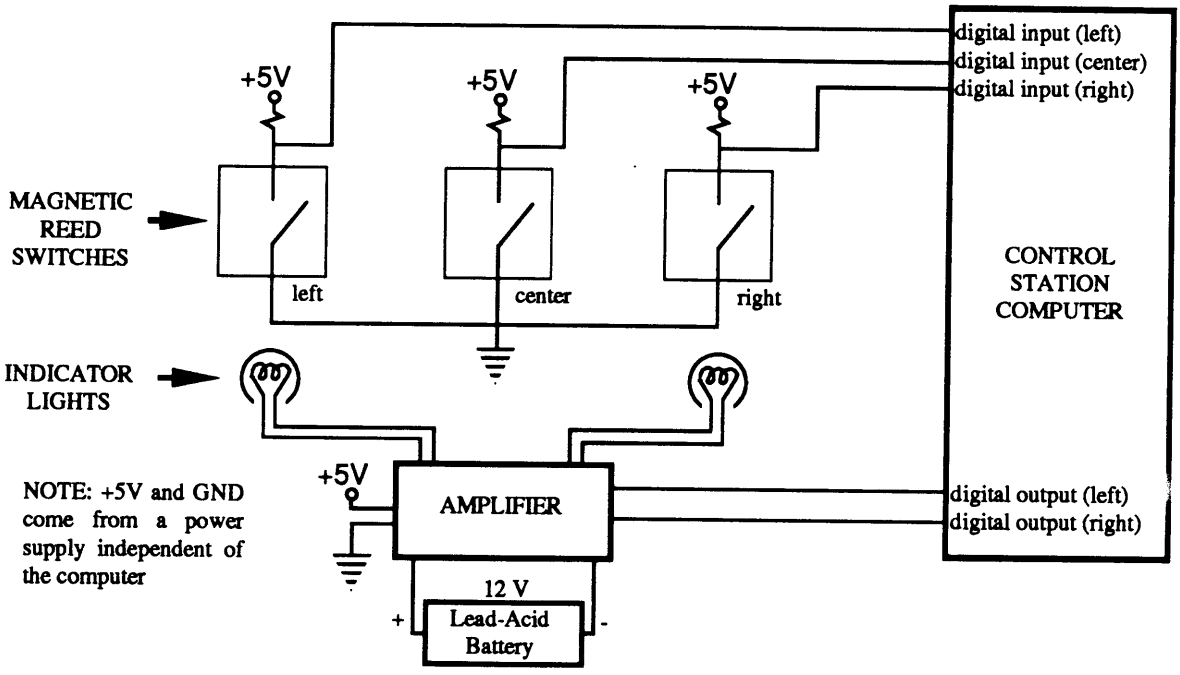

Figure 4-3: **Electronic Schematic of Target Tapping Apparatus**

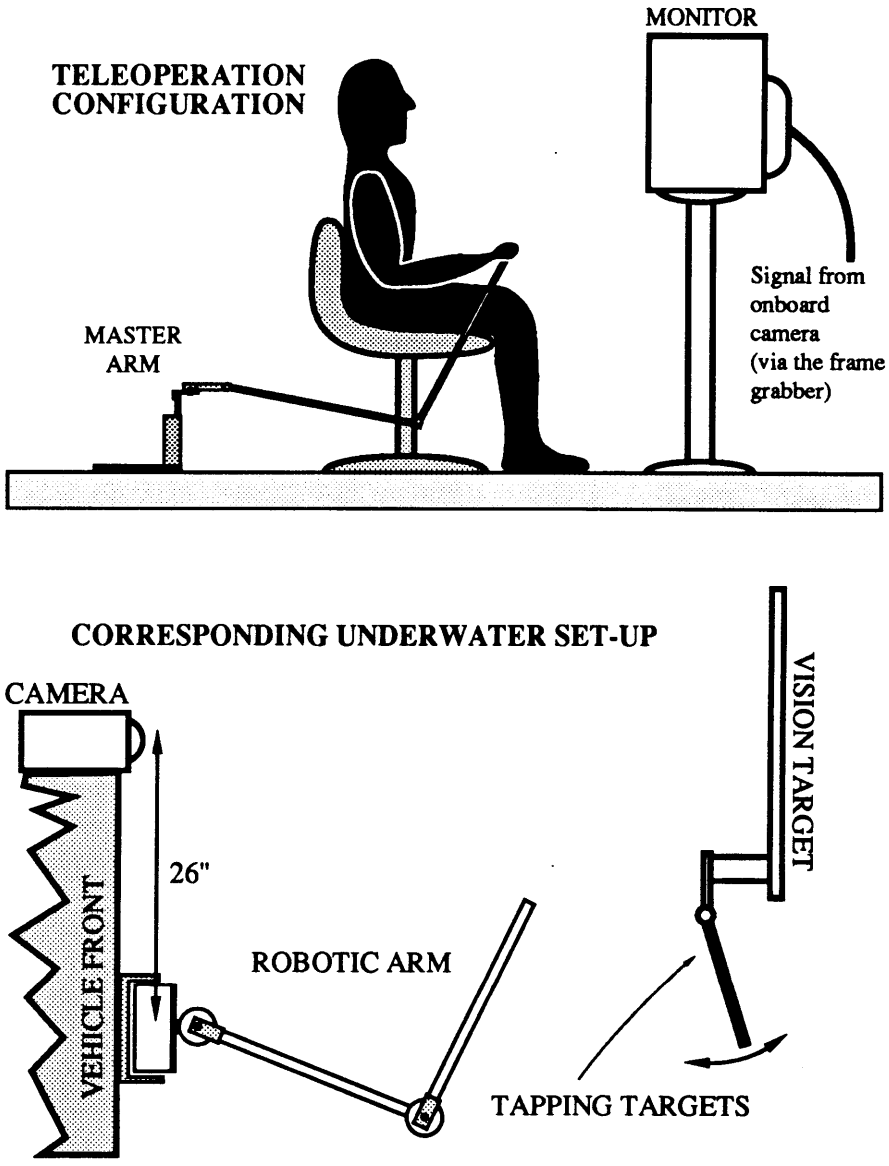

Figure 4-4: **Testing Configuration**

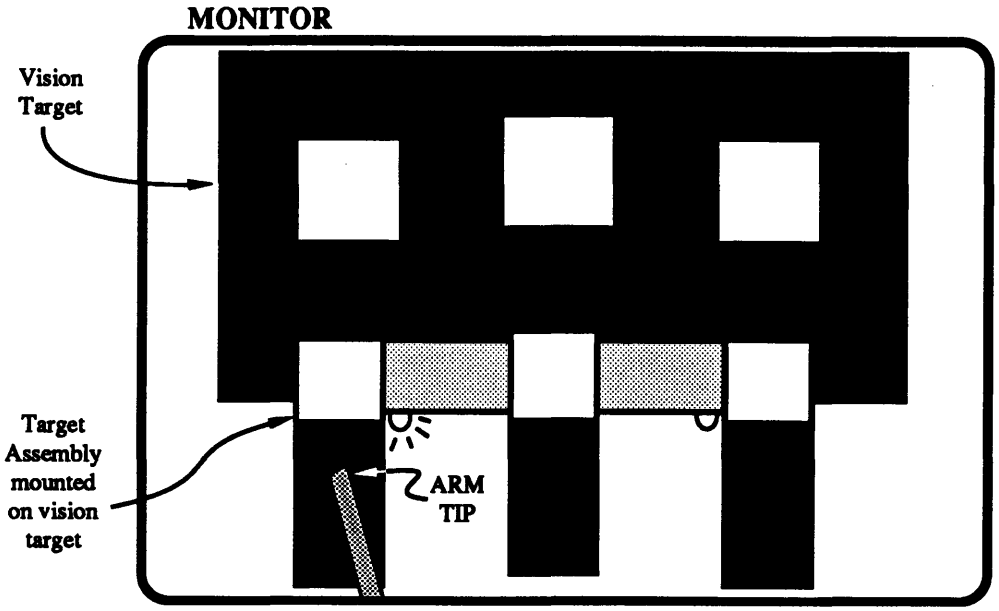

Figure 4-5: **Typical Visual Feedback** *View as seen by the test subject. Nominal vehicle position was 1.4 meters out and 0.9 meters down from the center of the vision target. Robotic arm is mounted 26 inches below the camera lens. The black portion of each target is a hinged panel pushed back by the robotic arm tip.*

software quirks, procedural inconsistencies, or hardware bugs be fixed before running the real experiment. Finally, test subjects needed time to practice the task and move sufficiently up the learning curve. For these reasons, an informal experiment was conducted in which four subjects each performed the task under the four possible test conditions.

When choosing test subjects, it was deemed infeasible to try and capture a complete cross-section of society. With this in mind the decision was made, somewhat arbitrarily, to use right-handed males between the ages of 18 and 25. In making this concession, it was hoped that potential effects due to age group, gender, and handedness would be minimized, and that any statistically significant variabilities in the data could be attributed to the factors in question.

During these preliminary tests, the two sets of vision system control gains were found to be satisfactory. The "tight" set of gains held the vehicle very steady, and the "loose" set was distinctly inferior, allowing the vehicle to move quite a bit during a tapping run. The intent of the tight setting was to simulate a perfectly still vehicle as closely as possible. Arm-induced vehicle motion under tight station keeping was found to be small compared to the arm movements required to complete the task. During particularly violent task runs under loose station keeping, the vehicle moved so much that the vision target squares nearly went off the screen — an indication that the station keeping was as loose as we could make it without intermittently disabling the vision system.

The tests also presented an opportunity to check the functionality of all the systems involved in the experiment. The target switch implementation that had been used in prior testing caused computer glitches which occasionally disabled the vision system. Therefore, they were replaced with the magnetic reed switches described earlier in this chapter. The experiments showed that the reed switches were consistent and reliable. The one other hardware problem revealed by the experiment was a loose linkage on the master arm, which was easily fixed. There was also a bug in the software causing data times to be stored incorrectly. This, too, was fixed after it was revealed during the experiments.

It is clear that there are two experimental factors each with two settings — tight or loose station keeping, and arm motion compensation on or off — giving four different factor combinations. Preliminary data suggested that within the time available, five test subjects could complete the task about 32 times. Since the desire was to collect the same amount of data for each factor setting, it was decided that each subject would perform the task 8 times within each of the four factor combinations. The detailed experimental design and data analysis is presented in the next chapter.

Chapter 5

Main Experiment

To this point, all discussion has dealt with background material. Chapter 2 described STAR, the free-flying neutral buoyancy vehicle used for the experiments, along with its vision navigation system and robotic arm. Chapter 3 discussed end effector disturbance compensation, the potential alternative to tight station keeping proposed in this thesis. The previous chapter discussed the representative teleoperation task used for the experiment, as well as the associated testing configuration, hardware and software.

In this chapter, the main experimental data analysis is described. However there is still a bit more background material required. The next section (5.1) describes the statistical experiment design and the notation used to refer to different factor combinations, or cells. Section 5.2 outlines the exact experimental procedure, and consolidates all the experiment-related information previously mentioned. Section 5.3 discusses a time series analysis of the data. Section 5.4 describes the process used to identify outliers and the justification for removing them. Section 5.5 discusses analysis of variance (ANOVA), the statistical tool used in this thesis. Finally, Section 5.6 provides an overview for the analysis that follows in Sections 5.7 through 5.9.

It is important to note up front that the original intention was to have five test subjects. During the evening of tests — the only available time slot in which the experiments could be conducted — a hardware failure allowed a complete set of runs for only two test subjects. The reader should keep this fact in mind, although the

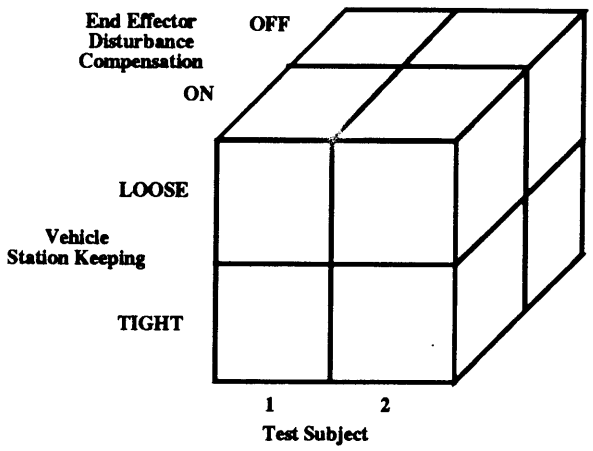

Figure 5-1: **Experimental Test Cells**

data did yield statistically significant results.

5.1 Design and Cell Reference Notation

The two factors tested in this experiment are: station keeping ("tight" or "loose") and end effector disturbance compensation ("on" or "off"). For the sake of analysis, we consider "test subject" as a third factor, also with two settings ("subject 1" or "subject 2"). Hence there are 2^3 possible factor combinations, or cells. Figure 5-1 illustrates how each cell can be thought of as a corner section of a cube. The approach is simply to collect data for every cell, having each test subject complete task runs under all four conditions. As previously mentioned, the intent was to have five subjects perform the task eight times under each test condition, in which case our "cube" would have been five cells wide. However, only two subjects completed all the tests.

Instead of simply numbering each cell, a more indicative notation is desirable. In

this thesis, the following notation is used to refer to cells in the experimental design:

$$\mathbf{C}_{(test\ subject)-(station-keeping)-(end effector\ disturbance\ compensation)}$$

For instance,

$$\mathbf{C}_{2-Loose-On}$$

refers to the specific cell for test subject 2, loose station keeping, and no arm motion compensation. To refer to the two cells for which station keeping is tight and arm motion compensation is off, we use

$$\mathbf{C}_{Tight-Off}$$

Cells for which arm motion compensation is on are referred to as

$$\mathbf{C}_{On}$$

which are the four front cells in Figure 5-1. While this notation is a bit more cumbersome than simply numbering each cell, it eliminates the need to refer back to a diagram since it incorporates all information relevant to the cell.

