Vision Based Navigation and Control for Unmanned Free-Flying Remote Vehicles

by

Harald Jürgen Weigl

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

September 1992

Signature of Author _____
Department of Aeronautics and Astronautics
August 21, 1992

Certified by _____
Professor Harold L. Alexander
Thesis Supervisor

Accepted by _____
Prof. Harold Y. Wachman
Chairman, Department Graduate Committee

Vision Based Navigation and Control for Unmanned Free-Flying Remote Vehicles

by

Harald Jürgen Weigl

Submitted to the Department of Aeronautics and Astronautics
on August 21, 1992, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

The development and implementation of a free-flying robot navigation and control system, using vision-based position and attitude sensing, is described. The system has been implemented in real-time for an actual free-flying underwater robot,the Submersible for Telerobotic and Astronautical Research (STAR), with standard, inexpensive computer hardware. Excellent performance and robustness characteristics are achieved for a variety of applications, including automatic station-keeping and small controlled maneuvers. Experimental results are presented indicating the precision, accuracy, and robustness to disturbances of the vision-based control system. The study proves the feasibility of using vision-based robot control and navigation and provides a foundation for developing the system for more general tasks.

The complex vision sensing problem is reduced through linearization to a simple algorithm, fast enough to be incorporated into a real-time vehicle control system. Vision sensing is structured to detect small changes in vehicle position and orientation from a nominal positional state relative to a specially designed navigation target. The vision target is used to provide sensitivity to six degrees of freedom of motion and to simplify image processing. The system uses a constant, linear inversion matrix to measure the vehicle state from the locations of navigation features in an image.

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

Acknowledgments

First of all, I wish to thank my best friend throughout the good times and not so good ones here at MIT. Shirley, your patient support over the past two years through the all-nighters and times when nothing seemed to work right, has kept me going. I will always love you very much.

I must also express my deepest gratitude to Sandy Alexander for the wisdom, support, humor, and understanding he has shared with me. Thank you for giving me the chance to work on a truly exciting and rewarding project and to help make your plans with STAR (which seemed very ambitious back when it was just a floating pizza oven) a reality. I wish you continued success in your future career.

The quality that makes MIT special are the people who work and play here. Thank you to my friends in LSTAR who made the many hours in Bldg. 41 and at the pool fun: Dean, Willy, Dewey, Ali, Anna, and Beth. I also owe a special note of thanks to: Mike (Take good care of STAR. I hope that the da Bulls - but NOT da Bears - win many more championships.), Paul (We made a great team. I wish you the best of luck down in Maryland and can't wait to hear about the incredible things you'll be doing.), Mark (Even though you're a Jays fan I wish you the best of luck with your virtual work. I'd much rather play darts on your team than against you.), and to Kurt & Matt down at OSC (Thank you Kurt, my massively pumped friend, for building an amazing robot. Matt, thank you for always being able to make me laugh with one of your stories or descriptions of your incredible dreams.)

I also have to thank my office mates in 33-407. The debates about life and the horse/dart games were a great stress relief. Good luck to you all, Stéphane, Kevin, Will, Didier, Sasi, Pierre, Errol, Jeff, and of course Alex. I also wish the very best to all the brothers and little sisters of ZBT especially the AZ's.

Finally, I must thank my family for all their love and support that helped me get here in the first place. I'll never forget our talk by the barbecue in summer of 1986 back when MIT seemed like a very scary place. Danke, Mammi, Papi, und Claudia!

Contents

List of Figures

Chapter 1

Introduction

Mankind's exploration has led us beyond the heavens into space and past our shores into the depths of the oceans. As interest in these exciting new worlds has grown, so has the need to develop robot technology to help humans overcome the dangers and high costs of working in these harsh environments. A main concern of the 1990 Advisory Committee on the Future of the U.S. Space Program was the risk and cost encountered by humans working in space. One of the study's major recommendations was for NASA to develop robotics technology to be used in the exploration and settlement of space.

Free-flying robots are being extensively studied for a variety of applications from satellite servicing to exploration of the ocean floors. Robots may perform these tasks either autonomously or by teleoperation with the aid of a human operator. In either case, it is crucial to have a precise measure of the vehicle's position and orientation as it interacts with its environment. For such functions as station keeping, docking, manipulator arm control, and obstacle avoidance, an accurate measure of the robot's positional state is essential.

One promising method for navigation of both space and underwater robots is based on machine vision using an on-board video camera and digital image processing hardware. A vision sensing system is relatively simple to implement in terms of hardware since most free-flying robots will already be carrying video cameras for teleoperation but also has the power to handle a wide range of applications. The

underlying assumption for such a system is that distinguishable features, of known physical shape and position in the environment, exist within the camera's field of view which can be used as navigation targets. This assumption can be met for most "interesting" tasks requiring precise position control by the physical properties of the environment (for example tracking a satellite's markings during a rendezvous) or with an artificially added navigation target.

The complexity and inflexibility of most current vision algorithms have prevented their use for real-time free-flying robot navigation. To be useful for robot navigation, a vision system must be fast and simple enough to be implemented on current computer technology and still have the power to perform useful robot tasks. This thesis presents a simple, yet accurate, vision-based navigation and control system implemented in real time for a free-flying underwater robot applicable to a variety of robotic applications.

1.1 Free-Flying Robot Navigation

Free-flying robots such as JASON and SEA SQUIRT[1] are being used today to explore underwater environments. Underwater vehicles can use a variety of sensors for navigation including inertial sensors, water pressure sensors, and sonar. Although there are currently no free-flying robots in space, there has been extensive research on their development using underwater robots. The Space Systems Lab (SSL), now at the University of Maryland, has been a pioneer in the use of neutrally-buoyant underwater space robot simulators. The SSL has studied the use of inertial sensing and acoustic ranging for navigation of its free-flying underwater robots [5]. Types of sensors proposed for robot navigation in the zero-gravity environment of space include inertial sensors, radar, laser range finders and radio beacons.

Although these space-based sensors are similar to underwater sensors, they are not directly transferable from one environment to the other. Vision-based position and orientation sensing can bridge the gap between the two environments. This is

[1]Built by the Woodshole Oceanographic Institute, JASON has been used to study the wreckage of the *Titanic*. SEA SQUIRT is an autonomous vehicle developed by the MIT Sea Grant program to study frozen lakes.

a great advantage for underwater space robot simulation since the actual technology to be used in space can be tested on Earth. Since vision sensing is passive, it has the added advantage of being easier to implement than an acoustical or radio beacon system.

The MIT Laboratory for Space Teleoperation and Robotics (LSTAR), directed by Prof. Harold Alexander, has built an underwater space robot simulator which has evolved from earlier SSL vehicles: the Submersible for Telerobotic and Astronautical Research (STAR). Unlike the vehicles from the SSL, STAR's navigation and control system employs a vision-based sensing system which is described in this thesis.

1.2 Submersible for Telerobotic and Astronautical Research (STAR)

STAR, shown in Figure 1-1 in the MIT Alumni Pool, is an underwater version, in form and function, of future free-flying space robots. The vehicle is neutrally buoyant in both depth and orientation in order to simulate zero-gravity six degree of freedom (DOF) movement as in space. The major discrepancy between the underwater and space environments is the mass and drag of the water an underwater vehicle interacts with as opposed to the vacuum of space. However, neutral buoyancy simulation is still an effective method for simulating space based operations on Earth [1]; the Neutral Buoyancy Facility, at the Marshal Space Flight Center, is used extensively by NASA for astronaut training.

LSTAR is using STAR to develope and test a wide range of space robot technology; the laboratory's areas of research include teleoperated vehicle control, autonomous control, and the use of manipulator arms. Pan and tilt camera platforms, helmet mounted stereoscopic displays, and the use of visual reference cues are being studied to aid teleoperated control. The automatic vehicle control research has concentrated primarily on vision-based sensing for navigation and control.

The vehicle's body, roughly 4 ft x 3.75 ft x 3 ft with the outside structure attached, houses the robot's on-board electronics and battery compartments. STAR is propelled

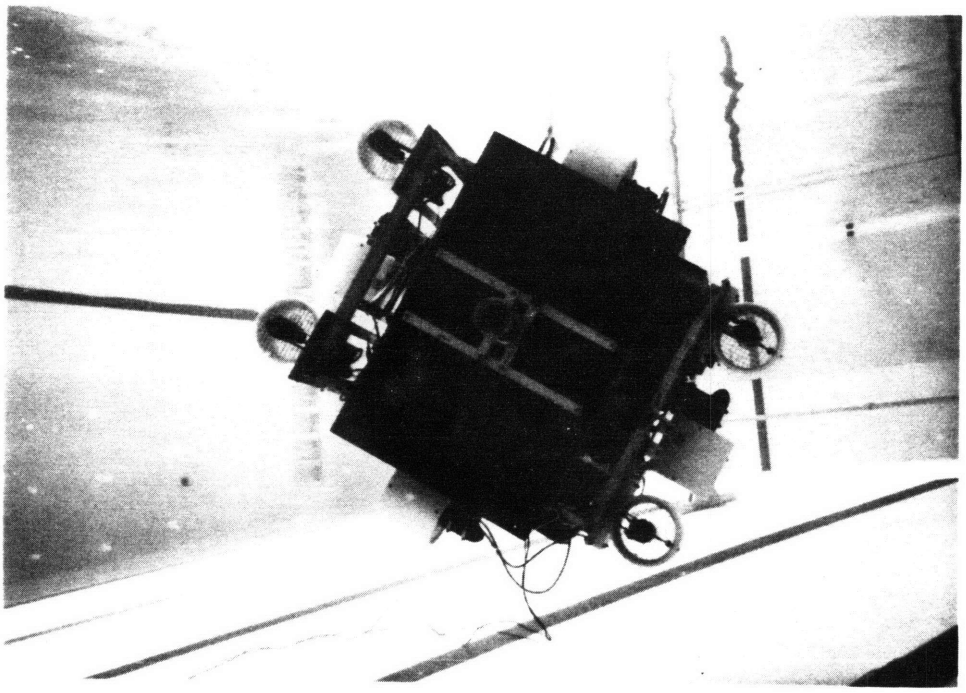


Figure 1-1: **STAR in MIT Alumni Pool**

by eight propellor motors which allow full six degree of freedom vehicle maneuvering[2]. The robot is connected to a surface control station through an umbilical linked to the vehicle's on-board computers. For a more detailed description of STAR's subsystems, see Section 3.1.

1.3 STAR Vision-Based Navigation and Control

STAR's current vision-based control system has evolved from earlier LSTAR machine vision research. St. John-Olcayto studied the use of high contrast vision targets for vision-based position and orientation sensing [14]. Azarbayejani developed an elegant vision navigator capable of tracking a navigation target based on an extended Kalman filter of a free-flying robot [6]; simulations of Azarbayejani's system running on an Apple Macintosh IIx ran at a speed of 2.5 steps per second.

The scope of the vision-based navigation and control system described here has been simplified in order to create a workable real-time system using STAR's computer hardware. Nevertheless, the system provides good and robust performance for a number of useful and interesting robot tasks such as station keeping and making the vehicle a stable platform for a manipulator arm. The system presented here also provides the foundation for adding expanded capabilities such as tracking an arbitrary target while flying a complex trajectory.

STAR's vision-based control system uses a specially designed navigation target (similar to the one used by Azarbayejani [6]) designed to provide motion sensitivity and to simplify vision processing as described in Section 2.1. In order to linearize the complex vision sensing problem, the control system operates about a predefined nominal vehicle position and orientation relative to the target; the vision sensing system is used to detect small angular and translational deviations from the nominal state. In the current framework, the vision-based control system can be used for autonomous station-keeping, small closed-loop teleoperated maneuvers, and flying

[2]The propulsion system for an underwater vehicle such as STAR is vastly different than those of space robots. Propellor motors have a continuous range of thrust but spacecraft reaction jets provide only "on/off" thrust.

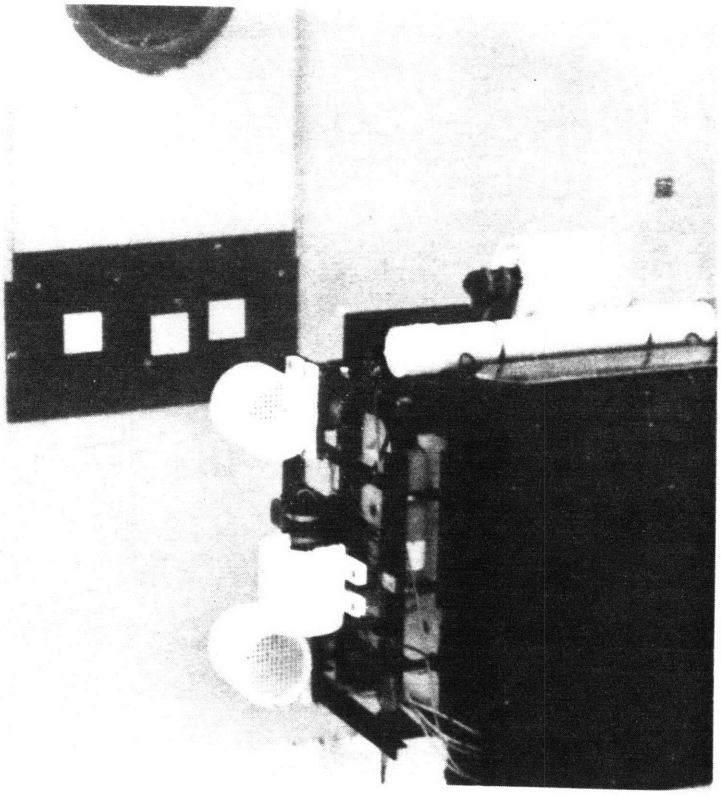


Figure 1-2: **STAR and Fixed Navigation Target**

trajectories from one nominal state to another. A picture of STAR in station-keeping mode opposite the navigation target is shown in Figure 1-2.

At each time step, the image of the target from STAR's video camera is digitized in real-time with a standard frame-grabber board and the digitized image is processed to compute the vehicle's position and orientation relative to the target. These vision-based state measurements are combined with angular rate sensor measurements to provide the feedback for the robot position and attitude control system. A digital controller updates commands to the vehicle's eight thrusters based on the current error between the commanded state input and the measured states. The thrusters, in turn, reposition and reorient the vehicle based on the control commands. A block diagram of the vision-based controller is shown in Figure 1-3.

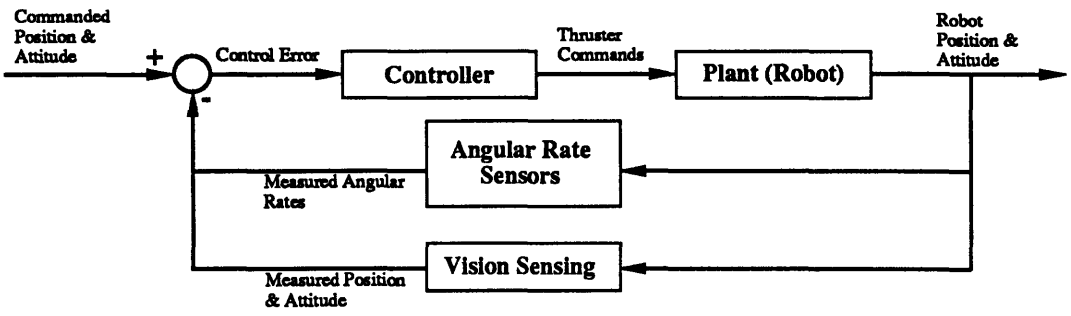


Figure 1-3: **Navigation and Control System Block Diagram**

1.4 Thesis Overview

The two major components of the free-flying vision-based navigation and control system are described in Chapters 2 and 3. Chapter 2 describes vision-based position and attitude sensing; the specially designed navigation target and vision sensing algorithm are also described in detail. The robot position and attitude control system is developed in Chapter 3. Specifics of the software and hardware implementation of STAR's vision-based control and navigation system are also covered there. Experimental results from tests of the final vehicle control system are presented in Chapter 4. Applications of the vision-based navigation and control system for free-flying robots and recommendations for further development are discussed in Chapter 5.