5.2 Experiment

5.2.1 Procedure

All testing was conducted at the MIT Alumni Pool the evening of January 12, 1993. The task is described in Section 4.1. For each subject the following procedure was used:

- Subject sat down in front of the monitor and grasped the tip of the master arm as shown in Figure 4-4.
- Subject was allowed to move the arm around using the master arm. There is a slight time lag — particularly for large, quick motions — between the master

arm and the robotic arm. For this reason, it was advantageous to have the subject refamiliarize himself with the feel of the arm control.

- As a warm-up exercise, the subject performed the task ten times under the Tight-Off configuration. Both subjects participated in the preliminary tests and therefore had practiced the task many times prior to that evening. These first ten runs gave the subject the opportunity to, in some sense, recall what was learned from previous practice. The purpose was to minimize effects on performance due to a subject "starting out cold".

- Subjects encountered the different settings in the order shown in Table 5.1. For each setting, the subject performed the task eight times, resting between runs when needed. Before every run, the vision system was allowed enough time to steady the vehicle again, and the subject was then allowed to position the arm where desired (usually in front of the center target). A short rest period was allowed after each setting (every eight runs).

- Subjects were encouraged to convey their own observations and perceptions regarding the task, either between runs or after the testing session.

As an example of the procedure, subject 1 first completed the ten warm-up runs, then vehicle station keeping was set to "loose" and arm motion compensation was turned off. The subject then performed the task eight times under this setting. Next, tight station keeping was engaged and arm motion compensation was turned on. Once again, the subject performed the task eight times. The procedure continued in the order given in Table 5.1. These sequences were generated using the table of random digits in [5]. The intent was to have every subject encounter the four settings in a different order. By scrambling the cell sequence for each person, it was hoped that any biases arising from a particular order — for example, every subject doing the Tight-Off runs third — would be minimized. With only two subjects, however, the benefits of this approach were limited since relevant effects (due to the order in which the settings were encountered) were "averaged" over only two different sequences.

	Subject 1	Subject 2
1st	Loose-Off	Tight-On
2nd	Tight-On	Loose-Off
3rd	Tight-Off	Tight-Off
4th	Loose-On	Loose-On

Table 5.1: Test Sequences for Both Test Subjects

The situation improves with more subjects since the effects are averaged over more different sequences.

5.2.2 Data

The time of each individual tap was recorded, yielding ten time measurements (in milliseconds) per run:

$$t_{center1},\ t_{side1},\ t_{center2},\ t_{side2}, \ldots,\ t_{center5},\ t_{side5}$$

Tapping period, being inversely related to frequency, was the desired measure of performance. However, simply calling the time between every tap the tapping period is not valid because a center-to-side tap sequence is fundamentally different from a side-to-center tap sequence. In the former case, the subject did not know beforehand which side target to tap next. In the latter case, the subject always knew to tap the center target next. With this in mind, it was decided that the best measure of performance is the time between center taps. This period incorporates both the "determined" tap phase and the "uncertain" tap phase, yielding a much more symmetric measure of tapping period.

Each run produced four center tap periods, Y_i, calculated as follows

$$Y_1 = t_{center2} - t_{center1}$$

$$Y_2 = t_{center3} - t_{center2}$$

$$Y_3 = t_{center4} - t_{center3}$$

Target Set-up	Three hinged panels in a horizontal row
Task	Using master arm to teleoperate robotic arm: 1) Tap the center target 2) Tap the chosen side target as indicated by the light bulbs 3) Repeat 1) and 2) four more times
Response variable	Time between each center tap in milliseconds
Factors	1) Station Keeping (Tight or Loose) 2) End Effector Disturbance Compensation (On or Off) 3) Test Subject (1 or 2)
Number of Cells	3 factors each with 2 settings $\Rightarrow$ 8 cells
Replications	8 task runs per cell $\Rightarrow$ 32 replications
Data	8 cells, 32 measurements per cell $\Rightarrow$ 256 measurements of the response variable. Five measurements were removed from each cell so that the analyzed data set contained 27 measurements per cell (see text).
Procedure	For the 4 combinations of factors 1) and 2), each subject performed the task 8 times
Test Subjects	Two right-handed males between the ages of 18 and 25

Table 5.2: Summary of Experiment

$$Y_4 = t_{center5} - t_{center4} \tag{5.1}$$

so that t_{side5} is not used. Since eight runs were conducted within a given factor combination, there were a total of 8x4 or 32 period measurements in each cell. In the context of the data analysis, center tap period is the "dependent variable", and each Y is a "measurement" of the dependent variable. For reference, all 256 measurements (8 cells $\times$ $32\frac{measurements}{cell}$) are listed in Appendix E.

As a convenience to the reader, information relevant to the experiment is capsulized in Table 5.2.

5.3 Time Series Analysis

As mentioned in Section 5.2, the two subjects encountered the four settings in the orders shown in Table 5.1. To investigate possible time dependencies in the data, a time series analysis was conducted. The 32 measurements in each cell are supposed to be a random sample — any obvious time dependence would violate that assumption.

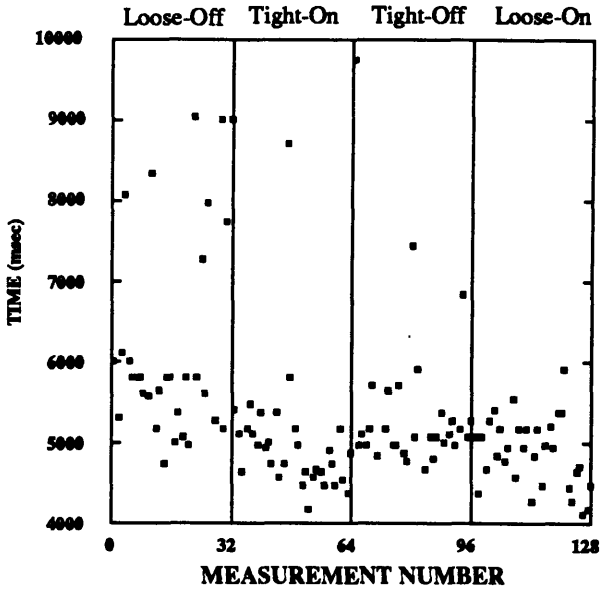

Figure 5-2: **Time Series Plot for Test Subject 1** *Measurements plotted in chronological order. Y-axis limited to 10000 milliseconds to display data at a more appropriate scale.*

For each subject, we plot the center tap period measurements Y in chronological order. Figure 5-2 shows the plot for subject 1 and Figure 5-3 shows the plot for subject 2. Each plot is divided into four regions representing the four different test conditions, or cells. For instance, the first 32 measurements in Figure 5-2 are from the 1-Loose-Off cell since subject 1 performed the Loose-Off runs first. Naturally, those 32 measurements themselves are also in chronological order. The end result is a complete performance history for both subjects.

In examining these plots, we are looking for any patterns, either overall or within a specific cell. For example, the overall plot might show a downward trend indicating that the subject was steadily improving throughout the test. Since a subject should be sufficiently up the learning curve *prior* to the tests, data which shows a downward pattern should not be considered valid. Another related example would be a downward trend within a given cell, suggesting that the subject needed time to get accustomed to the new setting.

The plot for subject 2 shows no apparent patterns. However, the plot for subject 1

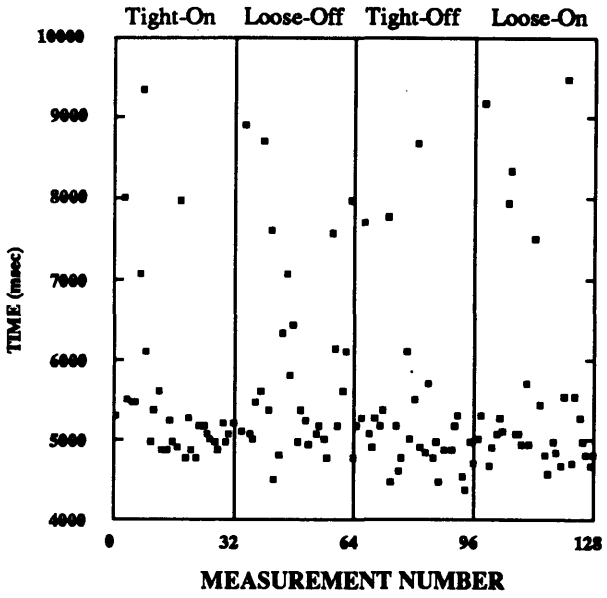

Figure 5-3: **Time Series Plot for Test Subject 2** *Measurements plotted in chronological order. Y-axis limited to 10000 milliseconds to display data at a more appropriate scale.*

shows an obvious downward trend in the first two regions, corresponding to the Loose-Off and Tight-On settings, which is a violation of the random sample assumption. However, the same constraints that limited us to only two test subjects also precluded the possibility of collecting a second set of data for these two cells. Our only option is to proceed with the analysis, keeping this fact in mind. On a more optimistic note, no patterns seem to exist in the next two regions corresponding to the Tight-Off and Loose-On settings, which happen to be the regions of most interest in terms of performance comparison.

Even though these two settings show no patterns, one might also notice that both subjects performed the Loose-On runs last — a random but unfortunate occurrence. Since this thesis focuses on comparing Loose-On performance to Tight-Off performance, it can be argued that there is a bias in favor of the Loose-On setting because those runs benefitted from 24 runs (8 runs x 3 cells) worth of "practice". However, the time series plots show that both subjects reached a generally consistent level of performance prior to the third region. Hence, one can argue that both the Tight-Off

runs (done third) and the Loose-On runs (done last) were conducted when the subjects were sufficiently up the learning curve. Therefore, comparisons between those two settings are valid.

In the time series plots, one might notice data points far above most of the other data points clustered below. Many of these were considered outliers, as described in the next section.

5.4 Outlier Analysis

Outliers were identified with the aid of box-and-whisker plots, or simply box plots [5]. Figure 5-4 shows a separate box plot for each cell. The ends of the box show the 25th and 75th percentiles for the cell, and the distance between them is appropriately called the inter-quartile range. The line inside the box shows the cell median. Values that are more than one-and-a-half inter-quartile ranges from the median are shown as asterisks, and values more than three inter-quartile ranges away are shown as dots. Aside from these outside values, the rest of the data falls within the range depicted by the whiskers, which extend outward from both ends of the box.

A box plot quickly conveys the median of the cell, the range in which the middle 50% of the data resides, and the range of the entire data set, excluding outside values. These features make it an effective tool for identifying outliers. One might notice that all the outside values are above the plots. They are from the runs in which the test subject got the arm stuck behind a target or missed it repeatedly, making the task take much longer. While this could have been an effect of the station keeping or arm motion compensation setting, the vast majority of numbers are within the limits of the whiskers and probably better represent the true performance distribution within the cell.

The box plot for each cell was considered separately. In three of the cells, for example, there are one or two outside values which are well beyond the whiskers. Since they are far from the other thirty numbers, we have good justification for removing them as outliers. In all, twenty numbers were removed as outliers — all the

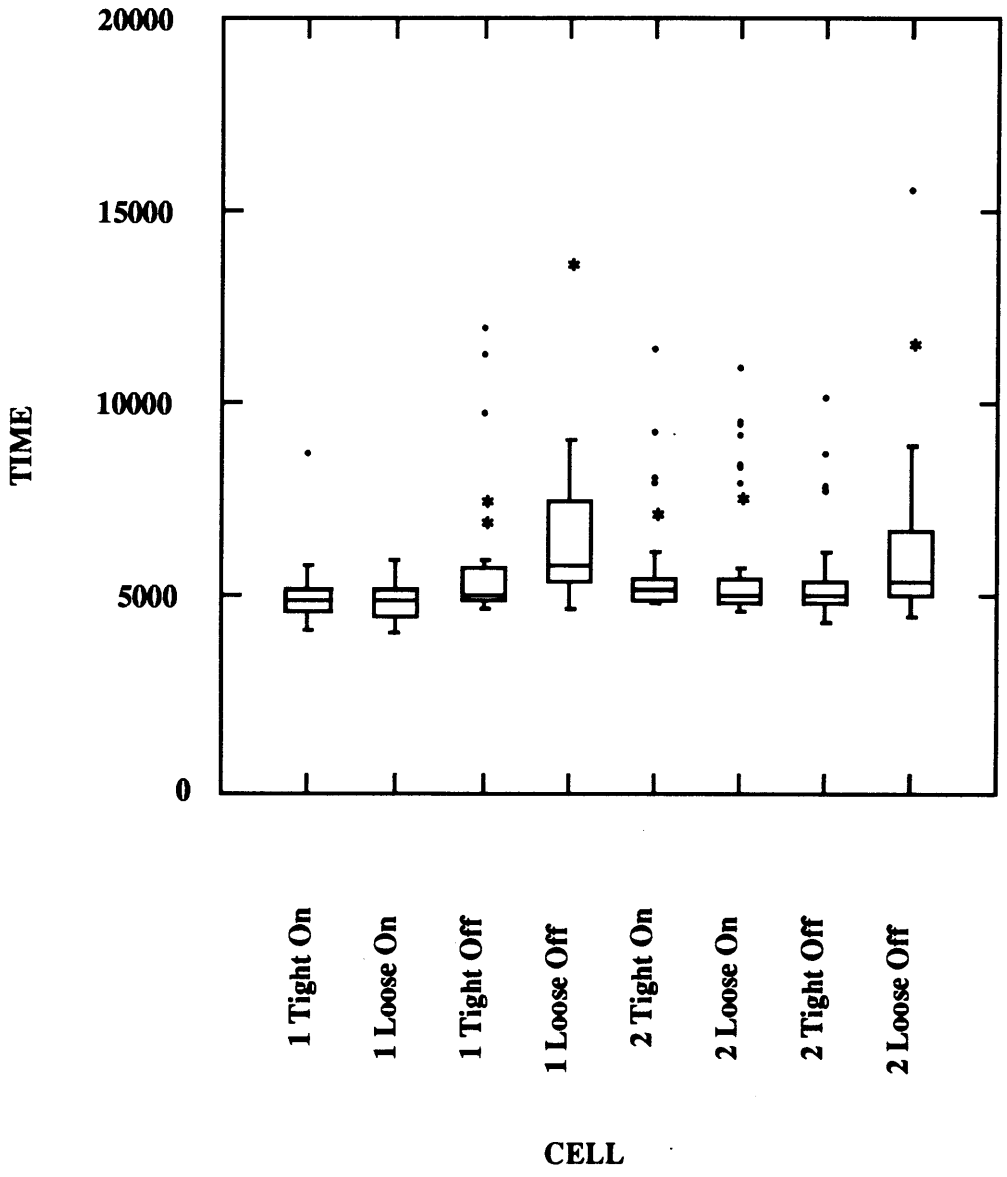

Figure 5-4: **Cell Box Plots** *Dots and Asterisks denote outside values. All dots and some asterisks were removed as outliers. See text.*

Source	SS	*df*	MS	F
A	SS_A	$a-1$	MS_A	$F_A = MS_A/MS_{Error}$
B	SS_B	$b-1$	MS_B	$F_B = MS_B/MS_{Error}$
AB	SS_{AB}	$(a-1)(b-1)$	MS_{AB}	$F_{AB} = MS_{AB}/MS_{Error}$
Error	SS_{Error}	$ab(n-1)$	MS_{Error}	
Total	$SSTO$	$abn-1$		

Table 5.3: Symbolic Two-Factor ANOVA Table. Taken from Hogg/Ledolter, *Applied Statistics for Engineers and Physical Sciences*

"dot" values and two "asterisks" — with five being the most numbers removed from any one cell. Appendix E indicates exactly which values were removed. The general criterion used to define an outlier was that the measurement be over two inter-quartile ranges from the ends of the whiskers.

Figure 5-5 shows the new box plots. With the numbers removed, "new" outside values appear in some of the cells as a result. Since the aim was only to remove values far from the majority of the cell, we did not repeat the process until outside values no longer appeared in the plot. Hence, we made only one pass for removing outliers and trusted that the remaining outside values, as well as these "new" outside values, are valid data points.

After removal of outliers, there are 236 measurements left. For reasons described in the next section, another twenty were removed so that the analysis was conducted on a final data set of 216 measurements.

5.5 Analysis of Variance (ANOVA)

5.5.1 The ANOVA Table

Analysis of variance, or ANOVA, is a statistical technique for determining which factors have a significant effect on some dependent variable. Details and relevant formulas are given in Appendix D. Here we present a symbolic ANOVA table and explain some of the elements and how they can be interpreted. Since most of the ANOVAs in this thesis deal with only two factors at a time, we specifically present a two-factor ANOVA (Table 5.3).

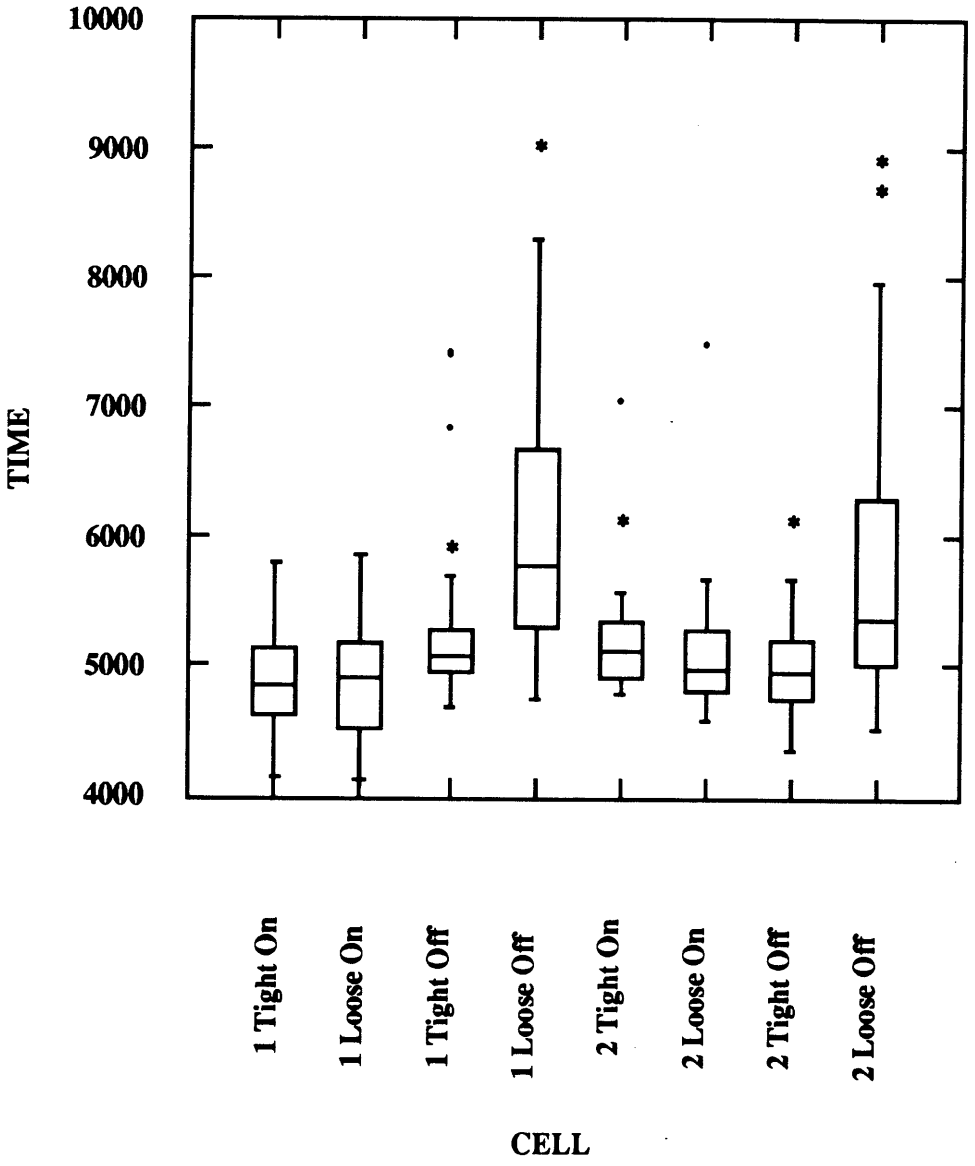

Figure 5-5: **Cell Box Plots with Outliers Removed** *Note that new outside values appear as a result of removing outliers. See text.*

Each row of the ANOVA table represents a different source of variation. They are the two factors (A and B), their interaction (AB), and the random errors. In our case, for example, factor A might be "test subject" and factor B might be "station keeping". When we subtract the grand mean (the average of every measurement in every cell) from each measurement, square each resulting difference, and add them all together, we obtain the total sum of squares or $SSTO$ as shown in the table. Symbolically,

$$SSTO = \sum_{i=1}^{N} (Y_i - \bar{Y})^2$$

where $\bar{Y}$ is the grand mean and N is the total number of data values, which in our case will turn out to be 108 (4 cells x 32 per cell). $SSTO$ is a measure of the aggregate variance of all the data from the grand mean, and can be decomposed into the SS terms above it:

$$SSTO = SS_A + SS_B + SS_{AB} + SS_{Error}$$

These SS terms are explicit functions of the measurements Y_i and the relevant equations are given in Appendix D.

The third column gives the "degrees of freedom" (df) for each source of variation, where a is the number of levels for factor A, b is the number of levels for factor B, and n is the number of replications within each cell. In our case, a and b are both equal to 2, since all three of our factors have 2 settings. Also, n is equal to 27 since there ended up being 27 measurements in each cell, as explained later. Mean sum of squares (MS) is equal to SS divided by df, and the associated F-ratios are given by MS/MS_{Error} as shown in the table.

If the F-ratio for a given factor is greater than what is called the critical F-ratio, then there is a statistically significant difference in the response variable between the different settings of that factor. The critical F-ratio for a given factor depends on that factor's df, the Error df and the particular confidence level desired, usually chosen to be 95% or 99%. Confidence levels can also be expressed in terms of the parameter α, where

$$(Confidence\ Level)\% = 100(1 - \alpha)$$

so that 95% confidence corresponds to an α of 0.05.

Three values are needed in order to find a critical F-ratio: α and two parameters called r_1 and r_2:

$$F[\alpha; r_1, r_2]$$

where r_1 is the factor df and r_2 the Error df. The critical value for factor A is

$$F_A(crit) = F[0.01; a-1, ab(n-1)] \tag{5.2}$$

Similarly,

$$F_B(crit) = F[0.01; b-1, ab(n-1)] \tag{5.3}$$

and for the AB interaction,

$$F_{AB}(crit) = F[0.01; (a-1)(b-1), ab(n-1)] \tag{5.4}$$

Knowing α and the two parameters, one can look up the corresponding $F(crit)$ in the appropriate table provided in statistics textbooks.

The statistics software used for the analysis in this thesis also calculates the p-value, which is the probability that a given F is greater than F_{crit}:

$$p\ value = P[F \geq F_{crit}] \tag{5.5}$$

In other words, a factor is significant at the $100(1-p)\%$ level, where p is the p-value for that factor. For example, factor A having a p-value of 0.05 is equivalent to saying that factor A is significant at the 95% level. Since the p-value test is more precise than merely comparing F-ratios, it is the significance test used in this thesis. Particularly because there were only two test subjects, a high confidence level is desired. Therefore, 99% confidence was chosen as a baseline requirement for significance, corresponding to a p-value of 0.01 or less.

5.5.2 Cell Equalization

The significance tests described above are only valid if there are the same number of measurements in every cell. If the cell sizes are not equal, the three sums of squares, SS_A, SS_B, and SS_{AB}, are not orthogonal, meaning we cannot test the significance of a factor independently of the others . For example, a B main effect might influence the significance test for the A main effect, and vice-versa. If that were the case, the ANOVA would be much less amenable to interpretation because it is very difficult, if not impossible, to ascertain the extent of that influence [4].

Unfortunately, the removal of outliers resulted in this very situation, with cells containing from 27 to 32 measurements. It was decided to remove the appropriate number of measurements from the end of each cell so that each contained 27 measurements. Hence, the final data set contained 8×27 or 216 numbers as mentioned earlier. As indicated on the data table in Appendix E, the measurements removed were merely the last (chronologically) few taken. Although it is generally unadvisable to discard data, equalizing the cell sizes was deemed an appropriate trade-off for two main reasons:

- Given that the original cell size was 32, the reduced cell size of 27 is not significantly smaller. Indeed, 27 is still a comfortably large sample size.
- The ANOVA is considerably more interpretable. The significance tests for ANOVAs with unequal cell sizes are vague at best, and in many instances are impossible to interpret [4].

5.5.3 Residual Diagnostics

Before accepting the results of an ANOVA, the residuals from the analysis must be checked. A residual is simply the difference between a data measurement and its cell average, hence there is a residual value for every data measurement (see Equation D.10).

ANOVA assumes that there is a certain amount of random noise in the measurements, i.e., the dependent variable data. Since the set of residuals is an estimate of

the noise, it should not have any discernible patterns. If it does, the simple additive model used by ANOVA is not valid and any conclusions cannot be accepted.

There are two residual checks used in our analysis. The first is the dot plot, which is a histogram with dots instead of bars. If the dot plot of residuals for each cell looks approximately normal (i.e., bell-shaped), then the residuals are probably random noise as desired. If the plot is not approximately normal, then there is probably something in the data not modelled by the ANOVA and therefore the analysis is invalid.

The second residual check is a plot of the residuals versus the cell averages. We do not want any patterns in this plot either, as a pattern would suggest that the supposedly random noise depends on the cell average. Again, this would indicate a violation of the ANOVA model.

5.6 Analysis Overview

The first question to ask is: Without arm motion compensation, does station keeping make a difference in teleoperation performance? If not, there would be little reason to even consider arm motion compensation. As described in Section 5.7, station keeping did indeed make a significant difference. The second question is then: With arm motion compensation, is there still a significant difference in performance? In Section 5.8, the analysis suggests that arm motion compensation eliminated the performance difference, allowing subjects to perform essentially as well under either station keeping setting. With this result in mind, we ask: Can arm motion compensation, in tandem with loose station keeping, provide the same performance benefits as tight station keeping alone? This question is the central focus of this thesis, and the analysis in Section 5.9 suggests that the answer is yes.

5.7 End Effector Disturbance Compensation Off

We first establish that in the absence of arm motion compensation, station keeping setting does make a difference. This involves the four cells, C_{Off}, where factor A is "test subject" and factor B is "station keeping". The reader is reminded of the obvious time dependence in the $C_{1-Loose-Off}$ measurements.

5.7.1 Data Transformation

The ANOVA assumes that variance in all cells is roughly equal. Looking at the four box plots for C_{Off} (extracted from Figure 5-4 and shown in Figure 5-6), this is clearly not the case. Transformations of the dependent variable can sometimes help make the spreads more equal [5]. Here we used:

$$Y^* = \frac{10000}{Y - 4100} \tag{5.6}$$

where Y is the original period measurement and Y^* is the transformed measurement. Figure 5-6 also shows box plots for the transformed data, whose inter-quartile ranges are slightly more comparable. The ANOVA in this section was done on this transformed data set, which satisfies the equal-variance assumption a bit better than the original data.