Chapter 2

Vision-Based Position and Orientation Sensing

This chapter describes a linearized method for detecting small deviations in position and orientation of a free-flying robot using images of known navigation features. The real-time system is developed using STAR's navigation target; however, the basic derivation also applies to arbitrary navigation features. Although some simplifications and assumptions have been made, the methodology can be expanded to more general cases as discussed in Section 5.1.

2.1 Vision Navigation Principles

The underlying function of vision based navigation for a free-flying robot is to determine the motion of the vehicle from a changing video image of the robot's environment. For example, assume an underwater vehicle's camera is pointed at the uniform stripes on the floor of a pool; as the vehicle moves in a direction perpendicular to the stripes, the image of the stripes would appear to move in the opposite direction in each successive video frame. By measuring the changing locations of the stripes in the image, the motion of the robot relative to the stripes can be determined. For vision-based free-flying robot navigation, there must be enough navigation features (for example, the edges of the pool stripes) in the camera's image to detect six degree

of freedom motion. In the previous example, if the vehicle were to move along the direction of the stripes, that motion would not be detectable since the location of the stripes' edges would not change in successive frames.

The function of the vision-based navigation system is to determine the change in camera position and orientation, or pose, from the changing image of the navigation features being used. Linearization about a nominal camera pose simplifies this complex relationship and leads to a very efficient real-time implementation which is described below.

2.2 Measurement Equations

For vision-based navigation, it is necessary to first be able to determine where a point in 3-D space (in this case, a navigation feature point) will fall on a video image. With certain simplifying assumptions, the problem of locating points on the image is very straightforward and can be solved through simple geometry.

2.2.1 Camera Model

The measurement equations which are the basis of the vision navigation system are derived from the ideal pinhole camera model which is a simple, yet accurate, model for most lensed cameras [12]. The camera is modelled as an enclosed box containing a light-sensitive plate[1] known as the *image plane*; the lens is modelled as a tiny pinhole located at a fixed distance from the image plane. All light coming into the camera is assumed to pass through the pinhole and land on the image plane. The corresponding location of any point in 3-D space on the 2-D image plane can therefore be found by tracing a single light ray from the point of interest, through the pinhole, and onto the image plane. The pinhole camera model is illustrated in Figure 2-1.

Figure 2-1 also illustrates some common machine vision terminology. The pinhole is known as the *center of projection* (COP). The *optical axis* is defined as the line

[1]Most small cameras being used for robotic applications today have a CCD imager.

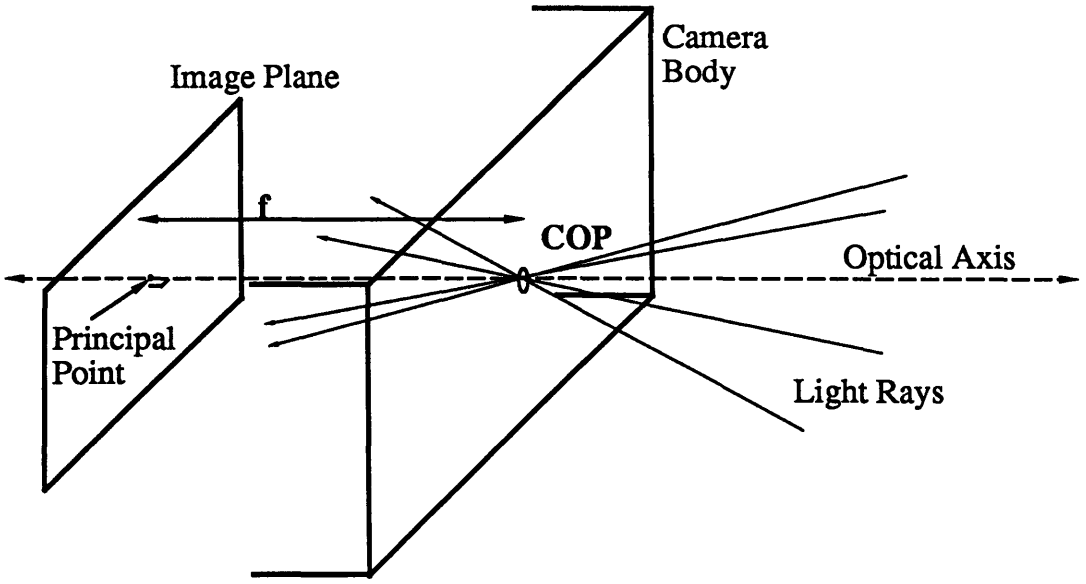


Figure 2-1: **Pinhole Camera Model**

which is perpendicular to the image plane and passes through the COP. The distance along the optical axis between the image plane and COP is the *principal distance* or *effective focal length* (f). The point of intersection between the optical axis and the image plane is the *principal point.*

2.2.2 Image Projection Equations

The geometry used to locate the corresponding position of a point in space on the image plane is derived from the pinhole camera model. Figure 2-2 shows a point, P, in 3-D space and its corresponding location on the image plane. The figure also illustrates the inversion of the image due to projection. For most applications, it is convenient to work with an equivalent image plane located at $+f$ along the optical axis which eliminates the confusion of dealing with an inverted image (as shown in Figure 2-3). The equivalent image plane, from now on refered to simply as the image plane, and the corresponding image point locations are shown in Figure 2-3.

The right-handed reference frame with axes (X_c, Y_c, Z_c) that is used to locate points on the image is also shown in Figure 2-3. The origin of the frame is located

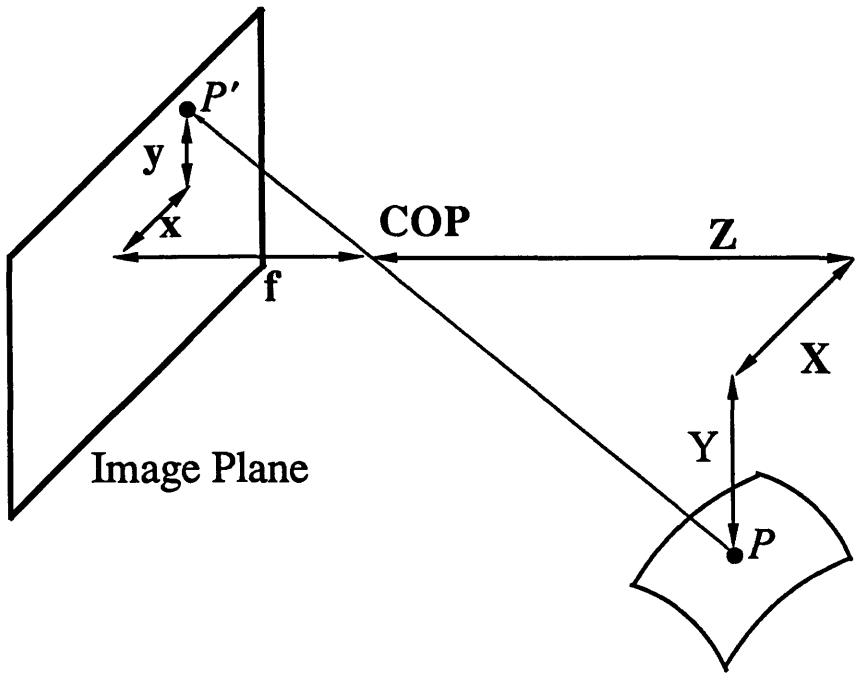


Figure 2-2: **Projection of Point in Space onto Image Plane**

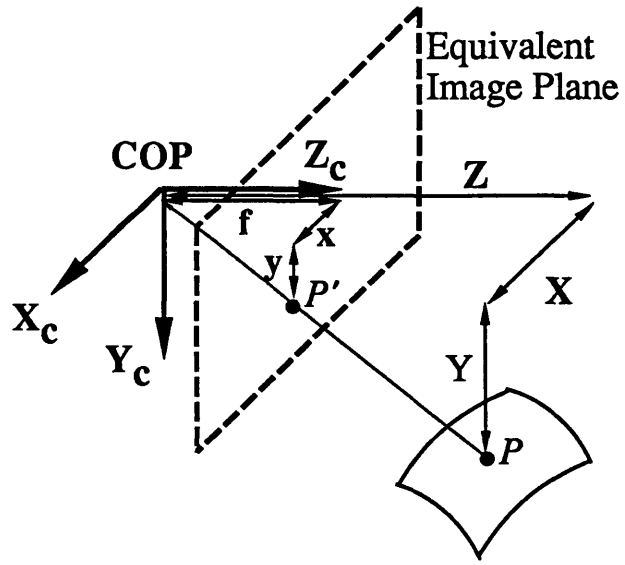


Figure 2-3: **Noninverted Image Point Locations**

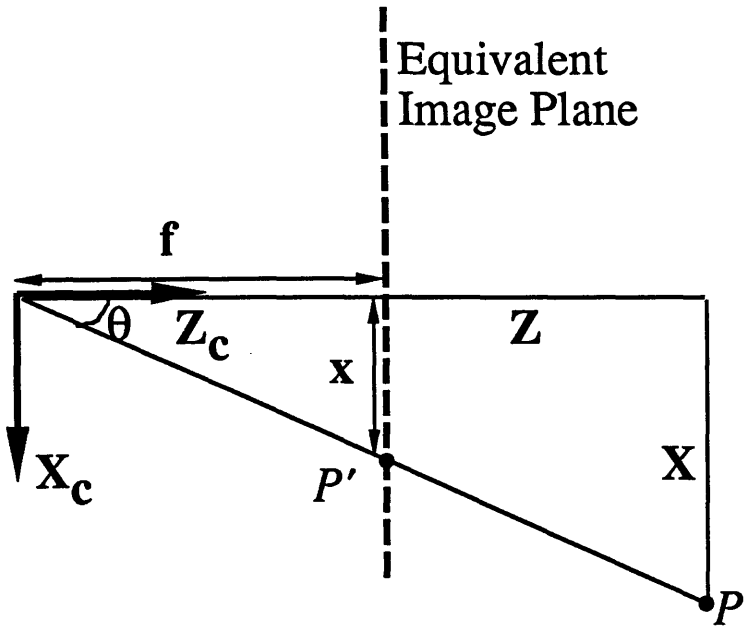


Figure 2-4: X_c **Projection Component**

at the COP with the Z_c axis pointing out of the camera along the optical axis. The image plane is located at $Z_c = +f$ and is parallel to the X_c and Y_c axes; X_c points "right" and Y_c points "down" if the camera is held upright.

Reducing the projection problem to the plane defined by $Y_c = 0$ shows the geometric relationship for finding x (Figure 2-4). The two triangles with the same included angle, Θ, yield the following *Image Projection Equation*:

$$x = \frac{fX}{Z} \tag{2.1}$$

Similarly for the Y_c component:

$$y = \frac{fY}{Z} \tag{2.2}$$

2.3 Vision-Based Camera Pose Computation

When viewing an object, the position and orientation of the camera determines where the image of the object will fall on the image plane. For example, when viewing a point in space (such as a small light source) straight-on, the image of the point will be in the center of the image plane. As the camera is turned away from the point, the image of the point will move away from the center of the image plane. Given the pose of the camera, the location of an image point can be calculated through the image projection equations.

Vision-based position and orientation sensing is the inverse problem to the previous example: given the location of image feature points, what is the pose of the camera? The methodology used measures small angular and positional deviations from a predetermined nominal camera pose. Using a fixed nominal state allows all the information of the navigation features (Section 2.3.2) to be pre-computed, thereby saving a great deal of computational time within each control loop. The nominal state can be used as a fixed state for automatic station-keeping, or several nominal states can be pre-computed and linked together for flying simple trajectories.

2.3.1 Camera State Computation

The general derivation of the camera pose computation is based on tracking N arbitrary points at (x_i, y_i) in the image plane. Several feature points are used to overcome tracking errors and to detect six degree of freedom camera motion, as discussed below.

At each time step, the current locations of N points in the image plane are measured. These point locations form a $(2N)$ x 1 vector, $\vec{\mathbf{p}}$, of the following form:

$$\vec{p} = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \vdots \\ x_N \\ y_N \end{bmatrix} \tag{2.3}$$

The pose of the camera relative to the target is given by the vector, $\vec{x}_c$, as defined by the camera reference frame in the nominal state:

$$\vec{x}_c = \begin{bmatrix} X \\ Y \\ Z \\ \alpha \\ \beta \\ \gamma \end{bmatrix} \tag{2.4}$$

The first three elements of $\vec{x}_c$ describe the translational displacement of the camera reference frame. The last three elements give its attitude in terms of three Euler angles. Since for this application the vision navigator only deals with small angular displacements, noncommutativity and singularities associated with Euler angles are neglected[2]. The directions of the three Euler angles are shown in Figure 2-5. These angles describe the roll, pitch, and yaw motions of the camera which are defined to be consistent with roll, pitch, and yaw as usually applied to aerospace vehicles. However, due to the conventions of assigning a camera reference frame, the Euler angles are defined about different axes than those normally used for body reference frames (see Section 2.4.1).

The nominal location of each of the feature points on the image plane can be pre-

[2]For vision-based tracking during large maneuvers, it would be necessary to use an alternative representation, such as a quaternion, for the angular state.

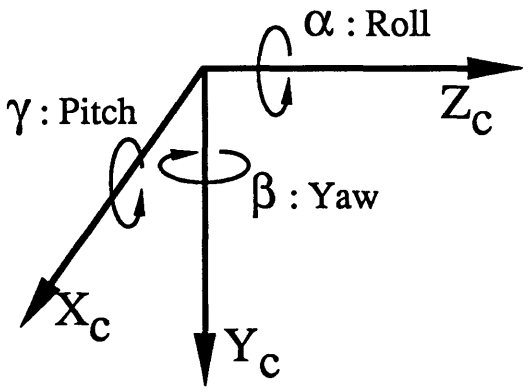


Figure 2-5: **Directions of Camera Frame Euler Angles**

computed from the image projection equations with the camera at its nominal state since the camera focal length, the dimensions of the target, and range to the target are all known. The nominal point locations are expressed below by the nonlinear function, h, in terms of the nominal state, $\vec{\mathbf{x}}_{\mathbf{c_o}}$:

$$\vec{\mathbf{p}}_{\mathbf{o}} = h(\vec{\mathbf{x}}_{\mathbf{c_o}}) \tag{2.5}$$

However, the nominal point positions alone are not of great interest since the goal is to measure deviations from the nominal camera pose. This is accomplished through a first order Taylor series expansion about the nominal state which is reasonable since state deviations are assumed to be small. This expansion gives a linear relationship which determines the changes in image point locations for small changes in vehicle state.