5.7.2 Analysis and Diagnostics

Table 5.4 shows the ANOVA results. The residuals from the ANOVA were plotted as described above and the results are in Figure 5-7. On the left is are dot plots which show that the residuals do form bell-shaped curves, suggesting a normal distribution as desired. The right graph displays a possible increase in residual spread as the cell average increases, but the trend is not obvious enough to say that a pattern exists. We can therefore accept the ANOVA results with some confidence.

Test subject is significant and somewhat expectedly so, since subjects are likely to differ in skill level. More importantly, station keeping is highly significant. Since,

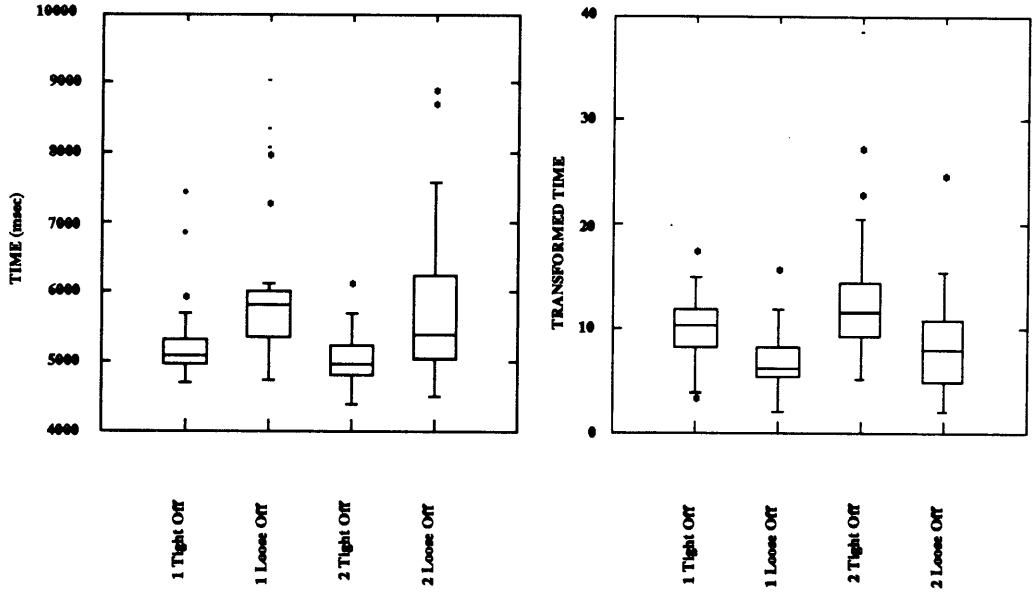

Figure 5-6: **Arm Motion Compensation Off: Box Plots Before And After Data Transformation**

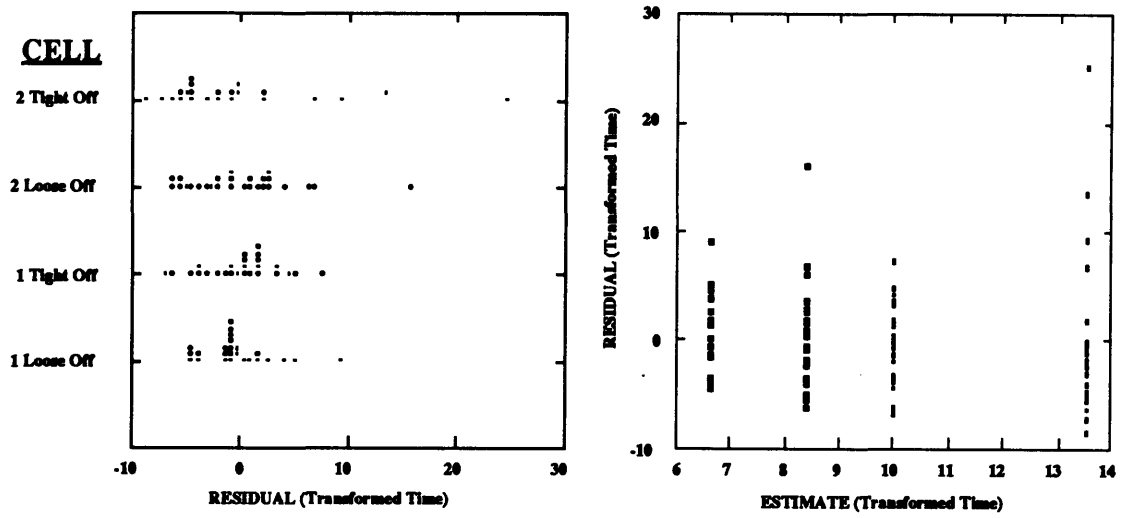

Figure 5-7: **Residual Diagnostic Plots: Arm Motion Compensation Off** *Left: Dot plot for each cell. Right: Residual vs Cell Average.*

as it turns out, the average of measurements in $C_{Tight-Off}$ is lower than the average of measurements in $C_{Loose-Off}$, the ANOVA results suggest that test subjects did indeed perform better with tight station keeping. Having established this, we next analyze the data where arm motion compensation was turned on to see if subjects still performed significantly better under tight station keeping.

Source	SS	df	MS	F	p-value
(A) Subject	191.1243	1	191.1243	7.5125	0.0072
(B) Station-Keeping	482.9784	1	482.9784	**18.9843**	**0.00003**
AB	21.1520	1	21.1520	0.8314	0.3640
Error	2645.8630	104	25.4410		
Total	3341.1177	107			

Table 5.4: ANOVA with Arm Motion Compensation Off: As the p-values show, both main effects are highly significant, particularly station keeping. Subject was either "1" or "2", and station keeping was "tight" or "loose". Note that the ANOVA was done on transformed data.

Source	SS	df	MS	F	p-value
(A) Subject	1.426×10^6	1	1.426×10^6	6.7241	0.0109
(B) Station Keeping	7.008×10^3	1	7.008×10^3	**0.0330**	**0.8561**
AB	1.968×10^5	1	1.968×10^5	0.9729	0.3376
Error	2.206×10^7	104	212072.2222		
Total	3.652×10^7	107			

Table 5.5: ANOVA with Arm Motion Compensation On: As the p-values show, station keeping is highly insignificant. Subject was either "1" or "2", and station keeping was "tight" or "loose".

5.8 End Effector Disturbance Compensation On

To investigate the effect of arm motion compensation, we want to perform the same analysis on the other four cells, C_{On}. Unlike the data for C_{Off}, there is no transformation required — the spreads within each cell are roughly equal as shown in Figure 5-4. We hope to find that station keeping is no longer significant, meaning that arm motion compensation has allowed subjects to perform as well under loose station keeping as they did under tight station keeping. As shown in Table 5.5, this is indeed the case. The p-value for station keeping is now 0.8561 so that it is now highly insignificant. Again, the reader is reminded of a possible time dependence in $C_{1-Tight-On}$.

Before accepting this result, we again do residual diagnostics. The two plots are shown in Figure 5-8. The dot plots are roughly normal and there are no patterns in the plot of residuals versus cell average. Arm motion compensation seems to eliminate the performance degradation caused by loose station keeping. Now we would like to find out if arm motion compensation yields the same performance benefits of tight station keeping.

5.9 Cross Comparison

In the two previous analyses, we performed an ANOVA where either all four cells were C_{Off}, or all four were C_{On}. Now we would like to compare the two cells, $C_{Tight-Off}$, with the two cells, $C_{Loose-On}$. Factor A is still test subject, but now factor B is

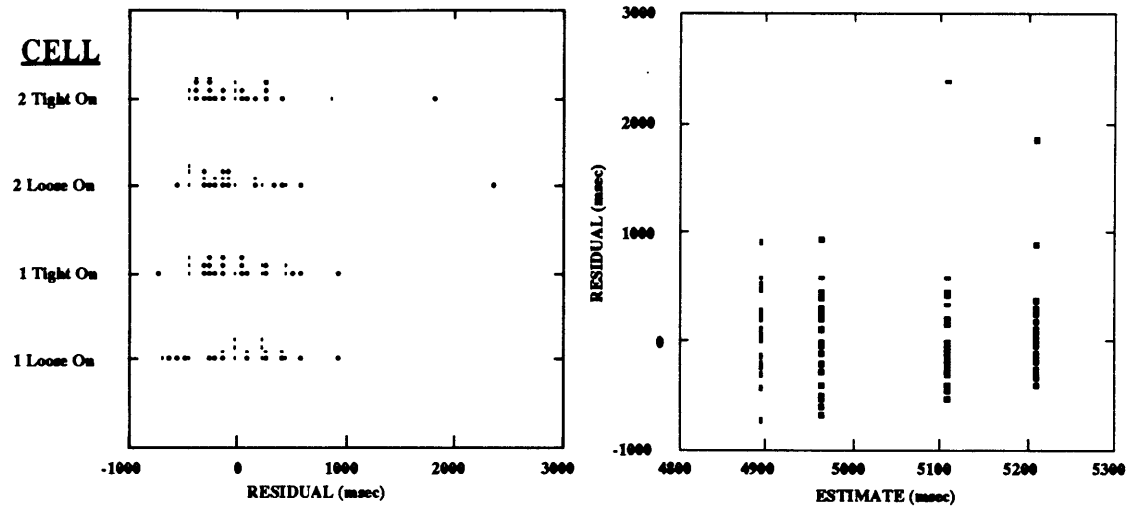

Figure 5-8: **Residual Diagnostic Plots: Arm Motion Compensation On** *Left: Dot plot for each cell. Right: Residual vs Cell Average.*

what we will call "configuration", which has two settings: Tight-Off and Loose-On. The main focus of this thesis is to see if there is a significant performance difference between these two configurations. As described in the next section, the two-factor ANOVA was somewhat inconclusive, so we did a separate one-factor ANOVA for each subject with configuration as the single factor.

5.9.1 Two-factor ANOVA

The two-factor ANOVA results are in Table 5.6 and the residual plots are shown in Figure 5-9. The interaction effect (AB) is not significant at the 99% level, but it is significant at the 95% level. A significant interaction would preclude the possibility of testing for main effects as we have done previously. Figure 5-10 is a plot of cell average versus configuration, for both test subjects, which illustrates the interaction. Because one graph slopes upward while the other slopes downward, the interaction effect in the ANOVA was significantly high and we cannot draw any conclusions from the two-factor ANOVA.

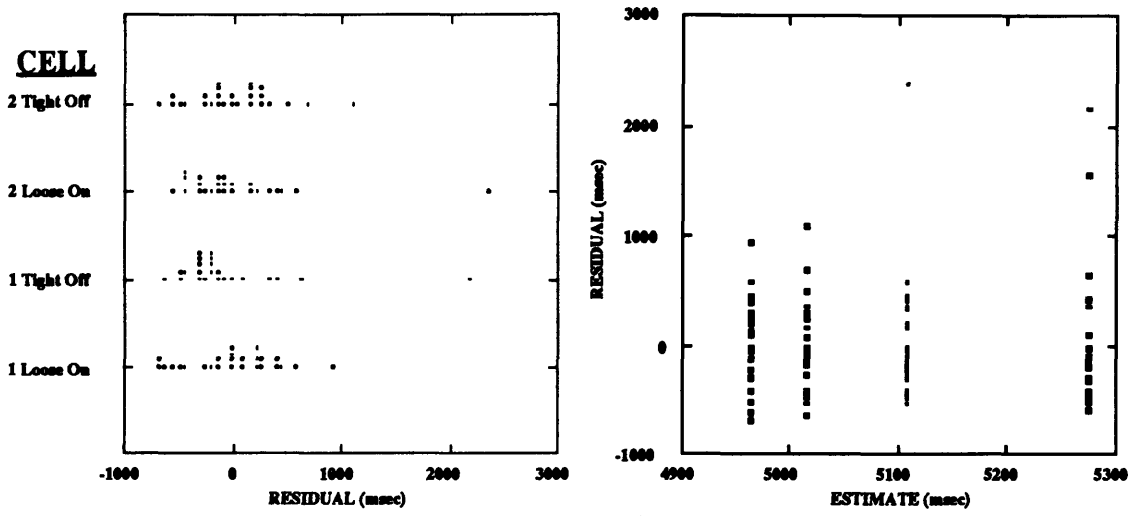

Figure 5-9: **Residual Diagnostic Plots: Cross Comparison**
Left: Dot plot for each cell. Right: Residual vs Cell Average.

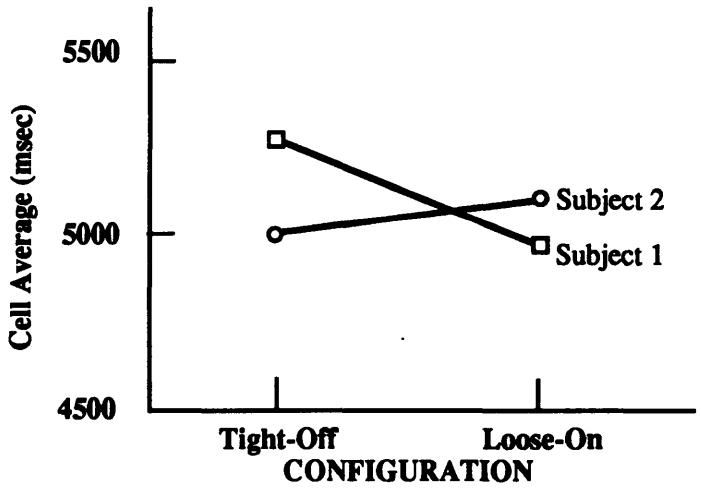

Figure 5-10: **Cell Average vs. Configuration: Cross Comparison**
Illustrates the interaction between the two factors "configuration" and "test subject"

Source	SS	df	MS	F	p-value
(A) Subject	89556.4815	1	89556.4815	0.3544	0.5529
(B) Configuration	314712.0370	1	314712.0370	1.2455	0.2670
AB	1.10211×10^6	1	1.10211×10^6	**4.3616**	**0.0392**
Error	2.6279×10^7	104	252682.7635		
Total	2.77853×10^7	107			

Table 5.6: Cross Comparison ANOVA: Factor B (configuration) has two settings: (1) Tight station keeping with no arm motion compensation and (2) Loose station keeping with arm motion compensation. Interaction effect (AB) is significant at the 95% level.

However, Figure 5-10 does suggest that subjects performed as well or better under the Loose-On configuration, which supports the hypothesis that arm motion compensation provides the performance benefits of tight station keeping. To test this, we can do a separate one-factor ANOVA for each subject where the one factor is "configuration".

5.9.2 One-factor ANOVAs

The one-factor ANOVA is very similar to the two-factor ANOVA. The critical F-ratio in this case is

$$F(crit) = F(\alpha; k-1, N-k) \tag{5.7}$$

where k is the number of levels for the factor (in our case, 2), and N is the total number of values in all cells. For both subjects, $N = 2 \times 27$ or 54. Again, we will use the p-value,

$$pvalue = P[F > F(\alpha; k-1, N-k)] \tag{5.8}$$

to test for significance.

Subject 1 seemingly performed much better with the Loose-On configuration. To test this, we do a one-factor ANOVA on the two cells, $\mathbf{C}_{1-Loose-On}$ and $\mathbf{C}_{1-Tight-Off}$. The result is shown in Table 5.7. The p-value of 0.0337 indicates significance at the 95% level. In conjunction with Figure 5-10, this is statistical evidence that subject 1 did indeed perform better under the Loose-On configuration.

For subject 2, we do a similar analysis on cells $\mathbf{C}_{2-Loose-On}$ and $\mathbf{C}_{2-Tight-Off}$,

Source	SS	df	MS	F	p-value
Configuration	1.29735×10^6	1	1.29735×10^6	**4.7609**	**0.0337**
Error	1.41701×10^7	52	272502.8940		
Total	1.546745×10^7	53			

Table 5.7: One-Factor ANOVA, Subject 1: Configuration has two settings: (1) Tight station keeping with no arm motion compensation and (2) Loose station keeping with arm motion compensation.

Source	SS	df	MS	F	p-value
Configuration	1.19474×10^5	1	1.19474×10^5	**0.5131**	**0.4770**
Error	1.21089×10^7	52	2.32862×10^5		
Total	1.222837×10^7	53			

Table 5.8: One-Factor ANOVA, Subject 2: Configuration has two settings: (1) Tight station keeping with no arm motion compensation and (2) Loose station keeping with arm motion compensation.

shown in Table 5.8. The residual dot plots for both one-factor ANOVAs are already shown in Figure 5-9; the two'residual versus cell-average" plots are given in Figure 5-11. The high p-value of 0.4770 is strong evidence that subject 2 performed equally well under either configuration.

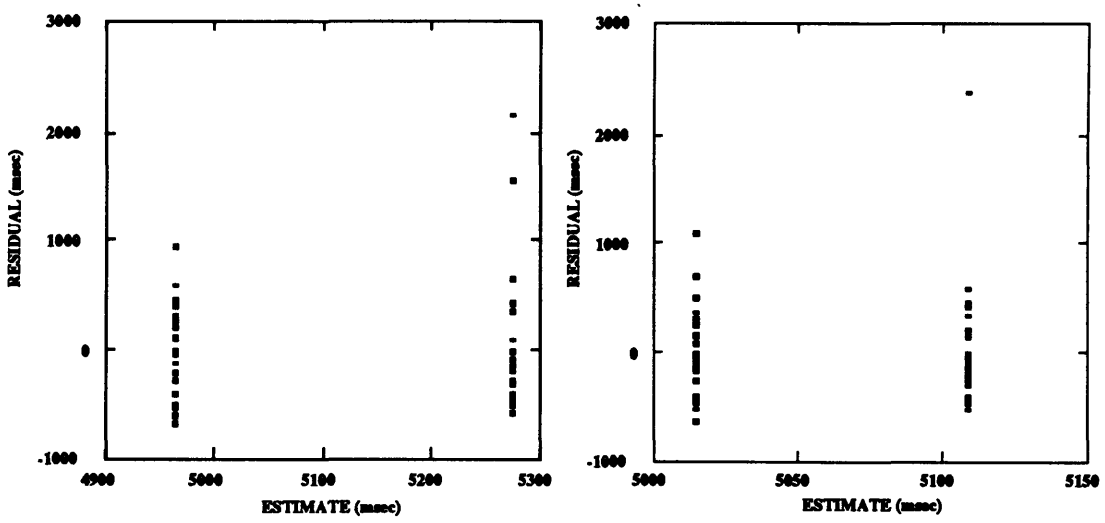

Figure 5-11: **One-Factor ANOVA Diagnostics: Residual vs. Cell Average** *Subject 1 on left, Subject 2 on right.*

5.10 Summary

In this chapter, the experimental design, procedure and data were presented. Next, a time series analysis and an outlier analysis was performed. Some background on analysis of variance (ANOVA) was then presented in anticipation of the analysis to follow.

It was first established that without arm motion compensation, subjects performed better with tight vehicle station keeping. Next, it was shown that arm motion compensation eliminated this performance difference. Finally, the two configurations Tight-Off and Loose-On were compared. The two-factor ANOVA was inconclusive so two separate one-factor ANOVAs were done. The results suggest that both subjects performed as well or better under the Loose-On configuration. While this agrees with the hypothesis, it is difficult to generalize regarding the benefits of arm motion compensation due to the small number of test subjects.

Chapter 6

Conclusion

6.1 Remarks

We conducted a factorial experiment with a neutral buoyancy robot to test whether vehicle station-keeping and end effector disturbance compensation significantly affect human teleoperation performance. The vehicle used for the experiment, called the Submersible for Telerobotic Astronautical Research, or STAR, is a free-flying underwater telerobot equipped with a three degree of freedom arm, a stereo pan/tilt camera platform, and a vision-based navigation system. Using visual feedback from a fixed onboard camera, test subjects performed a Fitts-type teleoperated tapping task with the robotic arm while the vision navigator station-kept the vehicle relative to a fixed visual reference target.

The data analysis was divided into three phases. First, it was established that there was a difference in tapping performance between tight and loose station keeping. Next, it was found that with end effector disturbance compensation activated, that performance difference was no longer significant. Finally, an analysis showed that arm motion compensation combined with loose station keeping allows equivalent teleoperation performance as tight station keeping alone. These findings suggest that we can get away with looser station keeping on a space-based teleoperator by intelligently controlling the arm to compensate for vehicle motion. The lower fuel consumption would substantially decrease the mass that must be launched into orbit

for fuel resupply, thus lowering the overall operating cost for free-flying telerobots.

Because of the large number of replications (27) within each cell, we can confidently accept the results of the analysis. However, with only two subjects, both right-handed males aged 18 to 25, it is difficult to make conclusions regarding human teleoperation performance in general. It is possible, for example, that both subjects are particularly well coordinated, or particularly uncoordinated, and hence very unrepresentative of the population. Even with only three more test subjects, as was originally intended, generalizations could be made much more confidently. If data from five test subjects yielded the same significance results as our data from only two test subjects, it could be more strongly concluded that arm motion compensation is indeed an effective alternative to tight station keeping.

6.2 Recommendations

Limitations of the arm precluded any type of gripping task for test subjects to complete, and yet such a task would certainly be more representive of actual space teleoperation. The small performance differences with our simple task imply that the differences could be quite substantial for a more difficult teleoperation task involving true manipulation. Similarly, a task requiring more forward-backward and up-down arm motion would also be more indicative.

With regard to loose station keeping, the difference we found was statistically significant, but it may not be practically significant. In other words, it may not matter if a task takes a bit longer. On the other hand, some test subjects commented that under loose station keeping, they could learn to use the vehicle's motion to their advantage. For example, the vehicle might already be moving in the desired direction, taking the arm along with it. In that case, the subject could simply hold the master arm still and let the target come to the robotic arm. This suggests that with practice, the operator could learn to perform just as well under loose station keeping, even without arm motion compensation engaged.

Since the camera is fixed to the vehicle, one issue that may be important is

the fact that arm motion compensation affects what the operator sees in the video monitor. For example, if the vehicle is moving left and the operator commands the arm left, the arm might move a bit less with arm motion compensation than without it. This in turn would induce a slightly different vehicle motion and hence affect the operator's view of the worksite. One possibility would be to move visual feedback to a pan/tilt camera that compensates for vehicle motion much like the arm does. An investigation of such a configuration might yield information about the effects of motion compensation in general.

Yet another factor that could be considered would be stereoscopic versus monoscopic visual feedback. In our experiments, the monitor gave very little depth information and that certainly limited the subjects' ability to do the task efficiently.

Another improvement to the procedure would have been to use pre-generated sequences of binary numbers to select which target indicator bulb to light, instead of using the computer's pseudo-random number generator as was done for this thesis. The data as collected gives the time at which a side target was tapped, but does not indicate whether it was a left target tap or a right target tap. Since the computer chose which side target was to be tapped, there is no record of specific tapping sequences, and therefore no distinction in the data between a right target tap and a left target tap.

The measure of performance was the time between taps of the center target. The reader will recall that in between each center tap was a tap of one of the side targets, but we do not know which one. It would be interesting to test for effects due to the side that the tap was on because possible biases may exist. For example, the right-handed subjects commented that a right tap was much more difficult because they had to move the master arm tip a large distance away from their body to reach that target. If a right tap is indeed fundamentally more difficult for a right-handed operator, then an explicit record of which side targets were tapped would allow the analysis to take this bias into account.

The requirement of having each subject encounter the four factor settings in a different order was imposed so that biases due to specific test sequences could be

minimized. With only two test subjects, there were only two different test sequences which were very similar in that both subjects performed the Tight-Off runs third and the Loose-On runs last. Since the Loose-On runs were of most interest in terms of performance evaluation, it was unfortunate that both subjects performed these runs last, raising the question of whether or not the extra practice was a significant factor. The obvious recommendation would be to use more test subjects, which would have resulted in more different test sequences. The test runs for a given factor setting, say Loose-On, would be done either first, second, third, or fourth relative to runs for the other three factor settings. With more test subjects, the Loose-On runs would have likely been distributed evenly among the four possible "slots", instead of always being done last, as was the case in our experiment.

Another benefit of having many test subjects is that one might include factors such as age-group or gender in the analysis. An experiment with a large and diverse group of test subjects, along with a large number of factors considered, might very well confirm our findings regarding the teleoperation performance benefits of end effector disturbance compensation.

References

[1] Eberly, Kurt, "An Underwater Neutral-Buoyancy Robot for Zero-Gravity Simulation with Attitude Control and Automatic Balancing," SM Thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, August 1991.

[2] Weigl, Harald, "Vision-Based Navigation and Control for Free-Flying Unmanned Remote Vehicles," SM Thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, August 1992.

[3] Fitts, P., "The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement," *Journal of Experimental Psychology*, Vol. 47, No. 6, June 1954.

[4] Lindman, H. R. *Analysis of Variance in Experimental Design.* Springer-Verlag, New York, 1992.

[5] Hogg, R. V. and Ledolter, J. *Applied Statistics for Engineers and Physical Scientists.* 2nd Edition, Macmillan Publishing, 1992.

Appendix A

Robotic Arm and Supporting Electronics

The motors used for the arm joints are Pittman GM14901C891 24V gearhead motors with a 19.7:1 gear ratio. They were ordered with a Hewlett-Packard 9100 Series encoder module and 500 CPR encoder wheel installed on the *motor* shaft (not the gearhead shaft).

Figure A-1 is a cutaway view of the cylindrical motor can described in Chapter 2. The motor is mated to the shaft by a coupling having some side-to-side play, which allows smooth turning even when the shaft becomes slightly misaligned due to lateral torques. An O-ring between the shaft and the endcap provides a water-tight seal while still allowing rotation. Between the endcap and the sidewall is an O-ring, which again provides a water-seal.

On the other endcap is a pipe-fitting through which the electrical and electronic signals are passed: motor + and −, +5V, GND, ENCA (Encoder A), and ENCB (Encoder B). For water-proofing, cable-splicing material was poured into the fitting and allowed to dry. Once closed and sealed, the can is literally a water-tight motor, with input wires, and a rotating shaft. The three motor can joints, one for each degree of freedom, are linked as shown in Figure 2-3. The yaw and pitch motor cans are linked side-by-side and perpendicular to each other. The pitch and elbow cans are linked via a combined yoke/bar assembly.

Figure A-2 shows the H-bridge circuit used to drive the arm motors. The PS2505 opto-isolator chip accepts FOR, REV, and MAG inputs from the controller board and drives the corresponding 2N2222A transistors. The 2N2222A in turn activates an MJ11033 (PNP) power transistor or an MJ11032 (NPN) power transistor. For a given direction input, a diagonal pair of power transistors is activated, driving current through whatever device happens to be placed between the MOT1 and MOT2 outputs. The encoder related signals (+5V, GND, ENC, and ENCB) are passed through the driver unit, and with the exception of +5V, are not part of the driver circuit. This arrangement is used so that all signals going to the motor are routed through the driver unit, eliminating the need to have two separate cables to the motor — one from the controller board, and one from the driver.