Arbitrary point locations are first expressed as small deviations from their nominal values and then expanded:

$$\begin{aligned} \vec{\mathbf{p}} &= \vec{\mathbf{p}}_{\mathbf{o}} + \delta\vec{\mathbf{p}} \\ &= h(\vec{\mathbf{x}}_{\mathbf{c_o}}) + \delta\vec{\mathbf{p}} \end{aligned}$$

$$\approx \; h(\vec{\mathbf{x}}_{\mathbf{c_o}}) + \left[\frac{\partial h}{\partial \vec{\mathbf{x}}_{\mathbf{c}}}\right]_{\vec{\mathbf{x}}_{\mathbf{c_o}}} \delta\vec{\mathbf{x}}_{\mathbf{c}} \tag{2.6}$$

Simplifying Equation 2.6 gives the following approximation for deviations in edge point locations for small changes in camera state:

$$\delta\vec{\mathbf{p}} \approx H\delta\vec{\mathbf{x}}_{\mathbf{c}} \tag{2.7}$$

2.3.2 Geometric Derivation of Sensitivity Matrix

This section explains a simple method for geometrically computing H, known as the *sensitivity matrix*, for the N image feature points of interest.

Structure of Sensitivity Matrix

Equation 2.7 is expanded below to show the structure of the sensitivity matrix for one image point at (x_i, y_i):

$$\begin{bmatrix} \delta x_i \\ \delta y_i \end{bmatrix} = \begin{bmatrix} \frac{\partial x_i}{\partial X} & \frac{\partial x_i}{\partial Y} & \frac{\partial x_i}{\partial Z} & \frac{\partial x_i}{\partial \alpha} & \frac{\partial x_i}{\partial \beta} & \frac{\partial x_i}{\partial \gamma} \\ \frac{\partial y_i}{\partial X} & \frac{\partial y_i}{\partial Y} & \frac{\partial y_i}{\partial Z} & \frac{\partial y_i}{\partial \alpha} & \frac{\partial y_i}{\partial \beta} & \frac{\partial y_i}{\partial \gamma} \end{bmatrix}_{\vec{\mathbf{x}}_{\mathbf{c_o}}} \begin{bmatrix} \delta X \\ \delta Y \\ \delta Z \\ \delta \alpha \\ \delta \beta \\ \delta \gamma \end{bmatrix} \tag{2.8}$$

Motion Sensitivity in the X_c Direction

Equation 2.1 is used to directly compute the sensitivity of the image point locations to translational motion for the i^{th} point:

$$\left.\frac{\partial x_i}{\partial X}\right|_{\vec{\mathbf{x}}_{\mathbf{c_o}}} = -\frac{f}{Z_{o_i}} \tag{2.9}$$

$$\left.\frac{\partial x_i}{\partial Y}\right|_{\vec{\mathbf{x}}_{\mathbf{c_o}}} = 0 \tag{2.10}$$

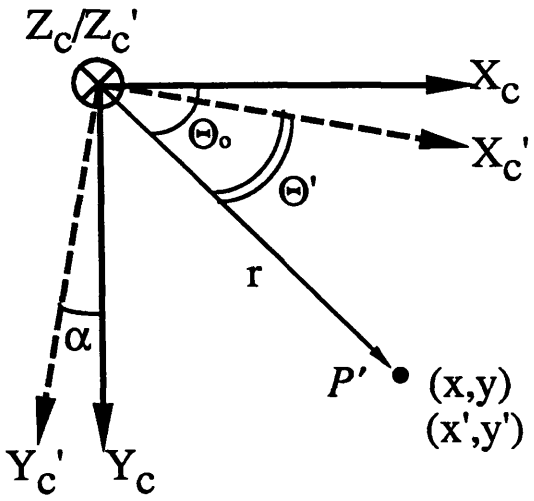


Figure 2-6: **Image Sensitivity to Camera Roll**

$$\left.\frac{\partial x_i}{\partial Z_i}\right|_{\vec{\mathbf{x}}_{\mathbf{c_o}}} = \frac{fX_{o_i}}{Z_{o_i}^2} = \frac{x_{o_i}}{Z_{o_i}} \tag{2.11}$$

To determine the signs of the sensitivity terms, it is useful to think in terms of the physical problem. For example, as the camera moves along $+X_c$, the image will move in the opposite direction; as the camera moves toward an object along the $+Z_c$ axis, the image of the object will appear to grow.

Figure 2-6 indicates the geometry necessary to compute the sensitivity of image feature points to camera roll motion. The perspective of the figure is looking forward from the COP at the feature point. The point, P', represents the corresponding image of the feature point. If the camera rolls right by α, the image plane and camera reference frame axes are thereby rotated by the same amount. The following equations compute the new location of the point, P', in the rotated frame under the assumption that the roll angle is small:

$$
\begin{aligned}
\Theta_o &= \Theta' + \alpha \\
x &= r\cos\Theta_o \\
y &= r\sin\Theta_o \\
x' &= r\cos\Theta'
\end{aligned}
$$

$$
\begin{aligned}
x' &= r\cos(\Theta_o - \alpha) \\
&= r(\cos\Theta_o\cos\alpha + \sin\Theta_o\sin\alpha) \\
&\approx r(\cos\Theta_o + \sin\Theta_o\alpha)
\end{aligned}
$$

From the expression for the new location of P', the sensitivity of X_c locations to roll motion can be calculated:

$$
\left.\frac{\partial x_i}{\partial\alpha}\right|_{\vec{\mathbf{x}}_{\mathbf{c_o}}} = r\sin\Theta_{o_i} = y_{o_i} \tag{2.12}
$$

Pitch rotation will change the location of the image points in the X_c direction since the range to the target will be changed by pitch motion. The sensitivity of the range, moving from Z to Z', due to a positive pitch rotation (as illustrated in Figure 2-7) is computed below given that the distance between the image point and the COP, r, is unchanged by the rotation:

$$
\begin{aligned}
\Theta' &= \Theta_o + \beta \\
Z &= r\cos\Theta_o \\
Y &= r\sin\Theta_o \\
Z' &= r\cos\Theta'
\end{aligned}
$$

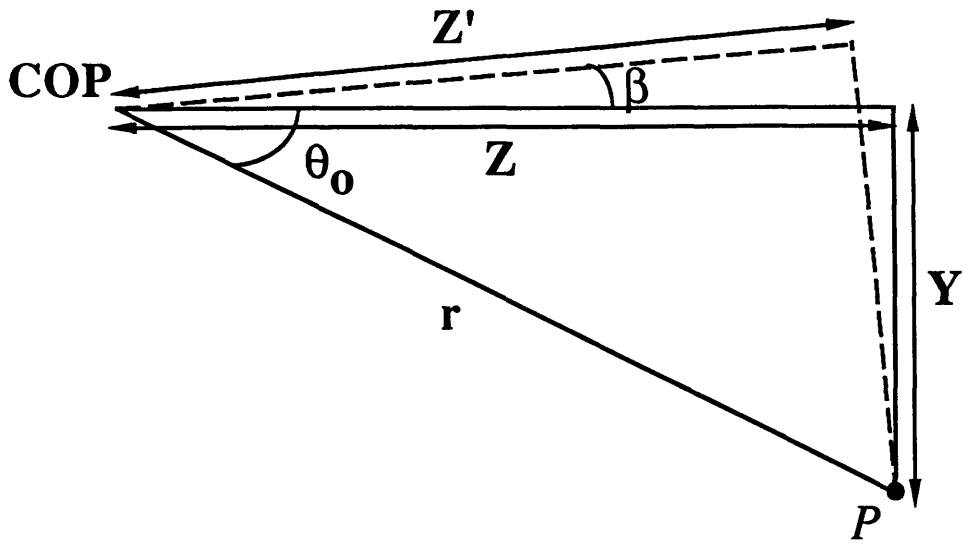


Figure 2-7: **Image Sensitivity to Camera Pitch**

$$\begin{aligned} Z' &= r\cos(\Theta_o + \beta) \\ &= r(\cos\Theta_o\cos\beta - \sin\Theta_o\sin\beta) \\ &\approx r(\cos\Theta_o - \sin\Theta_o\beta) \\ \frac{\partial Z}{\partial\beta} &= -r\sin\Theta_o \\ &= -Y = -\frac{Zy}{f} \end{aligned}$$

From this relationship between pitch and Z, the sensitivity of point locations along the X_c axis to pitch can be found:

$$\begin{aligned} \left.\frac{\partial x_i}{\partial\alpha}\right|_{\mathbf{x_{co}}} &= \left.\left(\frac{\partial x_i}{\partial X}\frac{\partial X}{\partial\beta} + \frac{\partial x_i}{\partial Z}\frac{\partial Z}{\partial\beta}\right)\right|_{\mathbf{x_{co}}} \\ &= 0 + \left(\frac{-x_{o_i}}{Z_{o_i}}\right)\left(-\frac{Z_{o_i}y_{o_i}}{f}\right) \\ &= \frac{x_{o_i}y_{o_i}}{f} \end{aligned} \tag{2.13}$$

Figure 2-8 indicates that a small yaw rotation by γ about the COP effectively

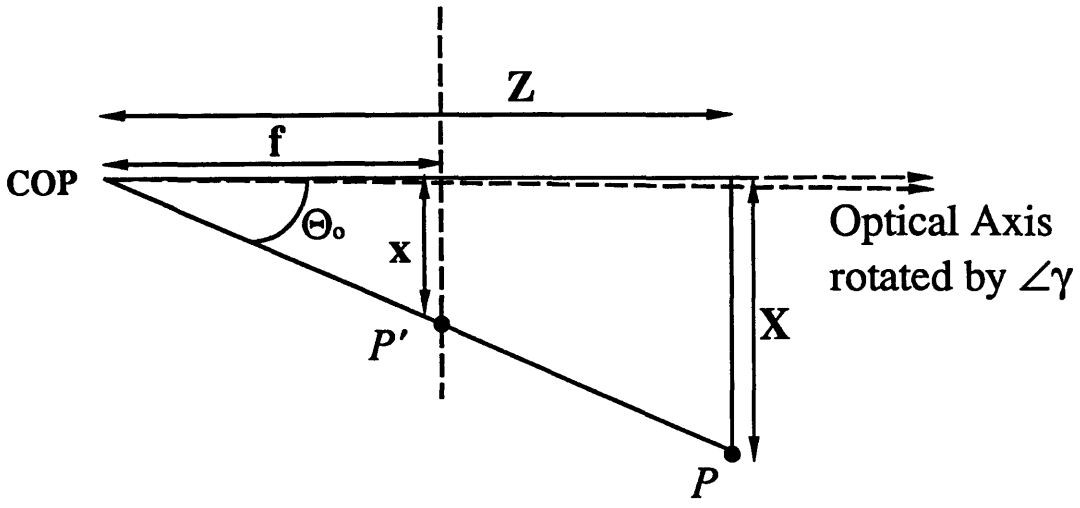


Figure 2-8: **Image Sensitivity to Camera Yaw**

changes the angle between the optical axis and the light ray from a point in space. The image of a point will move in the $-X_c$ direction for a positive yaw motion. The sensitivity of image point locations along X_c to a small yaw motion is computed below again in terms of the nominal location in the image :

$$\begin{aligned} \frac{x}{f} &= \tan\Theta_o \\ \left.\frac{\partial x_i}{\partial \gamma}\right|_{\mathbf{x_{c_o}}} &= -f\sec^2\Theta_o \\ &= -f\left(1+\left(\frac{x_{o_i}}{f}\right)^2\right) \end{aligned} \tag{2.14}$$

Motion Sensitivity in the Y_c Direction

By symmetry, the sensitivity of image point locations in the Y_c direction are derived in the same manner. In this case, the measured locations of points are unaffected by X translation:

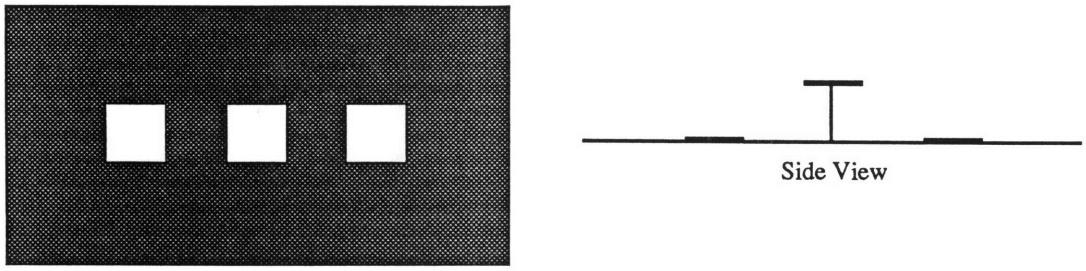


Figure 2-9: **Vision Navigation Target**

$$
\begin{aligned}
\left.\frac{\partial y_i}{\partial X}\right|_{\vec{\mathbf{x}}_{\mathbf{co}}} &= 0 \\
\left.\frac{\partial y_i}{\partial Y}\right|_{\vec{\mathbf{x}}_{\mathbf{co}}} &= -\frac{f}{Z_{o_i}} \\
\left.\frac{\partial y_i}{\partial Z}\right|_{\vec{\mathbf{x}}_{\mathbf{co}}} &= \frac{y_{o_i}}{Z_{o_i}} \\
\left.\frac{\partial y_i}{\partial \alpha}\right|_{\vec{\mathbf{x}}_{\mathbf{co}}} &= -x_{o_i} \\
\left.\frac{\partial y_i}{\partial \beta}\right|_{\vec{\mathbf{x}}_{\mathbf{co}}} &= f\left(1+\left(\frac{y_{o_i}}{f}\right)^2\right) \\
\left.\frac{\partial y_i}{\partial \gamma}\right|_{\vec{\mathbf{x}}_{\mathbf{co}}} &= -\frac{x_{o_i} y_{o_i}}{f}
\end{aligned}
\tag{2.15}
$$

Sensitivity Matrix for STAR's Navigation Target

The navigation features which are used to compute the camera pose with STAR's vision-based navigation system are parts of a specially designed navigation target. A schematic diagram of the vision target used by STAR's real-time system is shown in Figure 2-9. The target was designed to simplify the image processing needed to determine the locations of its image features. It allows fast vision processing, using STAR's current hardware.

The target consists of a row of three white squares, 15 cm wide and separated by 15 cm, centered on a black background. The center square is raised by 15 cm from the plane of the other squares providing the necessary sensitivity to translational and rotational motion as discussed below.

A key aspect of the target is the high contrast between the black and white regions. This reduces the image processing of the target to finding the edges between the target's white and black regions. Reducing the search for image features to simple edge detection saves a great deal of the computational time needed to access the digital video data. For example, since only edges are of interest in the image, it is not necessary to scan every single pixel in the center of a white square or far within the black background region.

The target's straight edges allow for a further simplification of the actual image processing by reducing the search for edge locations to one dimension. Since the vision-based navigation system is designed to detect small deviations from a nominal positional state, the targets edges are not expected to move very much in the image so it is only necessary to scan perpendicularly across edges. The edge detection algorithm searches across horizontal edges to find changes in edge location along the vertical image axis and similarly across vertical edges to find edge location changes along the horizontal image axis.