The driver circuit is built onto a small I-beam section and the electrical components are sealed with a silicon-rubber compound. The resulting driver unit is a 4″x4″x3″ water-tight 'brick' of aluminum and rubber. +24V power comes from the onboard lead-acid batteries.

Figure A-3 illustrates how the LM629 controller is interfaced to the driver. The chip outputs a SIGN and a MAGnitude signal, which are routed to a Darlington array (ULN2803A). The encoder signals coming from the motor can are buffered with an LS245 before being sent to the LM629. The pull-up resistors to +5V insure that the encoder signals are well behaved, i.e., pulled well high when they are not specifically driven low by the encoder. The actual controller board has three LM629s, one for each motor corresponding to each DOF.

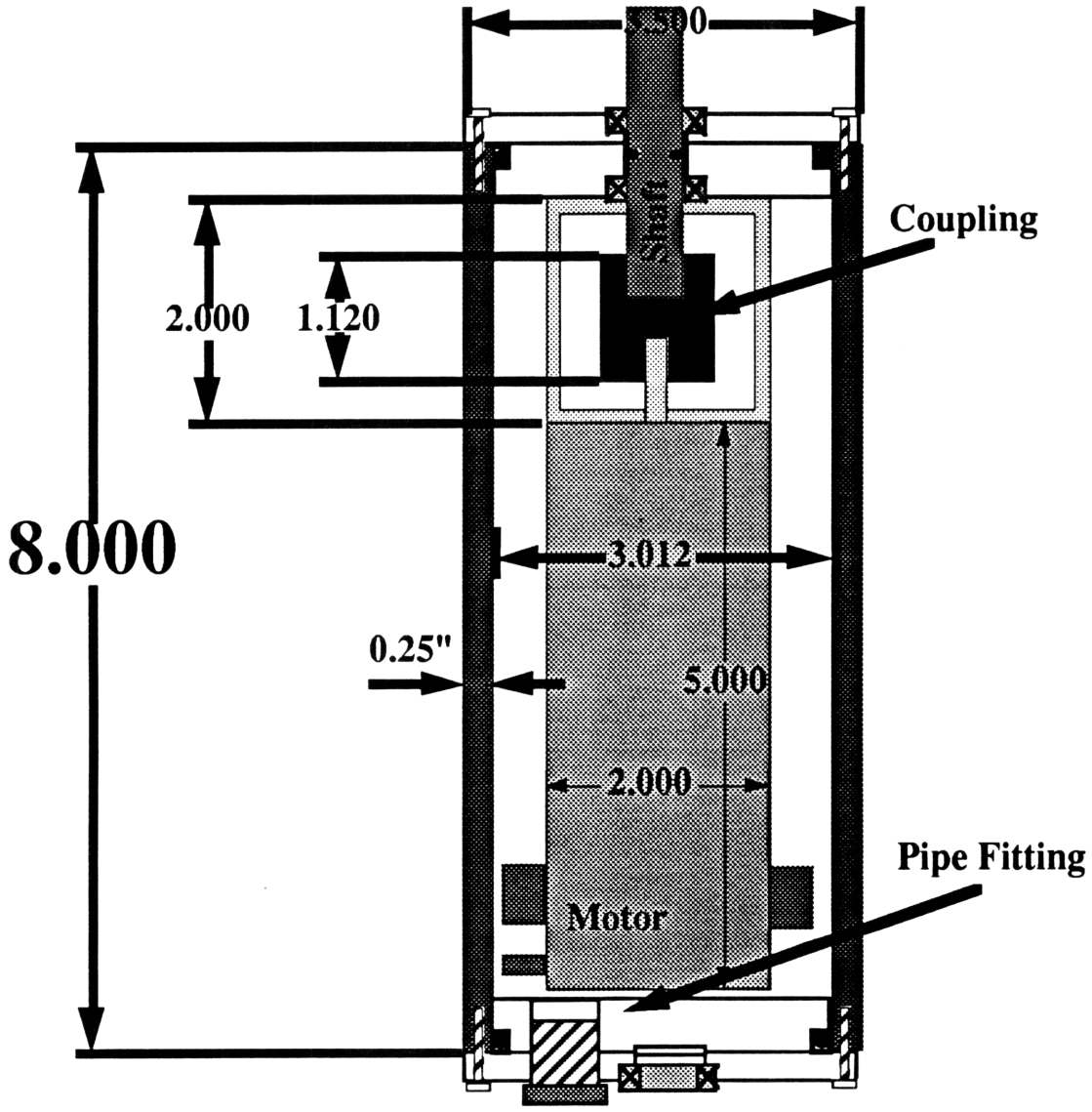

Figure A-1: **Motor Can Cut-away View** *Designed by Paul Stach and Dean Franck. Drawn by Paul Stach. Dimensions are in inches.*

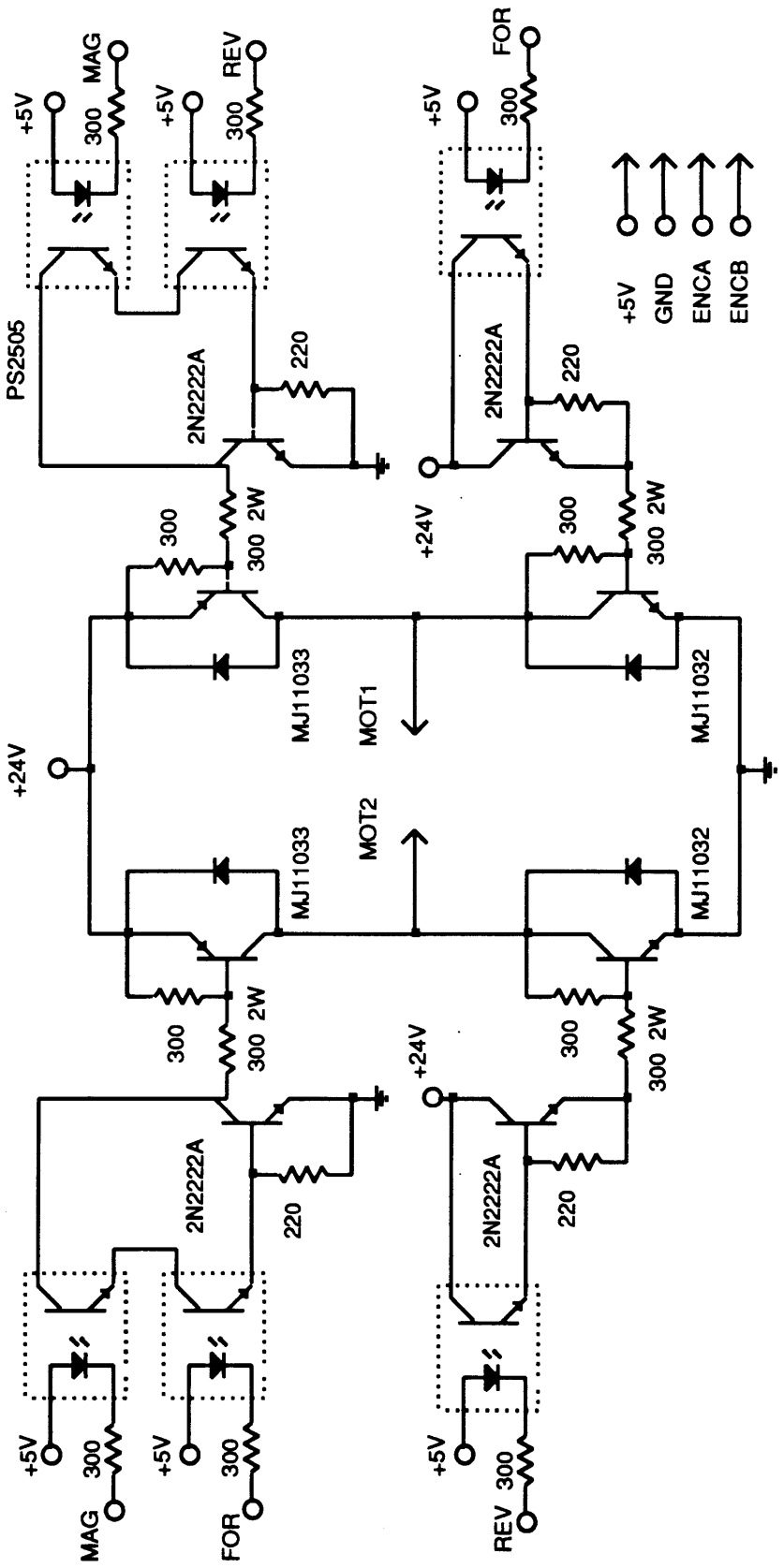

Figure A-2: **Motor Driver Circuit** *Circles denote signals in from the LM629 motor controller (MOCO) board. Arrows denote signals out to the motor can (motor and encoder).*

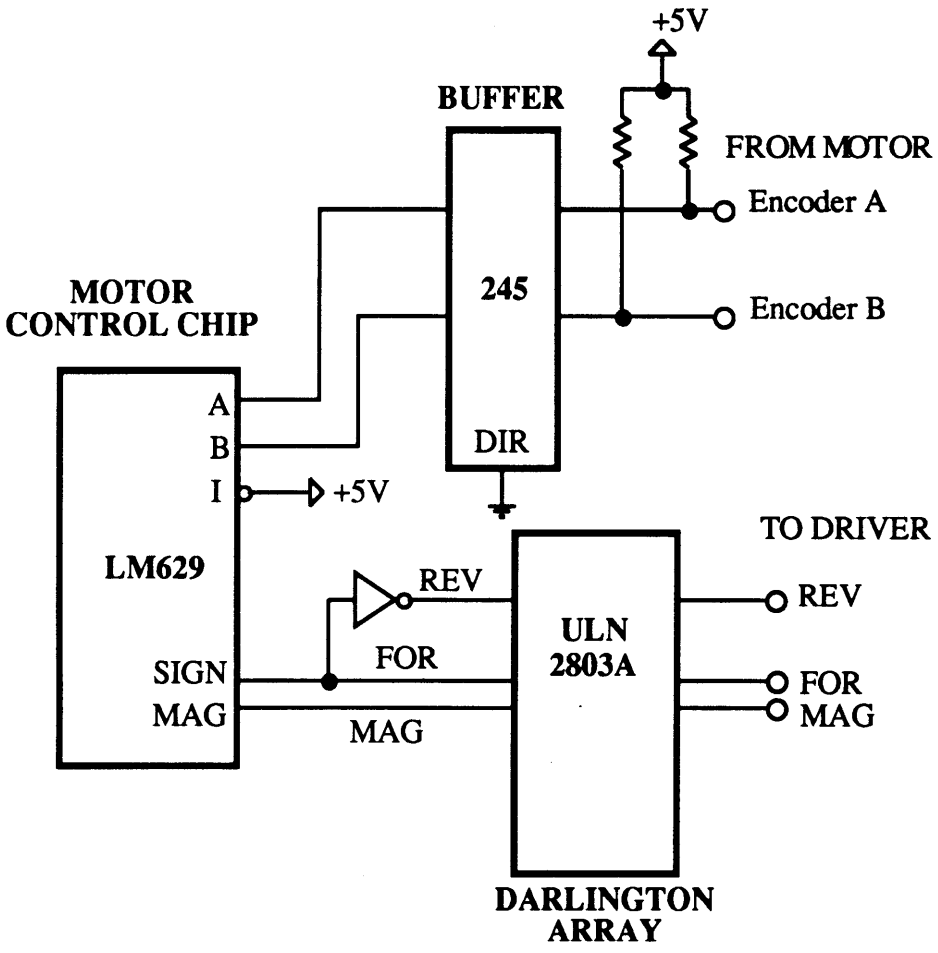

Figure A-3: **Interfacing the LM629** *Output of the LM629 is routed to a Darlington array (ULN2803A) whose outputs go directly to the motor driver.*

Appendix B

Jacobian Mathematics and Software Listing

This appendix section details the derivation of the Jacobian matrix for SAM, STAR's three DOF arm. Also included is a listing of the relevant 'C' functions within the control software — `initKin()`, `invKin()`, and `compensate()`.

B.1 Jacobian

Figure B-1 illustrates the robotic arm and its kinematic parameters:

- Link lengths L_1–L_4,
- Joint angular velocity vectors, $\vec{\omega}_{yaw}$, $\vec{\omega}_{pitch}$, $\vec{\omega}_{elbow}$, and $\vec{\omega}_{wrist}$,
- Joint angles θ_{yaw}, θ_{pitch}, θ_{elbow}, and θ_{wrist},
- The vectors $\vec{r}_1$, $\vec{r}_2$, and $\vec{r}_3$, which point to the arm tip from the yaw, pitch and elbow joints, respectively.

We can write

$$\begin{bmatrix} \dot{x}_{tip} \\ \dot{y}_{tip} \\ \dot{z}_{tip} \end{bmatrix} = \begin{bmatrix} \frac{\partial \dot{x}_{tip}}{\partial \dot{\theta}_{yaw}} & \frac{\partial \dot{x}_{tip}}{\partial \dot{\theta}_{pitch}} & \frac{\partial \dot{x}_{tip}}{\partial \dot{\theta}_{elbow}} \\ \frac{\partial \dot{y}_{tip}}{\partial \dot{\theta}_{yaw}} & \frac{\partial \dot{y}_{tip}}{\partial \dot{\theta}_{pitch}} & \frac{\partial \dot{y}_{tip}}{\partial \dot{\theta}_{elbow}} \\ \frac{\partial \dot{z}_{tip}}{\partial \dot{\theta}_{yaw}} & \frac{\partial \dot{z}_{tip}}{\partial \dot{\theta}_{pitch}} & \frac{\partial \dot{z}_{tip}}{\partial \dot{\theta}_{elbow}} \end{bmatrix} \begin{bmatrix} \dot{\theta}_{yaw} \\ \dot{\theta}_{pitch} \\ \dot{\theta}_{elbow} \end{bmatrix} \tag{B.1}$$

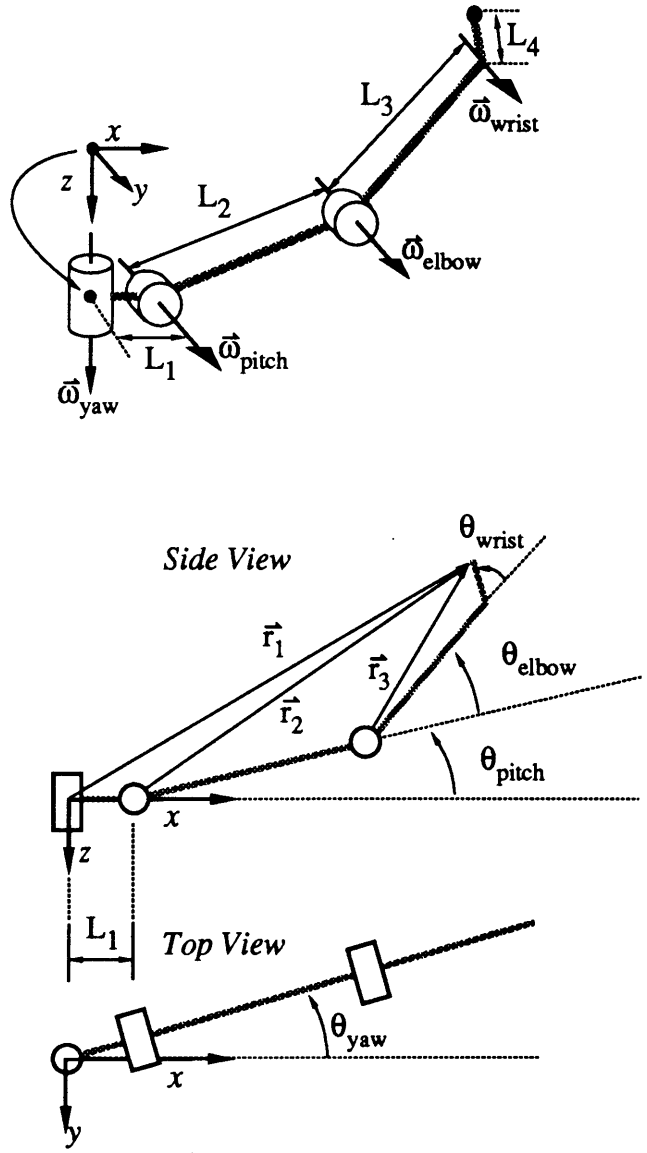

Figure B-1: **STAR Robotic Arm: Kinematic Parameters** *Top picture shows the coordinate system, the link lengths L_1–L_4, and the joint angular velocity vectors, $\vec{\omega}_{yaw}$, $\vec{\omega}_{pitch}$, $\vec{\omega}_{elbow}$, and $\vec{\omega}_{wrist}$. The other two diagrams show the arm joint angles θ_{yaw}, θ_{pitch}, θ_{elbow}, and θ_{wrist}, as well as the three vectors $\vec{r}_1$, $\vec{r}_2$, and $\vec{r}_3$.*

or

$$\vec{v}_{tip} = \mathbf{J}\ \dot{\vec{\Theta}}_{arm} \tag{B.2}$$

Since the wrist is at a fixed angle (there is no actuator at the wrist), $\dot{\theta}_{wrist}$ is zero and therefore not included in the above formulation . The wrist joint angle θ_{wrist} and link length L_4 are accounted for in the expression for $\vec{r}_1$, as described below.

To derive the Jacobian J, we need to express the three Cartesian arm tip velocity components in terms of the three joint angular velocities and the four link lengths. To do this, we use the vector relation:

$$\vec{v}_{tip} = \vec{\omega}_{yaw} \times \vec{r}_1 \quad + \quad \vec{\omega}_{pitch} \times \vec{r}_2 \quad + \quad \vec{\omega}_{elbow} \times \vec{r}_3 \tag{B.3}$$

Using geometry, we can express $\vec{r}_1$, $\vec{r}_2$, and $\vec{r}_3$ as follows:

$$\vec{r}_1 = \begin{bmatrix} L_1 \cos\theta_{yaw} + L_2 \cos\theta_{yaw} \cos\theta_{pitch} + L_3 \cos\theta_{yaw} \cos\theta_{PE} + L_4 \cos\theta_{yaw} \cos\theta_{PEW} \\ L_1 \sin\theta_{yaw} + L_2 \sin\theta_{yaw} \cos\theta_{pitch} + L_3 \sin\theta_{yaw} \cos\theta_{PE} + L_4 \sin\theta_{yaw} \cos\theta_{PEW} \\ -L_2 \sin\theta_{pitch} - L_3 \sin\theta_{PE} + -L_4 \sin\theta_{PEW} \end{bmatrix}$$

$$\vec{r}_2 = \vec{r}_1 + \begin{bmatrix} -L_1 \cos\theta_{yaw} \\ -L_1 \sin\theta_{yaw} \\ 0 \end{bmatrix}$$

$$\vec{r}_3 = \vec{r}_2 + \begin{bmatrix} -L_2 \cos\theta_{yaw} \cos\theta_{pitch} \\ -L_2 \sin\theta_{yaw} \cos\theta_{pitch} \\ L_2 \sin\theta_{pitch} \end{bmatrix} \tag{B.4}$$

where

$$\theta_{PE} = \theta_{pitch} + \theta_{elbow}$$

$$\theta_{PEW} = \theta_{pitch} + \theta_{elbow} + \theta_{wrist}$$

The angular velocity vectors are given by:

$$\vec{\omega}_{yaw} = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_{yaw} \end{bmatrix}$$

$$\vec{\omega}_{pitch} = \vec{\omega_{elbow}} = \begin{bmatrix} -\dot{\theta}_{pitch} \sin\theta_{yaw} \\ \dot{\theta}_{pitch} \cos\theta_{yaw} \\ 0 \end{bmatrix}$$

With these expressions, we can calculate the vector cross products in Equation B.3. To write out the J matrix, we arrange the resulting equations to match the form of Equation B.1. If r_{1x} refers to the x component of $\vec{r}_1$, and so on, we can express J as

$$\mathbf{J} = \begin{bmatrix} -r_{1y} & r_{2z}\cos\theta_{yaw} & r_{3z}\cos\theta_{yaw} \\ r_{1x} & r_{2z}\sin\theta_{yaw} & r_{3z}\sin\theta_{yaw} \\ 0 & -r_{2x}\cos\theta_{yaw} - r_{2y}\sin\theta_y & -r_{3x}\cos\theta_{yaw} - r_{3y}\sin\theta_y \end{bmatrix} \tag{B.5}$$

As stated in Section 3.2, the Jacobian should be updated for the current position of the arm, that is, for the current values of θ_{yaw}, θ_{pitch}, θ_{elbow}, and θ_{wrist} (fixed). Instead, the Jacobian was pre-calculated for the nominal position:

$$\theta_{yaw} = 0 \qquad \theta_{pitch} = 0$$

$$\theta_{elbow} = \frac{\pi}{2} rad \qquad \theta_{wrist} = -\frac{\pi}{4} rad$$

B.2 Code Listing

The following is a code listing for the three "C" functions within the control program which implement the arm motion compensation system.

- `initKin()` calculates the Jacobian matrix for the specific nominal position given above. This function is executed prior to the control loop.

- `invKin()` calculates the joint angle deviations necessary to produce the desired Cartesian motion of the arm tip. It is executed within `compensate()`.

- `compensate()` uses the vehicle state estimates from the vision navigator to calculate the necessary adjustments to the master arm joint angle commands. The extra components required to compensate for vehicle rotation are accounted for within this function, which is executed when arm motion compensation is engaged.

The code was written by Harald Weigl, a research assistant at the MIT Laboratory for Space Teleoperation and Robotics (LSTAR).

```
/* End effector disturbance compensation (EEDC) code listing */
/* initKin() is executed prior to the control loop           */
/* compensate(), and therefore invKin(), is executed         */
/* within the control loop when EEDC is engaged              */
/* Written by Harald Weigl, Fall 1992                    */

#define PI  3.1415926
/* conversion from angle (rad)  to encoder counts    */
#define angleToCounts ((19.6*4*500)/(2*PI))
#define INTOM .0254   /* inches to meters conversion */
#define L1  4.625*INTOM
#define L2  28.5*INTOM
#define L3  24.5*INTOM
#define L4  2.125*INTOM

/* calculate the Jacobian */
void initKin()
{
      float cY, cP, cPE, cPEW, sY, sP, sPE, sPEW;
      float r2x, r2y, r2z;
      float r3x, r3y, r3z;

      nomY = 0;
      nomP = 0;
      nomE = PI/2;
      nomW = -PI/4;

      cY = cos(nomY);
      cP = cos(nomP);
      cPE = cos(nomP + nomE);
      cPEW = cos(nomP + nomE + nomW);
      sY = sin(nomY);
      sP = sin(nomP);
      sPE = sin(nomP + nomE);
      sPEW = sin(nomP + nomE + nomW);

      r1x = L1*cY + L2*cY*cP + L3*cY*cPE + L4*cY*cPEW;
      r1y = L1*sY + L2*sY*cP + L3*sY*cPE + L4*sY*cPEW;
      r1z = -L2*sP - L3*sPE - L4*sPEW;

      r2x = r1x - L1*cY;
      r2y = r1y - L1*sY;
      r2z = r1z;

      r3x = r2x - L2*cY*cP;
```

```
        r3y = r2y - L2*sY*cP;
        r3z = r2z + L2*sP;

        J1 = -r1y;
        J2 = r1x;

        J3 = cY*r2z;
        J4 = sY*r2z;
        J5 = -sY*r2y - cY*r2x;

        J6 = cY*r3z;
        J7 = sY*r3z;
        J8 = -sY*r3y - cY*r3x;
}

/* Given Cartesian coordinates, find corresponding arm joint angles */
void invKin(float dx,float dy,float dz,
            float *dYaw,float *dPitch,float *dElbow)
{
        *dElbow = (J2*J5*dx - J1*J5*dy + J1*J4*dz - J2*J3*dz)/
                   (J1*J4*J8 - J1*J5*J7 - J2*J3*J8 + J2*J5*J6);
        *dPitch = (dz - J8**dElbow)/J5;
        *dYaw = dy/J2 - (J4*dz)/(J2*J5) +
                     (J4*J8**dElbow)/(J2*J5) - (J7**dElbow)/J2;
}

/* Use vehicle state estimates from vision navigator to adjust the   */
/* commands to the arm.                                              */
/* ayaw, apitch and aelbow are the commands read from the master arm */
void compensate(float vx, float vy, float vz,
                float vRoll, float vPitch, float vYaw,
                float ayaw, float apitch, float aelbow)
{
        float deltaX, deltaY, deltaZ;
        float dYaw, dPitch, dElbow;

        deltaX = -vx - r1z*vPitch + r1y*vYaw;
        deltaY = -vy + r1z*vRoll - r1x*vYaw;
        deltaZ = -vz - r1y*vRoll + r1x*vPitch;

        invKin(deltaX, deltaY, deltaZ, &dYaw, &dPitch, &dElbow);

        moveArmFast((ayaw+dYaw),(apitch+dPitch),(aelbow+dElbow));
}
```

Appendix C

Tapping Target Electronics and Software

A schematic of the indicator light circuit is shown in Figure C-1. A 12-volt lead-acid battery supplies power, and the two input signals come from the control station computer (a 80386 PC, see [2]) via the PC74 Analog-Digital I/O board from Industrial Computer Source.

The control loop, a task on the control station computer, is responsible for the following:

- Reading joysticks, master arm, and other controllers.
- Sending these inputs to the onboard computer, which commands the arm and the motors.
- Receiving sensor readings and other data from the onboard computer.
- Vision processing, including frame grabbing and vision calculations.
- Detecting target taps, recording times, and lighting the appropriate indicator bulbs.

The last item is accomplished by the function `rdTap()` that is executed once every control loop. A listing of the code for `rdTap()` follows.