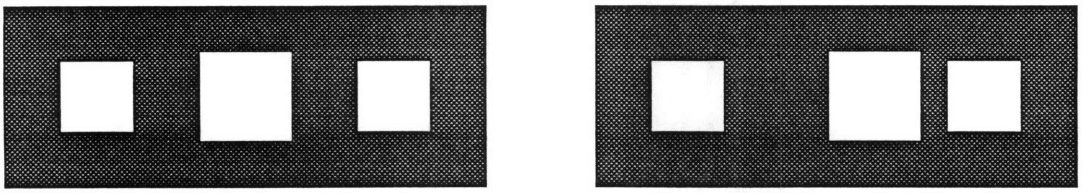
The simplest black and white edge detection algorithm is know as *thresholding* which compares the brightness of a picture cell (*pixel*) in the image to a constant threshold value. If the pixel is brighter than the threshold, it is assumed to be white and if not it is black. The major disadvantages of simple image thresholding are sensitivity to video image noise and the gradual brightness transition across the actual image of an edge. There are more sophisticated techniques for edge detection, such as the method used by Azarbayejani of finding the maximum of the brightness gradient across an edge [6]. However such techniques require increased search time due to their complexity. Since computation time is a major concern for implementing STAR's real-time system, thresholding was used with success in various lighting conditions during both day and night pool tests.

In the framework of the previous derivation, N measurements are taken along the target edges to determine the deviations from the nominal positions. The navigation target edge locations are measured only perpendicularly across an edge (the first half of the samples for the vertical edges in the X_c direction and the other half in the Y_c direction for horizontal edges). For STAR's navigation target, the edge deviation vector, $\delta\vec{\mathbf{e}}$ (expressed as, $\delta\vec{\mathbf{p}}$, in the previous general derivation) is given below:

$$\delta\vec{\mathbf{e}} = \begin{bmatrix} \delta e_1 \\ \delta e_2 \\ \vdots \\ \delta e_N \end{bmatrix} \tag{2.16}$$

The appropriate sensitivity matrix must be computed for STAR's vision target edges, which are the navigation features of interest. For a given nominal camera state with the squares at arbitrary angles, the edge deviation sensitivity can be computed by calculating the dot product of the edge point sensitivities in the X_c and Y_c directions and the perpendicular direction to each edge. For a nominal camera pose directly opposite the center of the target with the target's squares aligned with the X_c and Y_c directions, the previously derived sensitivity equations can be applied directly (since the X_c and Y_c directions are perpendicular to the edges). Equation 2.7 is expanded below, term by term, to show the complete structure of the sensitivity matrix for the measured vertical and horizontal edge deviations on the navigation target:

$$\delta\vec{\mathbf{e}} \approx H\delta\vec{\mathbf{x}}_{\mathbf{c}}$$



View from Camera Directly Opposite Target Center

View from Rotated Camera to Left of Target Center

Figure 2-10: **Motion Sensitivity of Center Square**

$$\begin{bmatrix} \delta e_1 \\ \vdots \\ \delta e_{N/2} \\ \delta e_{N/2+1} \\ \vdots \\ \delta e_N \end{bmatrix} = \begin{bmatrix} \frac{\partial x_1}{\partial X} & \frac{\partial x_1}{\partial Y} & \frac{\partial x_1}{\partial Z} & \frac{\partial x_1}{\partial \alpha} & \frac{\partial x_1}{\partial \beta} & \frac{\partial x_1}{\partial \gamma} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial x_{N/2}}{\partial X} & \frac{\partial x_{N/2}}{\partial Y} & \frac{\partial x_{N/2}}{\partial Z} & \frac{\partial x_{N/2}}{\partial \alpha} & \frac{\partial x_{N/2}}{\partial \beta} & \frac{\partial x_{N/2}}{\partial \gamma} \\ \frac{\partial y_1}{\partial X} & \frac{\partial y_1}{\partial Y} & \frac{\partial y_1}{\partial Z} & \frac{\partial y_1}{\partial \alpha} & \frac{\partial y_1}{\partial \beta} & \frac{\partial y_1}{\partial \gamma} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial y_{N/2}}{\partial X} & \frac{\partial y_{N/2}}{\partial Y} & \frac{\partial y_{N/2}}{\partial Z} & \frac{\partial y_{N/2}}{\partial \alpha} & \frac{\partial y_{N/2}}{\partial \beta} & \frac{\partial y_{N/2}}{\partial \gamma} \end{bmatrix}_{\vec{\mathbf{x}}_{\mathbf{co}}} \begin{bmatrix} \delta X \\ \delta Y \\ \delta Z \\ \delta \alpha \\ \delta \beta \\ \delta \gamma \end{bmatrix} \tag{2.17}$$

Six DOF Motion Sensitivity

The derivation of the sensitivity matrix illustrates how the raised center square of STAR's navigation target allows for the needed sensitivity to six degree of freedom camera motion. Since the center square is closer to the COP when the camera is in its nominal state, its edges will move by a different amount than those of the other squares for a particular motion. If the center square was not raised, there would be no information available to distinguish a Y_c translation and a pitch by an angle β (similarly X_c translation and yaw rotation would be indistinguishable). This is illustrated in Figure 2-10 showing the large motion of the center square in relationship to the outer squares for combined lateral translation and yaw of the camera.

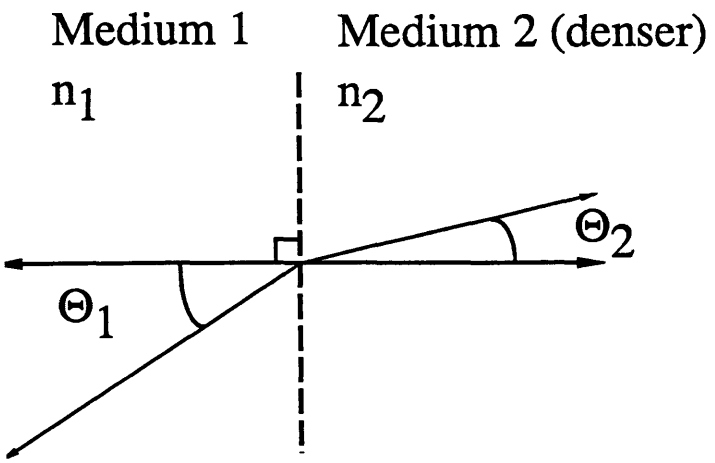


Figure 2-11: **Refraction**

Refraction Effects Underwater

Scuba divers are familiar with the effect that objects appear to be larger and closer underwater when viewed through a diving mask. This is caused by the bending of light, called *refraction*, as it passes between the denser water medium and air. Obviously, for an accurate underwater vision-based position sensing system, refraction must be taken into account. With a few simple assumptions however, refraction can easily be incorporated in the vision-based navigator.

The amount that light bends as it travels from one medium to the other can be calculated by *Snell's Law* in terms of the *index of refraction* (n) of the two media [11]; This is illustrated in Figure 2-11 showing that as light passes at an oblique angle through the boundary of two different media it will bend toward the denser medium:

$$n_1 \sin \Theta_1 = n_2 \sin \Theta_2 \tag{2.18}$$

To model the underwater video camera, a few assumptions are made to simplify the problem. The boundary between air and water is modelled at the camera's COP (ignoring the small gap of air between the lens and the waterproof camera housing and also the glass of the housing). It is also assumed that all the angles involved

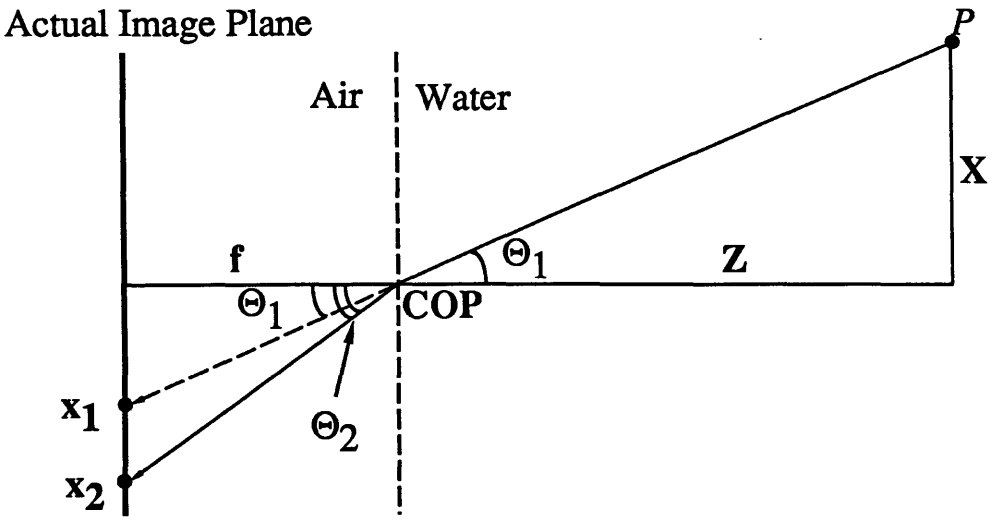


Figure 2-12: **Refraction Effect on Underwater Image**

are small, which is reasonable given the target dimensions and the nominal ranges of interest. Figure 2-12 illustrates the altered location due to refraction of the image of a point when viewed underwater. The image of a fixed point will move from location x_1, when viewed in the air, to x_2 because of refraction. Refraction causes a magnification of the image underwater.

To deal with the effects of magnification underwater, a new effective focal length is derived below from Figure 2-12 given that $n_{water} = 1.33$ and $n_{air} = 1$. This scaled focal length simply replaces the original focal length in the calculation of the sensitivity matrix for an underwater vision sensing system. From the indices of refraction, the relationship between x_1 and x_2 can be computed:

$$\begin{aligned} 1.33 &= \frac{\sin\Theta_2}{\sin\Theta_1} \\ &\approx \frac{\tan\Theta_2}{\tan\Theta_1} \\ &= \frac{x_2/f}{x_1/f} \end{aligned}$$

From the ratio of x_1 and x_2, a new effective focal length can be found:

$$\begin{aligned} x_2 &= 1.33x_1 \\ &= (1.33f)\frac{X}{Z} \\ f' &= 4/3f \end{aligned} \tag{2.19}$$

2.3.3 Positional State Computation From Edge Locations

Equation 2.17 is the basis for computing the camera state deviations from the measured changes of navigation target's edge locations. However, H is not square since with even only one sample along each square's edge ($N = 12$) the sensitivity matrix has dimensions of 12 x 6 and is therefore not invertible. To reduce the necessary computation time of the real-time system, a simple linear least-squares inversion (for which the accuracy increases with more samples) is performed through the pseudoinverse of H [15]:

$$\begin{aligned} H^{\dagger} &= \left(H^T H\right)^{-1} H^T \\ \delta\vec{\mathbf{x}}_{\mathbf{c}} &= H^{\dagger}\delta\vec{\mathbf{e}} \end{aligned} \tag{2.20}$$

The pseudoinverse of H can be computed if the product $(H^T H)$ is invertible. This condition is met if the columns of the sensitivity matrix are independent, which is the mathematical requirement that navigation features must be sensitive to six degree of freedom motion. Because of the raised center square, the sensitivity matrix of STAR's navigation target is invertible. With ($N > 60$) edge samples, the resulting $\delta\vec{\mathbf{x}}_{\mathbf{c}}$ was found to be quite accurate despite noisy edge location measurements.

$H^{\dagger}$ is also constant for a given nominal state and can therefore be computed beforehand outside of the control loop again saving computation time. The position and orientation sensing of the camera has therefore been reduced to a linear multiplication of a constant matrix and the measured changes in edge locations.

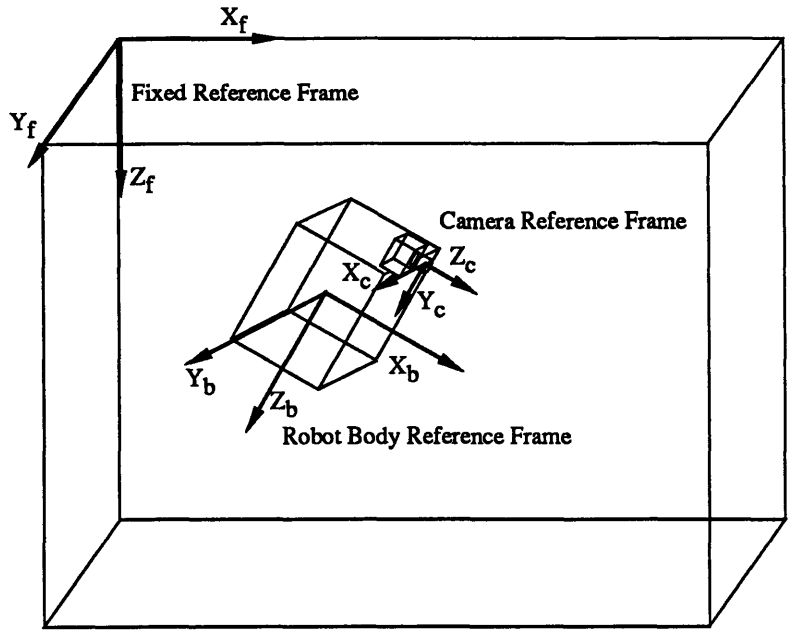


Figure 2-13: **Reference Frames**

2.4 Robot Positional State Computation

The final step is to transform the measured camera states to the states of the robot which are used as inputs to the vehicle control system. This is done through a reference frame transformation.

2.4.1 Fixed, Body, and Camera Coordinate Frames

Figure 2-13 shows the three main reference frames associated with the vision navigation problem (in this case, an underwater robot in a pool) [8]. These frames are used to determine the relative positions and orientations of the camera, the robot body, and the fixed environment. The camera, body, and fixed frames are all right-handed with axes (X, Y, Z).

The fixed (environment) reference frame can have any orientation. However, there may be an orientation which is "natural" to a particular physical situation. For the case of a free-flying underwater robot, the fixed frame's origin is naturally located at the water's surface with the Z_f axis pointing down (which allows direct correlation

to depth measurements).

The robot body reference frame can also be defined according to the most convenient orientation; in general, the origin of the body frame is located at the robot's center of mass. The body frame assigned to STAR follows the same conventions as those of airplanes with the X_b axis pointing forward out of the upright vehicle, the Y_b axis pointing right, and the Z_b axis pointing down. Its origin is located at the vehicle's center of mass. The body reference frame Euler angles are assigned with the usual convention with roll about X_b , pitch about Y_b , and yaw about Z_b.

As discussed in Section 2.2.2, the camera's reference frame is defined according to common machine vision conventions.

2.4.2 Camera to Body Frame Transformation

For the most general application, a camera will not be rigidly mounted on a free-flying robot. For example, a pan and tilt unit would allow the camera to view a navigation feature in one direction while the robot flies in another. However for the scope of this thesis, it is assumed that the camera is rigidly mounted to the body of the robot which results in a constant transformation from the camera to body reference frames. In the more general case, the transformation would need to be updated as the camera's orientation relative to the vehicle changes.

For STAR, the vision navigation camera is usually mounted on the top, right corner on the vehicle at coordinates $(L_x, L_y, -L_z)$ relative to the origin of the body frame. The changes in camera state, measured by the vision system, must therefore be transformed to the corresponding vehicle state changes. Pure translation along an axis of the camera frame is simply transformed as the same translation along the corresponding body frame axis. However, as Figures 2-14, 2-15, and 2-16 indicate, transforming the angles is a bit more complicated. Rotation of the camera frame is equivalent to the same rotation of the body frame with an added translational displacement which must also be calculated.