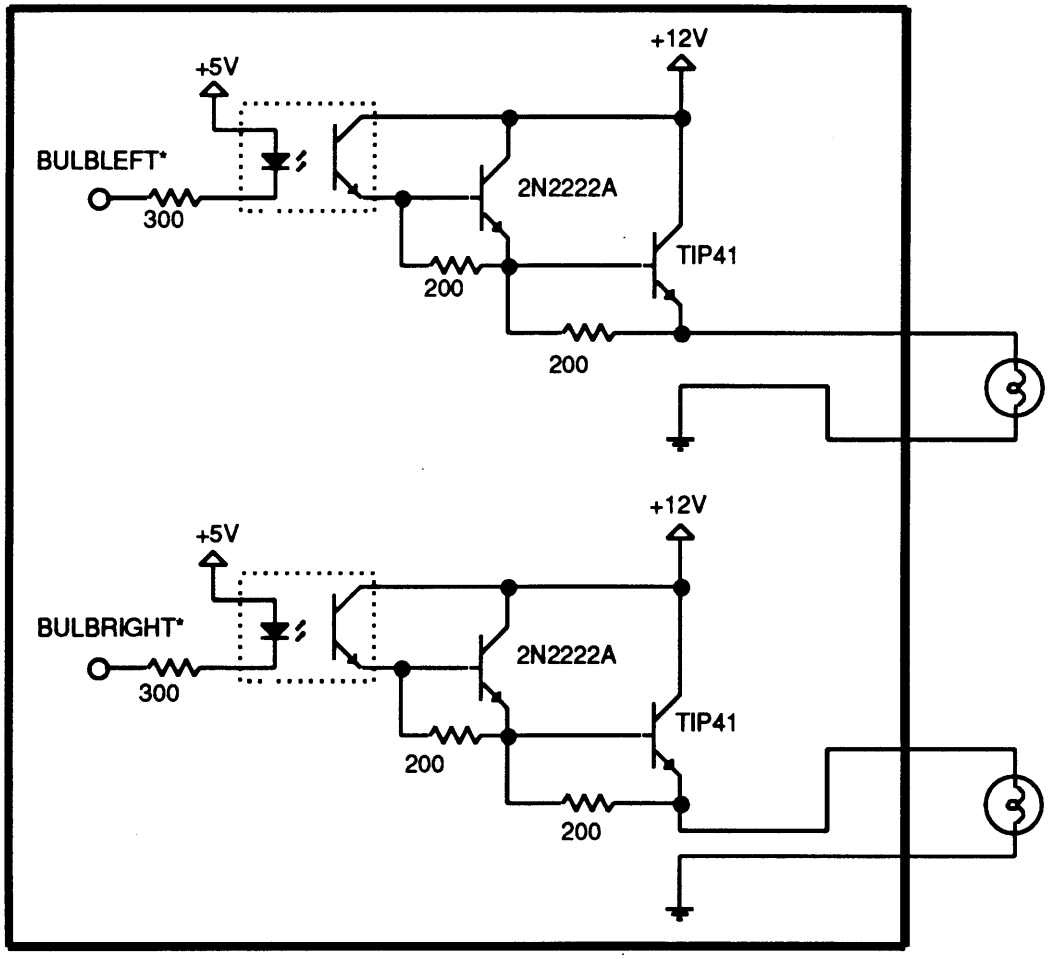

Figure C-1: **Amplifier Circuit for Target Indicator Lights**

```
#define START   1            /* modes in which ... */
#define REGULAR 2            /* ... rdTap() is invoked */
#define LEFT    0x20         /* mask for left tapping target */
#define RIGHT   0x10         /* mask for right tapping target */
#define CENTER  0x40         /* mask for center tapping target */
#define ONL     0x20         /* left indicator bulb */
#define ONR     0x08         /* right indicator bulb */
#define OFF     0x28         /* both indicator bulbs */
#define TH      (32767./2.)  /* threshold for deciding which bulb to light */
#define NTAPS   10           /* number of taps in a run */
#define BEEP    putch(7)     /* sound the PC's bell */


/* This function is executed once every control loop */
/* dOut() and dIn() are functions to read from and write to the PC74 */
/* digital I/O Board from Industrial Computer Source */
/* Code written by Michael Valdez, Fall 1992 */

int rdTap(int status)
{
   int              i;
   static int       expect, n;
   static long int  dt[NTAPS];
   FILE             *fp;

   switch(status)
   {
      case START:        /* beginning of a task run */
         dOut(OFF);        /* both bulbs off */
         n=0;              /* start a new data run */
         expect = CENTER;  /* start looking for a tap of the center target */
         break;
      case REGULAR:      /* not the beginning of a task run */
         switch(expect)
         {
            case CENTER:
               dOut(OFF);
               if(dIn()&CENTER)==CENTER) {
                  BEEP;
                  dt[n++] = clock();
                  if (rand()>=TH)   expect = LEFT;
                  else              expect = RIGHT;
                  }
               break;
            case LEFT:
```

```
            dOut(ONL); /* turn on left light */
            if(dIn()&LEFT)==LEFT {
               dt[n] = clock();
               BEEP;
               dOut(OFF);        /* bulb off */
               expect = CENTER; /* start looking for center tap again */
               n++;
               }
            break;
         case RIGHT:
            dOut(ONL); /* turn on right light */
            if(dIn()&RIGHT)==RIGHT {
               dt[n] = clock();
               BEEP;
               dOut(OFF);        /* bulb off */
               expect = CENTER; /* start looking for center tap again */
               n++;
               }
            break;
      }

      if(n==NTAPS) {    /* run is over */
         fp = fopen("temp.dat", "a");
         for(i=0;i<NTAPS;i++)
            fprintf(fp, "%ld\t", dt[i]-dt[0]);
         fprintf(fp, "\n");
         fclose(fp);
         return(1)
         }
      else
         return(0);
      break;
   }

}
```

Appendix D

Two-Factor Analysis of Variance (ANOVA)

This discussion of two-factor ANOVA is based on material found in [5] and readers are referred to that book for more details. For the three two-factor ANOVAs presented in Chapter 5, there were 2 factors each with a=b=2 levels resulting in 2^2 combinations, or cells. Within each cell, there were n=27 observations. Table D.1 illustrates the layout, where the Y's are the observations within each cell. Therefore, for a given ANOVA there were a total of abn=108 observations, generated by the assumed model:

$$Y_{ijk} = \mu_{ij} + \varepsilon_{ijk} \tag{D.1}$$

where $i = 1,\ldots,a$ (factor A), $j = 1,\ldots,b$ (factor B), $k = 1,\ldots,n$ (replications), and ε_{ijk} is $N(0,\sigma^2)$ noise. Each cell has its own mean, μ_{ij}. Define the overall mean

	Factor B			
Factor A	1	2	$\cdots$	b
1	$Y_{111}, Y_{112},\ldots, Y_{11n}$	$Y_{121}, Y_{122},\ldots, Y_{12n}$	$\cdots$	$Y_{1b1}, Y_{1b2},\ldots, Y_{1bn}$
2	$Y_{211}, Y_{212},\ldots, Y_{21n}$	$Y_{221}, Y_{222},\ldots, Y_{22n}$	$\cdots$	$Y_{2b1}, Y_{2b2},\ldots, Y_{2bn}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
a	$Y_{a11}, Y_{a12},\ldots, Y_{a1n}$	$Y_{a21}, Y_{a22},\ldots, Y_{a2n}$	$\cdots$	$Y_{ab1}, Y_{ab2},\ldots, Y_{abn}$

Table D.1: Two-Factor Factorial Design. Based on a similar table in Hogg/Ledolter, *Applied Statistics for Engineers and Physical Scientists*

as

$$\bar{\mu}_{..} = \frac{1}{ab}\sum_{i=1}^{a}\sum_{j=1}^{b}\mu_{ij} \tag{D.2}$$

the row (factor A) means as

$$\bar{\mu}_{i.} = \frac{1}{b}\sum_{j=1}^{b}\mu_{ij}, \quad i = 1, \ldots, a \tag{D.3}$$

and the column (factor B) means as

$$\bar{\mu}_{.j} = \frac{1}{a}\sum_{i=1}^{a}\mu_{ij}, \quad j = 1, \ldots, b. \tag{D.4}$$

As mentioned in Chapter 5, an additive model is used:

$$\mu_{ij} = \bar{\mu}_{..} + \alpha_i + \beta_j + (\alpha\beta)_{ij} \tag{D.5}$$

where the effects α_i (factor A), β_j (factor B), and $(\alpha\beta)_{ij}$ (AB interaction) are given by

$$\begin{aligned} \alpha_i &= \bar{\mu}_{i.} - \bar{\mu}_{..} \\ \beta_j &= \bar{\mu}_{.j} - \bar{\mu}_{..} \\ (\alpha\beta)_{ij} &= \mu_{ij} - \bar{\mu}_{i.} - \bar{\mu}_{.j} + \bar{\mu}_{..} \end{aligned} \tag{D.6}$$

Combining this with the observation model (Equation D.1) yields the following expression for Y_{ijk}

$$Y_{ijk} = \bar{\mu}_{..} + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \varepsilon_{ijk} \tag{D.7}$$

and estimates for μ_{ij}, $\bar{\mu}_{i.}$, $\bar{\mu}_{.j}$, and $\bar{\mu}_{..}$, are respectively given by:

$$\begin{aligned} \bar{Y}_{ij.} &= \frac{1}{n}Y_{ij.} = \frac{1}{n}\sum_{k=1}^{n}Y_{ijk} \\ \bar{Y}_{i..} &= \frac{1}{bn}Y_{i..} = \frac{1}{bn}\sum_{j=1}^{b}\sum_{k=1}^{n}Y_{ijk} \end{aligned}$$

$$\begin{aligned}
\bar{Y}_{.j.} &= \frac{1}{an} Y_{.j.} = \frac{1}{an} \sum_{i=1}^{a} \sum_{k=1}^{n} Y_{ijk} \\
\bar{Y}_{...} &= \frac{1}{abn} Y_{...} = \frac{1}{abn} \sum_{i=1}^{a} \sum_{j=1}^{b} \sum_{k=1}^{n} Y_{ijk}
\end{aligned} \tag{D.8}$$

and the effects α_i (factor A), β_j (factor B), and $(\alpha\beta)_{ij}$ (AB interaction), are estimated by:

$$\begin{aligned}
\alpha_i(est) &= \bar{Y}_{i..} - \bar{Y}_{...} \\
\beta_j(est) &= \bar{Y}_{.j.} - \bar{Y}_{...} \\
(\alpha\beta)_{ij}(est) &= \bar{Y}_{ij.} - \bar{Y}_{i..} - \bar{Y}_{.j.} + \bar{Y}_{...}
\end{aligned} \tag{D.9}$$

and the errors ε_{ijk} are approximated by the residuals e_{ijk}:

$$e_{ijk} = Y_{ijk} - \bar{Y}_{ij.} \tag{D.10}$$

From Equation D.7, the following data composition can be written:

$$Y_{ijk} = \bar{Y}_{...} + (\bar{Y}_{i..} - \bar{Y}_{...}) + (\bar{Y}_{.j.} - \bar{Y}_{...}) + (\bar{Y}_{ij.} - \bar{Y}_{i..} - \bar{Y}_{.j.} + \bar{Y}_{...}) + (Y_{ijk} - \bar{Y}_{ij.}) \tag{D.11}$$

and the corresponding the sum-of-squares decomposition is

$$SSTO = SS_A + SS_B + SS_{AB} + SS_{Error}, \tag{D.12}$$

as given in Chapter 5, where

$$\begin{aligned}
SSTO &= \sum_{i=1}^{a} \sum_{j=1}^{b} \sum_{k=1}^{n} (Y_{ijk} - \bar{Y}_{...})^2 \\
SS_A &= bn \sum_{i=1}^{a} (\bar{Y}_{i..} - \bar{Y}_{...})^2 \\
SS_B &= an \sum_{j=1}^{b} (\bar{Y}_{.j.} - \bar{Y}_{...})^2 \\
SS_{AB} &= n \sum_{i=1}^{a} \sum_{j=1}^{b} (\bar{Y}_{ij.} - \bar{Y}_{i..} - \bar{Y}_{.j.} + \bar{Y}_{...})^2
\end{aligned}$$

$$SS_{Error} = \sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n}(Y_{ijk} - \bar{Y}_{ij.})^2 \tag{D.13}$$

These are the expressions for the SS terms in Table 5.3. Statistics software packages typically use different froms of these expressions that are more suited to numerical calculation.

Appendix E

Raw Data Table

Table E.1 shows the raw data for the 32 runs in all 8 cells. Measurements removed as outliers are marked with a double dagger (‡). Measurements removed to make the cells equal (n=27), as explained in Chapter 5, are marked with a dagger (†).

All measurements are time intervals in milliseconds between center target taps, as described in the text, and are precise to the nearest hunderedth of a second (10 milliseconds).

Subject 1				Subject 2			
Tight-On	Loose-On	Tight-Off	Loose-Off	Tight-On	Loose-On	Tight-Off	Loose-Off
5410	5070	9730	6000	5290	5000	5170	‡15620
5100	4350	4950	5290	‡11380	5300	5260	5110
4640	5060	5100	6110	‡7990	‡9170	‡7680	8890
5160	4680	4950	8070	5490	4650	5070	5060
5470	5260	5160	6000	5480	4890	4900	5010
5110	5410	5700	5790	5460	5050	5260	5450
4950	4840	4840	5790	7050	5270	5170	5590
5370	5170	‡11900	5780	‡9310	5090	5360	8680
4940	4750	5180	5590	6100	‡7930	‡7770	5380
5010	4930	5630	5570	4950	‡8320	4470	7590
4740	5530	4960	8310	5380	5050	5160	4510
5370	4550	4970	5180	5580	5060	4590	4790
4580	5180	5690	5620	4850	4940	4750	6320
4740	4920	4850	4740	4860	5690	6100	7060
‡8680	5160	4770	5790	5230	4930	5000	5790
5800	4280	7420	5790	4950	7480	5500	6430
5170	4840	5080	4990	4900	‡10970	‡8670	4980
4950	5170	5910	5370	‡7960	5440	4890	5380
4460	4460	‡11270	5060	4780	4790	4840	5220
4640	4950	4680	5790	5270	4570	5690	4930
4160	5210	5070	4950	4850	4980	4750	‡11440
4550	4940	4800	9010	4780	4830	4950	5070
4670	5380	5070	5790	5180	4650	4470	5170
4640	5370	5360	7270	5160	5540	4860	5010
4450	5890	4990	5590	5080	‡9470	‡10170	4750
4900	4420	5090	7960	5000	4690	4860	7560
4740	4280	5260	‡13550	4950	5530	5160	6120
4460	†4630	4960	5280	4860	5280	5310	5170
†5180	†4690	5150	†8990	5200	4980	4540	5590
†4540	†4100	6830	†5170	4950	4810	4360	†6100
†4350	†4170	†5050	†7740	5060	4670	4970	†7970
†4860	†4480	†5280	†9000	†5200	4790	†4690	†4770

Table E.1: Raw Data

Outliers are marked with a double dagger (‡), and measurements removed to make the cells equal are marked with a dagger (†)