Since all state deviations are assumed to be small, the linear transformation matrix shown below can be used to convert camera states to robot body states for translation

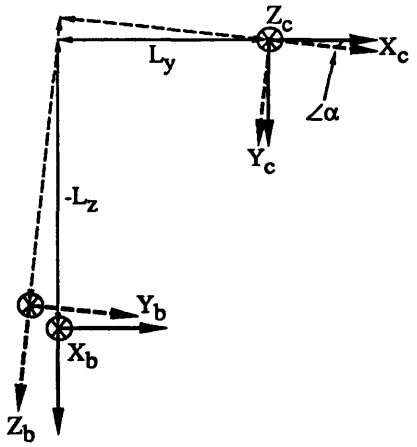


Figure 2-14: **Roll Transformation from Camera to Body Frame**

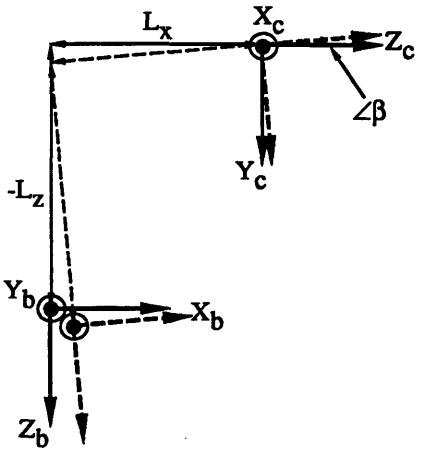


Figure 2-15: **Pitch Transformation from Camera to Body Frame**

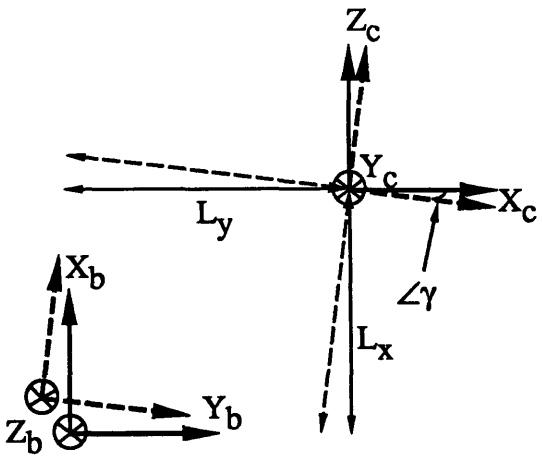


Figure 2-16: **Yaw Transformation from Camera to Body Frame**

and orientation:

$$\delta\vec{\mathbf{x}}_\mathbf{b} = T\delta\vec{\mathbf{x}}_\mathbf{c} \tag{2.21}$$

$$T = \begin{bmatrix} 0 & 0 & 1 & 0 & L_z & L_y \\ 1 & 0 & 0 & -L_z & 0 & -L_x \\ 0 & 1 & 0 & -L_y & L_x & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.22}$$

This constant transformation matrix can also be pre-computed and combined with the pseudoinverse of the sensitivity matrix so that one constant pre-computed matrix multiplication is sufficient to compute the robot positional state from measured changes in the target's edge locations during each control loop:

$$\delta\vec{\mathbf{x}}_\mathbf{b} = \left(TH^\dagger\right)\delta\vec{\mathbf{e}} \tag{2.23}$$

Chapter 3

STAR Vision-Based Navigation and Control

Without six degree of freedom control, free-flying space and underwater robots cannot effectively function in their environments. Almost all robotic tasks, such as maneuvering from place to place, station-keeping, and docking, depend on accurate vehicle position and orientation control. Six degree of freedom position and attitude sensing is therefore essential to the development of free-flying robot controllers. As discussed in Chapter 1, vision-based sensing, in space and underwater, has several advantages over conventional systems such as radar, laser range finders, and sonar.

The difficulty with vision-based position and attitude sensing is to develop a system which is accurate and applicable to a variety of tasks, but is also fast enough to be incorporated in a real-time robot control system. The goal of the vision algorithm, developed in Chapter 2, is to provide sensing for an actual six degree of freedom robot control system with relatively basic PC hardware. The test-bed for the vision-based navigation and control system is the Submersible for Telerobotic and Astronautical Research (STAR), a neutrally buoyant space robot simulator.

The specifics of STAR's vision-based navigation and control system, including hardware, control system design, and software implementation, are described in this chapter.

3.1 STAR Subsystems

Underwater simulation is an effective reproduction of the zero-g environment of space here on Earth. By being neutrally buoyant in both depth and orientation, space robot simulators, such as STAR, have six degrees of freedom of motion available like a free-flying space robot. The major shortcoming of underwater simulation is the mass and drag of water which is totally unlike the vacuum of space. Nevertheless, much can be learned from underwater simulation. STAR is a platform for development of a variety of space robot functions such as teleoperated control, use of a manipulator arm, and autonomous control.

Most of the design, construction, and integration of STAR's major subsystems was completed in the summer of 1991 under project manager Kurt Eberly, then a graduate student in LSTAR [9]. There were several major factors which influenced the development of STAR. The main requirements of the vehicle were to carry battery power for eight hour pool test sessions, to house the powerful electronic systems which are required for such advanced tasks as vision-based vehicle control and manipulator control, and to be precisely neutrally buoyant in depth and orientation. Low maintenance between pool tests was also a primary concern; all of the vehicle's subsystems were therefore designed to be modular and easily serviceable. Finally, minimizing development cost was also a design factor.

3.1.1 Structure

Unlike earlier space robot simulators, STAR has no large pressurized compartments. With these earlier vehicles, air from a scuba bottle would be used to regulate the pressure in the large electronics compartments. As the air supply would be decreased during the course of a pool test session, the buoyancy and balance characteristics of the vehicle would be dramatically changed. Because of the danger of expelled hydrogen from discharged batteries, the battery compartments of these earlier robots could not also be pressurized. However, the battery housings were not very stiff so their volume would change as the vehicles ascended and descended. This caused

further unwanted changes of their depth buoyancy and rotational balance.

To avoid this problem, the main body of STAR is an entirely waterproof, unpressurized housing made of very stiff, 1/2 inch thick aluminum. The 3 ft x 2 ft x 2.25 ft main body contains two battery drawers and a spacious electronics drawer which can all be conveniently slid out for easy access. The drawer doors are sealed with O-rings.

Each battery drawer contains six 12V lead-acid gel cell batteries which provide the power for the vehicle's electronic, propulsion, and other various subsystems (such as cameras, pan and tilt camera unit, and manipulator arm). The on-board computer system, power converters, and angular rate sensors are contained in the electronics drawer. The main vehicle body was sized to carry these drawers and still be neutrally buoyant.

Aluminum cages are attached to all sides but the front of STAR. These cages protect the vehicle's body and provide attachment points for the thruster motors. With the side cages attached, the dimensions of the vehicle are roughly 4 ft x 3.75 ft x 3 ft. To simplify balancing the vehicle, weights are also attached to the aluminum cages which can easily be repositioned along the vehicle's axes. Camera housings, a thruster motor pressurization system, and a manipulator (which is currently being built and tested) are also attached to the outer surfaces of the vehicle. Besides the main aluminum body, all of STAR's structural parts can be easily disassembled and reconfigured. Figure 3-1 provides a view of the back side of STAR showing the three main drawers and side cages. The vehicle's thruster motors, which are enclosed in white ducts and plastic grates, are also visible.

3.1.2 Electronics Drawer

The electronics drawer houses the three main components of the vehicle's on-board computer system, a computer rack, STD I/O bus, and patchboard, as well as the vehicle's inertial sensors and power converters. The computers, I/O bus, and patchboard are conveniently housed on a 19 inch horizontal rack near the top of the electronics drawer. The inside of the electronics drawer is shown in Figure 3-2.

STAR is equipped to carry up to three Ampro Little Board computers. The

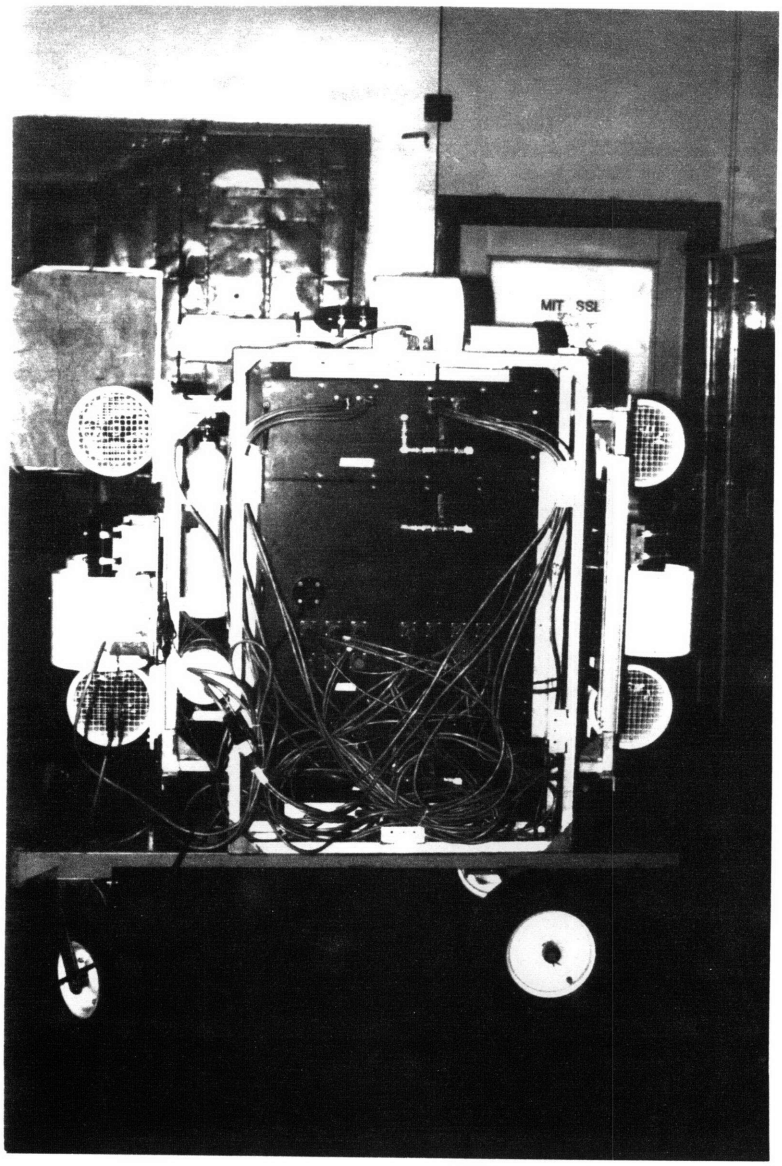


Figure 3-1: **Rear View of STAR**

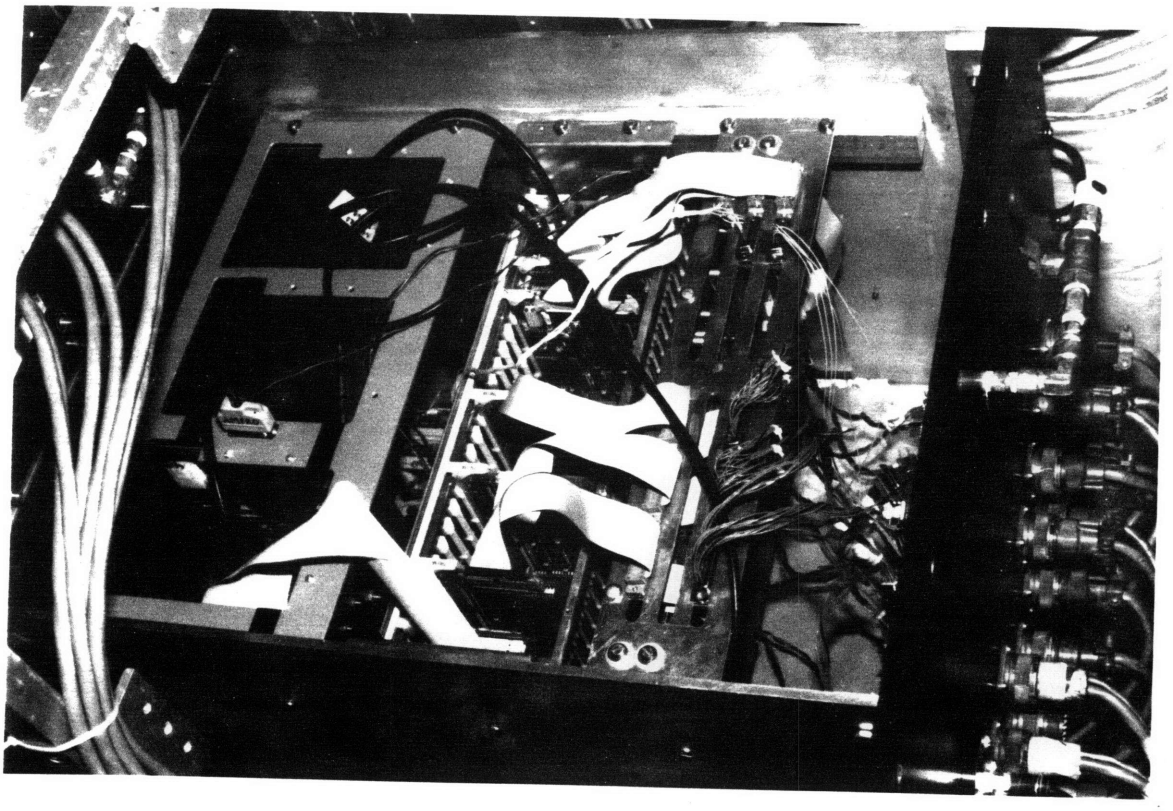


Figure 3-2: **STAR Electronics Drawer**

compact size, reliability, and low cost of the PC compatible Little Board computer makes it ideal for this type of application. For the vision-based control system, only one 20 MHZ Little Board/386 (with an Intel 80386-based processor and 80387 math co-processor) is used, primarily to implement the digital compensator and activate the vehicle's thrusters.

A major component of STAR's computer system is the QNX 4.01 operating system from Quantum Software Systems Ltd. of Kanata, Ontario, Canada. QNX 4.01 is a POSIX compatible "UNIX-like" operating system for PC's. QNX allows convenient networking and communication between the robot's on-board and control station computers; for example, software can be changed and down-loaded to the robot from the control station allowing on-site software editing during pool tests. The control station computer is linked to the vehicle with a 93Ω coaxial umbilical cable which provides a reliable link as the vehicle maneuvers underwater. QNX is designed to be a fast, real-time, multi-processing, multi-tasking operating system capable of more than 7,200 task switches per second. This feature makes a real-time vision-based robot control and navigation system feasible with relatively basic PC computer hardware.

The various I/O cards which connect the on-board computers to the vehicle's subsystems are housed in a STD bus rack. The STD bus was chosen because of its compact size and relatively low cost. A specially designed interface is used between the PC computers and the STD bus which allows multi-computer access to the I/O bus. The primary I/O cards currently being used are a power switching card for turning the main battery relays on and off, A/D converters for monitoring the on-board inertial sensors, and motor control boards connected to the vehicle's thrusters and manipulator arm motors.

A patchboard is included as an organized connection point for the many electronic signals going from the STD bus to various points in the electronics drawer and outside the vehicle. The patchboard makes all the electronic subsystems modular and easily serviceable.

Most free-flying vehicles in space and underwater have inertial sensors, such as gyroscopes, for measuring angular rate. STAR also has three piezoelectric angular rate

sensors, manufactured by Watson Industries, mounted along the three vehicle axes on the base of the electronics drawer. The angular rate measurements are used for an orientation controller for teleoperated flight (as described below in Section 3.1.4) and in conjunction with the vision-based orientation control system. Unfortunately, the sensors suffer from a slight bias drift which must be updated when running orientation controllers for more than half an hour.

The electronics drawer also contains the DC-DC power converters which provide power to the vehicle's electronics. These converters are connected to four of the vehicle's twelve batteries; the other eight batteries power the vehicle thrusters, manipulator motors, and camera platform. Throughout the vehicle's subsystems, a careful attempt has been made to isolate the sensitive computer signals from the noisy analog environment of the motors. Single point grounding rules have also been strictly followed.

3.1.3 Propulsion

STAR is propelled by eight bi-directional thrusters which provide three degrees of translational and three degrees of rotational motion. The four thrusters along the X_b axis provide thrust for yaw and pitch rotation as well as translation along the X_b direction. There are two thrusters directed in both the Y_b and Z_b axis directions for Y and Z translation and roll.

The thrusters are actually electric DC fishing trolling motors (the 35W model manufactured by Minn Kota is used on STAR) with plastic propellors. Since the seals on these motors are not designed for STAR's operating depths of up to 40 ft, the motor housings are pressurized to 10 psi over ambient with a modified scuba regulator. The motor assemblies are enclosed with fiberglass ducts and plastic grates to protect divers during pool tests.

A National Semiconductor LM629 motion-controller chip controls each of the thruster motors according to commands from the on-board computer. The LM629 is interfaced to the motor with an H-bridge Darlington transistor driver circuit for motor actuation. The sensitive computer signals are separated from the noisy analog

signals with optoisolators.

Until recently, the thruster motors' rotation speed was driven open-loop with a pulse-width-modulated (PWM) signal. However, the performance of the robot's position and orientation control system suffered from sluggish motor response, stiction, and limit cycling (see Section 4.3.2 for experimental results). It was therefore decided to add closed-loop velocity control for the thrusters. This was accomplished by mounting an optical encoder to each motor to provide feedback of the motor shaft angular velocity. The LM629 motor controller easily interfaces with an optical encoder and provides closed-loop motor velocity control. Appendix B gives a complete description of the mounting of the Hewlett Packard HEDS-9140-A00 optical encoder module and HEDS-5140-A13 encoder wheel in the thrusters, the digital compensator implemented with the LM629 motor controllers, and the maximum motor shaft velocity. With tight, closed-loop thruster velocity control, it is possible to implement a responsive and accurate vehicle position and orientation control system.

3.1.4 Control Station

The control station at the surface of the pool serves as the interface between STAR and a human operator. A pilot uses two 3-DOF joysticks, one for translational and one for rotational motion, to manually fly the vehicle. Television monitors and a helmet mounted display attached to a two-DOF head-tracker are used to provide video feedback from STAR's on-board cameras. A Gateway 4000 computer, with a 20 MHz 80386-based processor, monitors commands from the pilot's input devices and communicates with the vehicle's on-board computers. Since STAR is not equipped with mass storage devices, the control station computer is also used to down-load vehicle software and record data via the QNX operating system.

The vehicle's rate sensors have been incorporated into a closed-loop control system which allows the pilot to command the rotation rate with the rotation joystick when manually flying the robot (translation is controlled with open loop thrust commands). This is a great advantage when flying the vehicle since it is much more difficult to precisely point the vehicle with only open-loop torque commands. With the closed-

loop orientation control, a pilot has the necessary precision to engage the vision-based controller as described in Section 4.1.

3.2 Machine Vision Hardware

Currently, the processing for the vision-based sensing system is performed with the control station computer; the on-board camera's video signal is fed directly to the control station and digitized with a frame grabber[1]. The frame grabber being used is the Overlay Frame Grabber (OFG) from Imaging Technology, one of many standard, inexpensive image processing boards available today.

The analog video signal from the vehicle's on-board black and white camera is digitized with the frame grabber. A frame of the video image (composed of interlaced odd and even scans) is divided into 480 rows and 640 columns giving 307,200 individual pixel elements. An analog-to-digital (A/D) conversion of the incoming video signal assigns a discrete *brightness* value to each pixel. An eight-bit frame grabber, such as the OFG, provides a range of brightness values from 0 to 255 from black to white. An image's brightness values can be accessed and analyzed through the two dimensional array of pixels.

The camera used for STAR's vision-based control system is a Pulnix TM-7CN which combines high resolution and compact size. The camera's CCD imager is also divided into a two dimensional, 494(V) by 768(H), array of pixels. Each pixel of the TM-7CN is very small with dimensions of 8.4 μm x 9.8 μm. Since the frame grabber being used only digitizes 480 scan lines, 14 lines of the camera image are lost. Some resolution is also lost since the camera's 768 horizontal image pixels are divided into only 640 pixels with the frame grabber.

The development of the measurement equations in Chapter 2 relies on measurements of image feature locations relative to the camera reference frame on the image plane. It is therefore necessary to determine the location of a camera's principal

[1]This is the merely the most convenient setup for the current hardware; for an autonomous vision-based controller, vision processing would have to be moved to a dedicated computer with frame grabber on the robot.

point with a particular frame grabber to determine the origin of the camera frame. Azarbayejani determined the location of the TM-7CN's principal point at pixel location [303,278] on the OFG.

The scale factors to convert from OFG pixels, measured by the computer, to actual distances along the TM-7CN's image plane, are derived below:

$$S_{X_c} = \frac{(768)\ (8.4\mu\text{m})}{640} = 1.008\text{x}10^{-5}\ \text{m/pixel} \tag{3.1}$$

$$S_{Y_c} = 0.98\text{x}10^{-5}\ \text{m/pixel} \tag{3.2}$$

The frame grabber used has a feature which can modify the image displayed on a monitor by overlaying colors on the original video signal. This capability is used with the vision-based navigator to provide reference marks for aiding the pilot in manually engaging the target as described in Section 4.1.

3.3 Control System Design

The goal for the control system design is to find a simple and fast real-time implementation that provides good performance and robustness to disturbances. The main requirement of the vision-based control system is to accurately position and orient the vehicle relative to the target for a variety of related applications. The control system is used for automatic station-keeping, closed-loop maneuvers about the nominal state, and positioning the vehicle during manipulator arm operation. Fast time response with minimal overshoot is therefore essential for all six degrees of freedom. A tight control system that can withstand disturbance forces and torques on the vehicle is also essential especially when using the vehicle as a manipulator platform.

Controlling six positional states of the robot with eight thrusters is a complex multi-input, multi-output (MIMO) control problem. It is difficult to derive a complete model of the vehicle's dynamics in the water necessary for complex model-based

control algorithms. Fortunately, for the applications of interest, the vehicle's motions are small and slow so that the complete nonlinear dynamics of the vehicle should not effect the control system performance. A further useful simplification is to treat each degree of freedom as decoupled from the others. From experiences of manually flying STAR, this is a realistic first order approximation (for example an X translation command does not result in large yaw or pitch rotations). By decoupling the plant, one single-input, single-output (SISO) compensator can be implemented for each degree of freedom. Since each thruster is used for both translation and rotation, the commands for each degree of freedom activated by a particular thruster are commanded as a weighted sum; this ensures that a single degree of freedom's command does not saturate the thruster.

Section 3.3.1 describes an attempt to model each degree of freedom; unfortunately, the procedure used had limited success. However, from the general structure of the plant and previous experiences with orientation rate control (see Section 3.1.4), it was decided to first try simple proportional/derivative (PD) control for each degree of freedom. Proportional/derivative control has several advantages over more complex control algorithms: it requires minimal computation, it is possible to intuitively tune the control gains for desired system performance, and it does not rely on a precise plant model. Pure proportional/derivative compensators did work well for the attitude degrees of freedom. For the translational degrees of freedom, proportional/derivative control had to be developed into lead compensation since no rate measurement is available. Sections 3.3.2 and 3.3.3 describe the digital compensators used for orientation and translation control. Step responses of the control system for each degree of freedom are presented in Chapter 4.

A primary concern during the development of STAR's vision-based control system was the sampling period that could be achieved with the basic PC hardware. The edge detection algorithm, vision sensing computation, and control system implementation were all designed for a simple, fast implementation (without strict optimization of the source code). The final six degree of freedom vision-based control system has a sampling period of 0.1 sec. The system running at 10 Hz does an excellent job of

stabilizing and controlling STAR as shown in Chapter 4.

3.3.1 Plant Model

To aid the control system design, an attempt was made to experimentally model the dynamics of STAR by measuring the vehicle's responses to commands for each degree of freedom. With closed-loop control on the robot's thrusters, it was possible to accurately measure the applied thrust. A yard-stick and underwater video camera were used to measure the robot's responses.

Although a useful system model with accurate plant parameters was not obtained, an important point was learned which aided the design of the final vision-based control system. This information will be useful for future modelling of the vehicle (this will be necessary for example for implementing an observer to estimate the vehicle states). It was first thought that each of STAR's degrees of freedom could be modelled as a double integrator (like a free-flying space vehicle) for the typical small and slow motions of the vehicle under control. However, measurements from the video showed that water damping plays a significant role in the vehicle's response. With half thrust commands, the vehicle reached a terminal velocity within about 0.2 sec instead of continuously accelerating over a longer period of time. The effects of water drag on the vehicle's dynamics must therefore be included in a realistic model of STAR. For the typically slow speeds of the vehicle, the water drag can best be modelled as increasing linearly with velocity.

3.3.2 Proportional/Derivative Orientation Control

Designing a controller for STAR's angular degrees of freedom is greatly simplified since STAR is equipped with angular rate sensors. Combining the angular rate measurements from the inertial sensors with the vision-based angle measurements leads to a very straightforward digital proportional/derivative (PD) compensator design [10]. The control command (u) is computed for the current time step (k) from a proporitional gain constant (K_p) multiplied by the current angle (θ) and a derivative

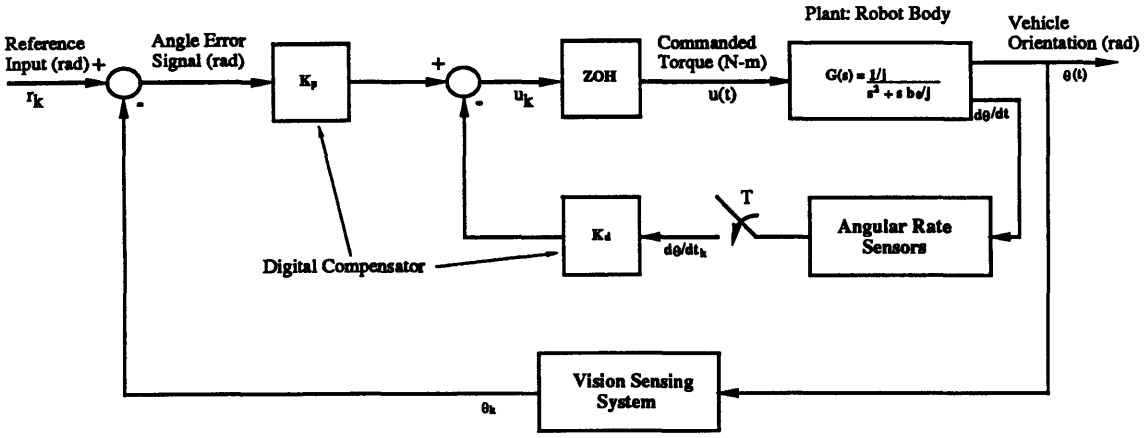


Figure 3-3: Attitude Control System

gain constant (K_d) multiplied by the current angular velocity:

$$u_k = K_p\theta_k + K_d\dot{\theta}_k \tag{3.3}$$

Since the compensator has no dynamics (the angular velocity is measured directly from the inertial sensors), the control equation implemented in software is very simple and computationally fast.

Figure 3-3 shows a block diagram of the digital control system wrapped around the continuous time plant for the vehicle's rotational degrees of freedom.

3.3.3 Lead Translation Control

Control of the robot's translational degrees of freedom is also based on simple proportional/derivative control. However, unlike with the rotational degrees of freedom, there are no direct velocity measurements available for the translational degrees of freedom. An estimate of the velocity, from the position measurements, must therefore be computed by the digital compensator in order to provide damping. The major drawback of differentiating the position measurements is the amplification of measurement noise; this will cause the thrusters to "twitch" in response to noisy measurements. This violent motor reaction was noticed with the preliminary control

system and actually broke some of the plastic propellors. In order to calm the motor response, a roll-off on the derivative control can be added (which is equivalent to lead compensation).

The derivation of the digital lead compensator used for translational control is given below [10]. The discrete compensator is derived from a continuous time proportional/derivative controller with an added pole to roll off the derivative gain (a lead compensator):

$$G(s) = \frac{u(s)}{x(s)} = K_p + K_d \frac{s}{s+a} \tag{3.4}$$

A backwards difference approximation of the derivative is used to derive the discrete time version of the compensator:

$$\begin{aligned} \dot{x}(t) &\approx \frac{x_k - x_{k-1}}{T} \\ s &\rightarrow \frac{1-z^{-1}}{T} = \frac{z-1}{Tz} \\ D(z) &= K_p + K_d \frac{z-1}{z-1+aTz} \end{aligned} \tag{3.5}$$

(3.6)

The denominator of the derivative control term can be expressed in terms of a single constant, $a' = 1/aT$. As a' is increased from 0 to 1 in the z-plane, the derivative control is rolled off faster:

$$\begin{aligned} D(z) &= K_p + K_d \frac{z-1}{z-a'} \\ &= \frac{(K_p + K_d)z - (K_p a' + K_d)}{z-a'} \end{aligned} \tag{3.7}$$

Finally, the control difference equation implemented by the on-board computer is derived from Equation 3.7:

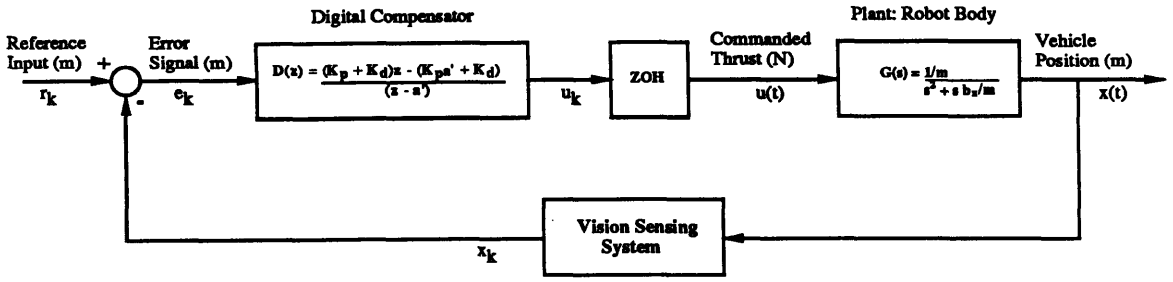


Figure 3-4: **Translation Control System**

$$u_k = (K_p + K_d)\, x_k - (K_p a' + K_d)\, x_{k-1} + u_{k-1} \tag{3.8}$$

Figure 3-4 shows a block diagram of the digital lead compensator used to control each of the robot's translational degrees of freedom.

3.4 Software Implementation

The software for STAR's navigation and control system is written in "C" for the control station and on-board computers. Vision processing, reading pilot commands, and recording data are done with the control station computer. Monitoring the angular rate sensors, computing the control commands, and activating the robot's thrusters is performed by the on-board computer.

The vision sensing system currently uses a very simple method for finding the changes in target edge positions. The search algorithm is simplified to be a fast implementation. The edge detection algorithm performs a very fast binary search for the current position of the edge between the black and white region; these fixed binary scans are done from the nominal edge position determined by the vision system's nominal state. The scans are performed only on the central 5 cm of each edge to ensure that scans will find an edge for a particular maneuver. Figure 3-5 illustrates the fixed scan method for finding the new edge positions; the dotted lines across the edges represent the fixed scan locations and the dots represent the newly found edge

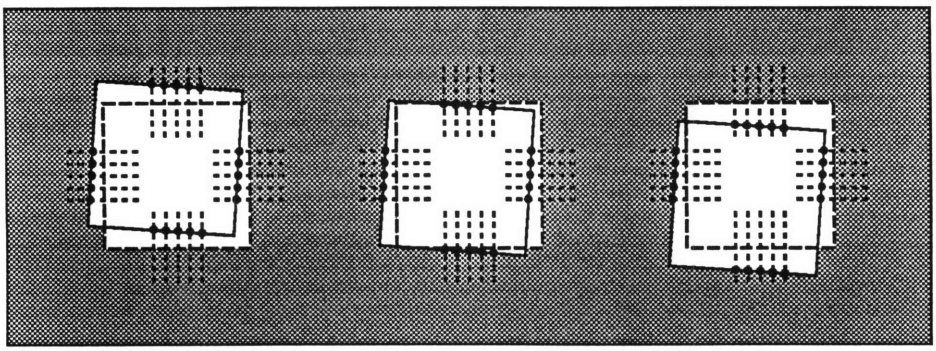


Figure 3-5: **Fixed Scans for Edge Detection**

locations. In general, seven samples are taken along each edge to reduce the measurement noise when computing the positional state through the sensitivity matrix. In case an edge cannot be found along a scan line due to a large vehicle motion, the vision-based control system is turned off and the vehicles thrusters are reset.

This simple search algorithm works well for station-keeping and for very small commanded changes from the nominal state. However, the fixed search method is the major drawback of the current system since the target cannot be tracked during large maneuvers or for large disturbance inputs. A more sophisticated search algorithm is currently being developed by LSTAR and is discussed in Chapter 5.

At the beginning of the control loop, the deviations of the edges from the nominal positions are determined from the current image. The current vehicle position and orientation is computed from the edge deviations through the precomputed state transformation matrix, T, and precomputed pseudoinverse of the sensitivity matrix, $H^{\dagger}$ (see Section 2.3.3). The current vehicle state measurements and pilot commands are passed to STAR's on-board computer. Onboard the vehicle, the corresponding thruster commands are computed from the current measured states and angular rate measurements. The thrust commands are activated through the separate closed-loop motor velocity controllers for each thruster. A schematic diagram of the software is shown in Figure 3-6.

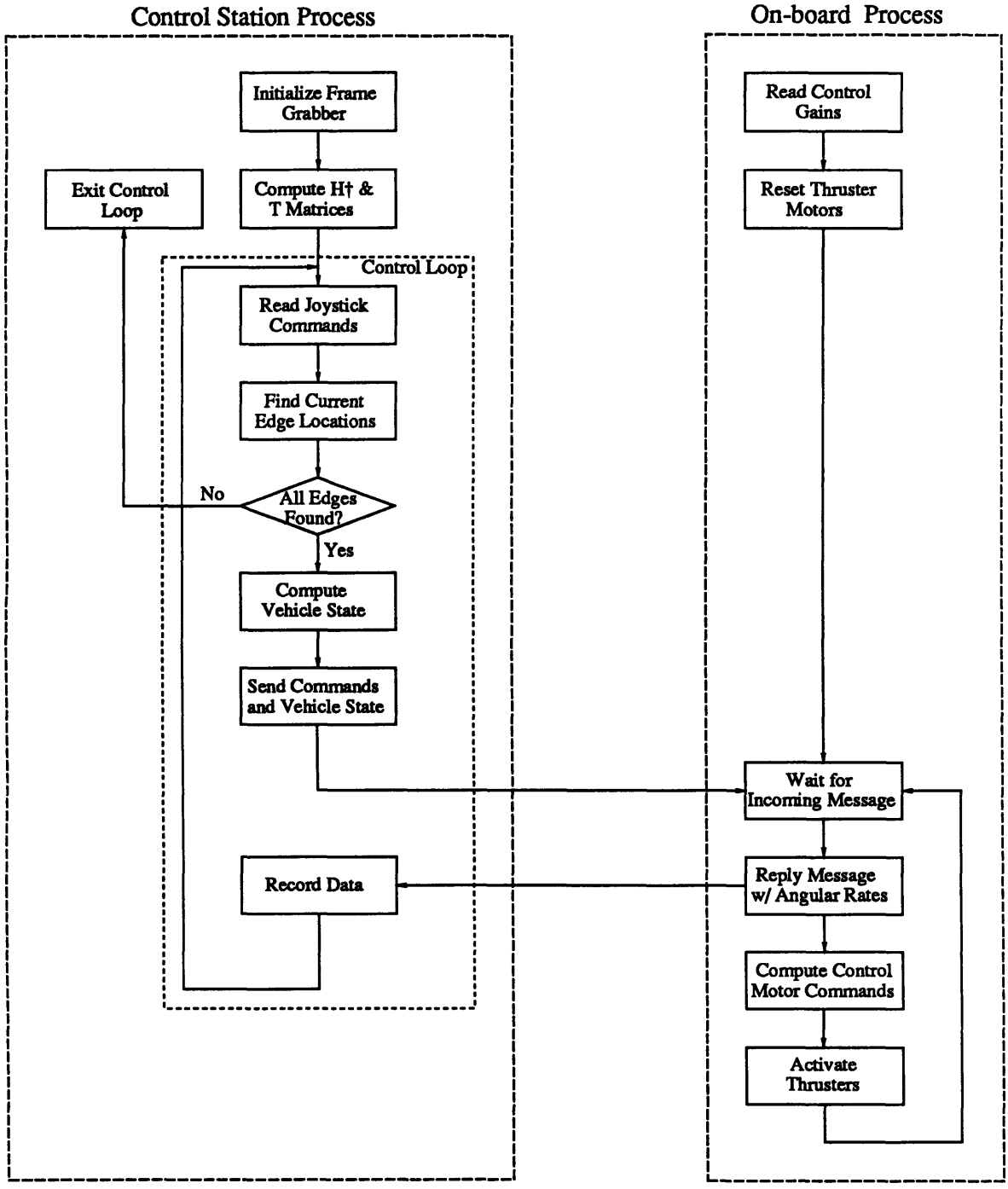


Figure 3-6: **Fixed Scans for Edge Detection**

Chapter 4

Experimental Results

Performance of the real-time vision-based control system during underwater tests of STAR is discussed in this chapter. The experimental setup and responses of the system to step inputs to each of the vehicle's six degrees of freedom are described. Interesting observations of the vision-based navigation and control system's performance and capabilities are also presented.

4.1 Testing Procedure

STAR's current vision-based navigation and control system was developed in stages during five pool test sessions between late 1991 and spring of 1992. The three major components of the system, which were developed simultaneously, are the vision-based position and orientation sensing algorithm, inner-loop control of the thrusters, and the robot position and attitude control system.

To determine if it is possible to implement a real-time vision-based control and navigation system for STAR with relatively inexpensive and basic hardware (which is described in detail in Chapter 3), the problem was first simplified as much as possible to find the fastest possible implementation. The results of each stage of the system's development were used to enhance the system to its current capabilities.

The initial development of the control system was as a regulator to keep the vehicle in a fixed position and orientation relative to the navigation target. The first success-

ful orientation control was run with a fast and very simple vision sensing algorithm which directly detects camera orientation changes from target edge location changes (this primitive algorithm is discussed in detail in Appendix A). This preliminary system was able to actively point the vehicle at the target but not control translation. Orientation control was implemented first because the accurate angular velocity measurements from the on-board rate sensors made tight proportional/derivative orientation control possible.

The next major step was full six degree of freedom station-keeping control, with proportional/derivative attitude and proportional translational control, again using the same simplified vision sensing algorithm. Damping was not yet added to the translational states since the vision position measurements were quite noisy. The poor system response from this series of tests demonstrated the need for closed-loop thrust control as discussed below in Section 4.3.2.

During the following pool test, with closed-loop angular velocity control on all eight thrusters, the full six degree of freedom vision-based control system using the linearized sensitivity matrix was implemented. The tight vehicle control of the system and the low noise for the translational state measurements made it possible to add damping for translational motion (the control system discussed in Section 3.3). Roll-off was added to the derivative control to calm the response of the motors to save the plastic propellors. As expected, damping dramatically improved the time response for the translational states.

With this control system, the capability for adding a reference input to the system was added. Step responses for the six degrees of freedom are presented below in Section 4.2. The vision-based control system was able to robustly fix the robot's position and orientation in the station-keeping mode. The system can also be used for small but precise closed-loop position and orientation changes from the nominal state. The reference inputs can be used for having the vehicle fly along a pre-computed trajectory (automatic docking for example) or from the pilot's commands for closed-loop teleoperated vehicle control.

The system offers the flexibility to be used in a variety of different situations;

various successful tests were run with two different camera lenses (a change was made from a 9 mm to 4.6 mm focal length range to provide a closer nominal range to the target), with different configurations of the camera on the vehicle, and with various nominal states. This flexibility is useful for obtaining the most convenient work space for a given robot task.

To engage the vision-based control system, the vehicle is first flown manually close to the nominal state in order to position the target image within the fixed search area used to scan for the location of the target's edges. To simplify the alignment process, an overlay of the limits of the fixed scans is added to the video display with the frame grabber as shown in Figure 4-1. Once the vehicle's position and attitude are close to the nominal state, the vision control system is engaged and controls the robot to the current nominal state and the reference inputs.

The power and flexibility of the QNX operating system made it possible to fine tune the control gains in real-time during the pool test sessions which was extremely useful. With the simple proportional/derivative control system it was possible to intuitively adjust the proportional and derivative gains for stability, time response, and overshoot. Fine tuning of the control system was simplified by concentrating on each degree of freedom individually. The control system for each angle was first developed; then with attitude control, the control for each translational degree of freedom was added one at a time.

4.2 Control System Performance

To illustrate the performance of the vision-based robot position and orientation control system, step responses along each of the vehicle's degrees of freedom are presented below; all tests were conducted with a 0.015 rad roll offset to increase the accuracy of the vision state measurements as discussed below in Section 4.3.1.

Figure 4-2 shows the response of the vehicle to a step input along the X_b direction (the commanded input is indicated by the solid line in the figure). In general, given the limitations placed on the control gains by the durability of the plastic propellors

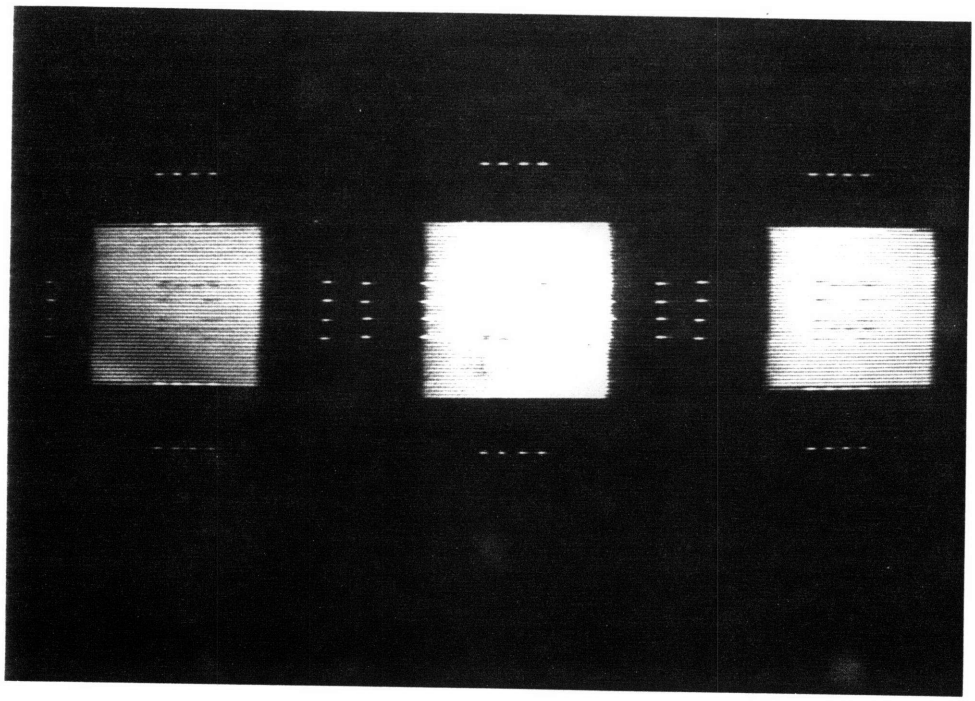


Figure 4-1: **Overlay Markings for Engaging Vision Target**

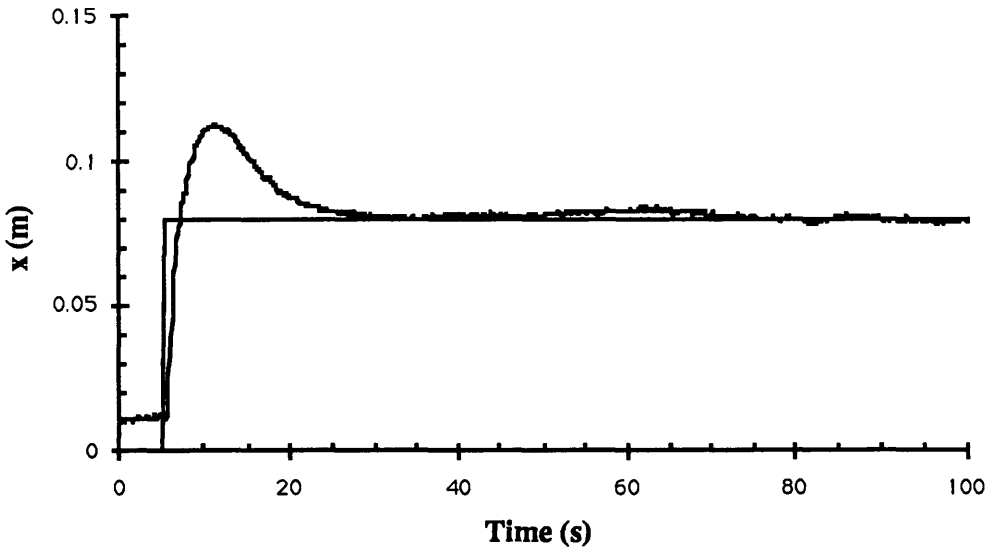


Figure 4-2: **X Step Response**

and thruster strength, the system's response is quite good. The vehicle settles close to the commanded position within 30 sec. The measurement noise along X_b is also extremely small. Figure 4-3 shows the thrust commands for one X thruster during the step response; once the state settles to the commanded input very little thrust is required to keep the vehicle in position.

Figures 4-4 through 4-8 show the other states during the X_b step input in Figure 4-2. Although all twelve edges of the target respond to motion along X_b, the control of the other states is decoupled from the X_b motion as desired. The positions along Y_b and Z_b remain close to the nominal zero positions throughout the X_b maneuver; without integral control to correct the steady-state error, the positions along these two axes do drift by a few centimeters. Since the position measurements along the Y_b and Z_b directions each depend on the location of only six of the target edges, instead of the changing size of the target's three squares as with range measurements along the X_b axis, the measurement noise for these two degrees of freedom is larger. Attitude control is also unaffected by the X_b step; the pitch and yaw angles do have small

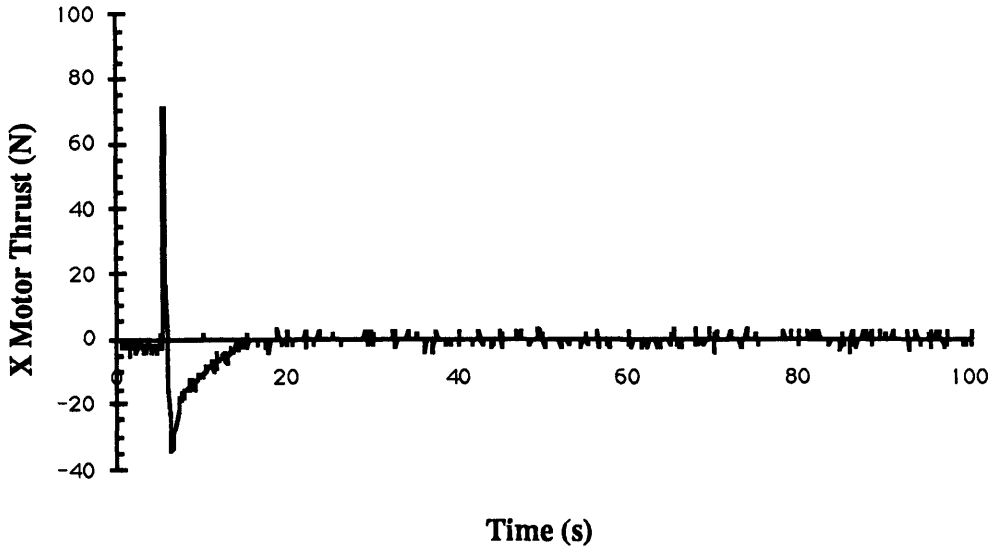


Figure 4-3: **Thrust Commands for X Step Input**

initial reactions when the X_b step is commanded, but they are quickly cancelled out by the control system as indicated in Figures 4-7 and 4-8. The largest peak-to-peak noise for the three angle measurements is only about 0.005 rad for roll.

The step responses for the Y_b and Z_b directions are shown in Figures 4-9 and 4-10. The positions settle close to the commanded values within 20 sec with a smaller overshoot than for the X step.

The advantage of having direct angular velocity feed-back is shown by the roll, pitch, and yaw step responses (Figures 4-11 through 4-13). With accurate velocity measurements, the orientation proportional and derivative controller has a very fast time response for all three angles. However, without integral control for translation, the pitch angle does not settle to the commanded input value as it also compensates to point the vehicle at center of the target for the error in Z. Figure 4-14 shows the thrust commands for one Y_b motor in response to the roll step input.

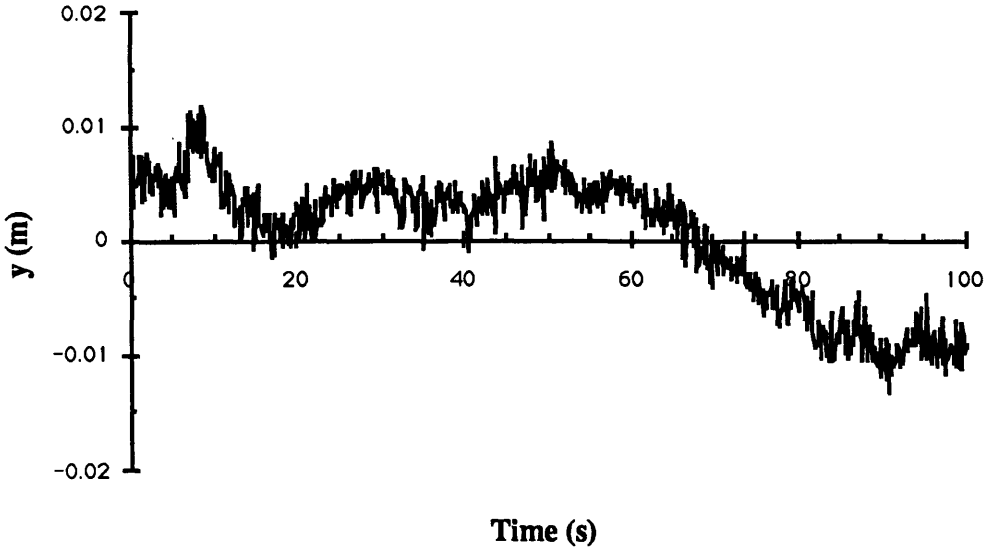


Figure 4-4: **Y Response to an X Step Input at T=5(s)**

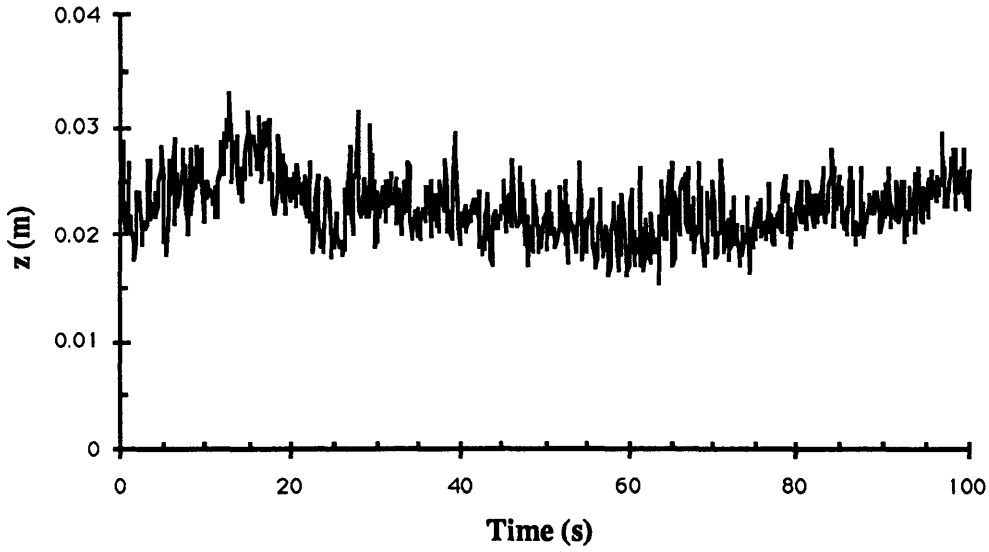


Figure 4-5: **Z Response to an X Step Input at T=5(s)**

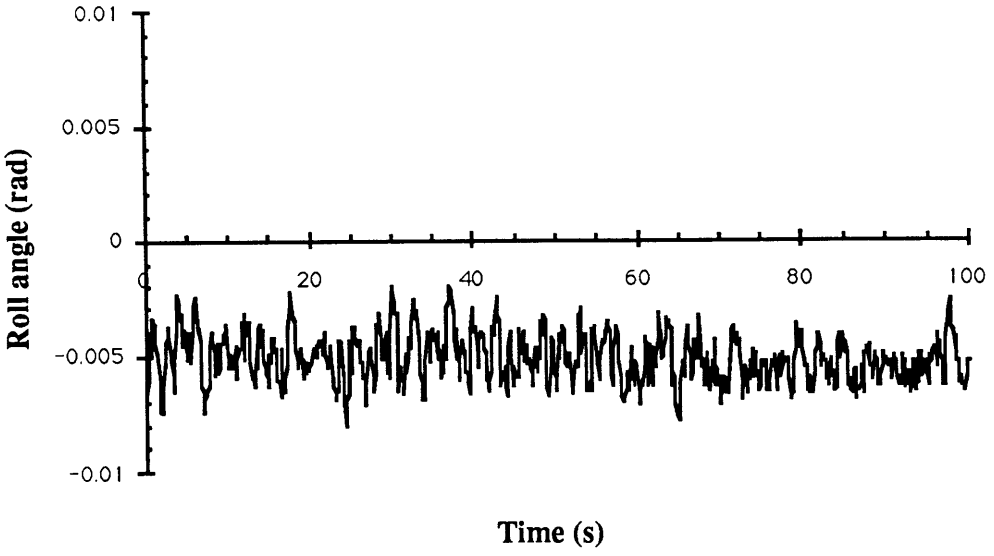


Figure 4-6: **Roll Response to an X Step Input at T=5(s)**

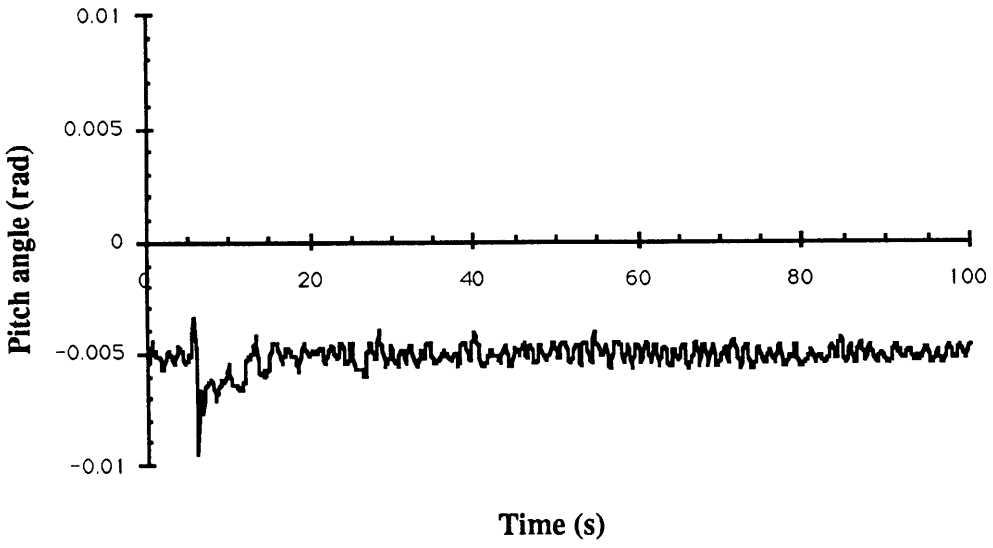


Figure 4-7: **Pitch Response to an X Step Input at T=5(s)**

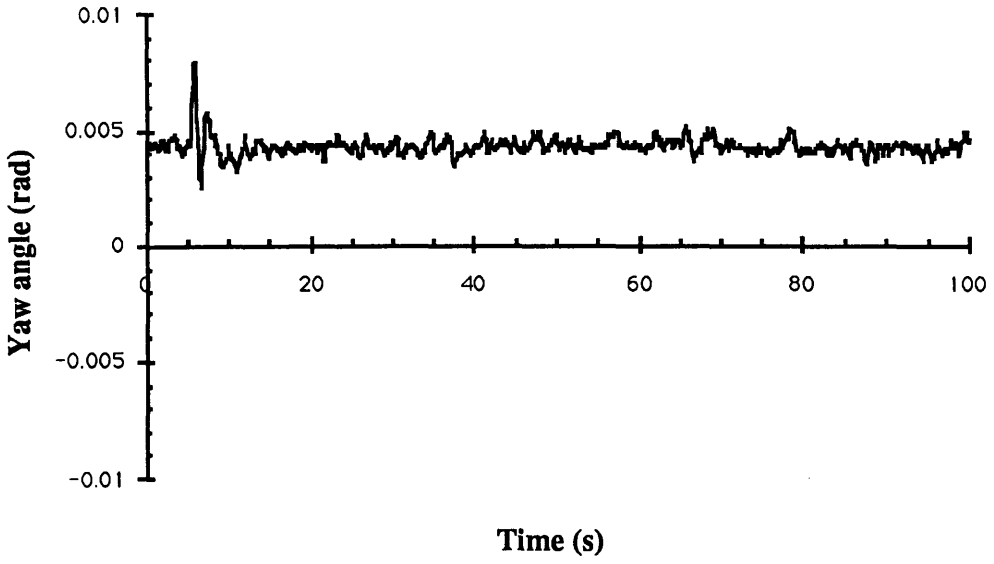


Figure 4-8: **Yaw Response to an X Step Input at T=5(s)**

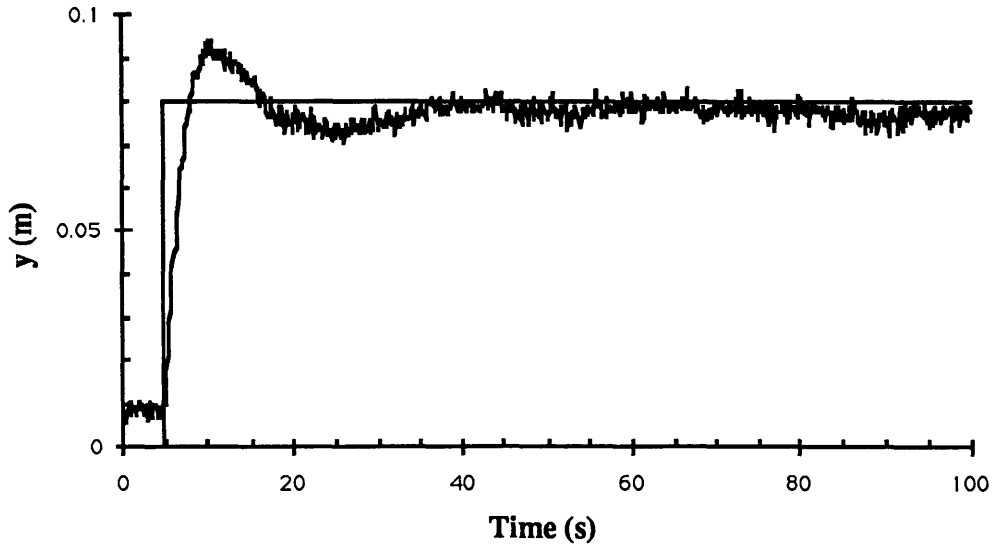


Figure 4-9: **Y Step Response**

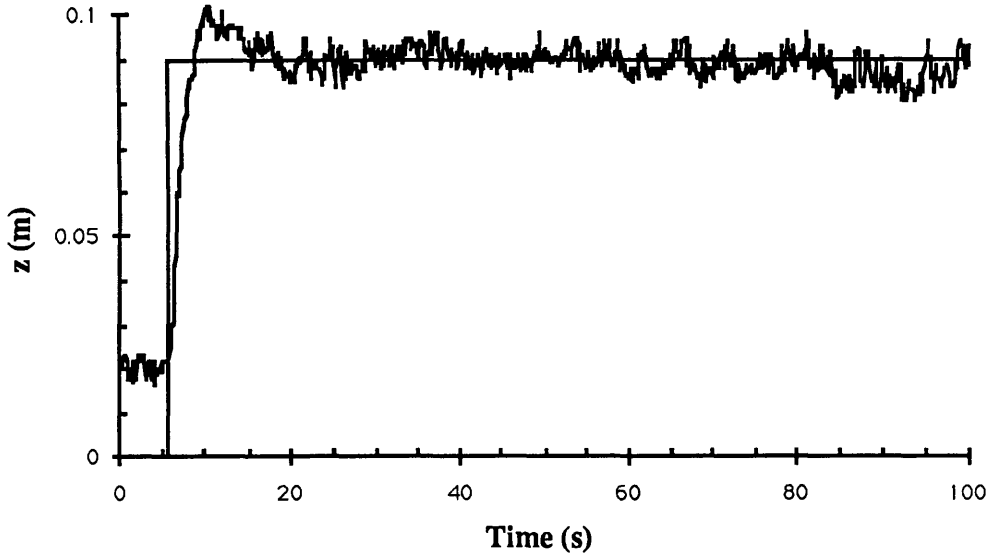


Figure 4-10: **Z Step Response**

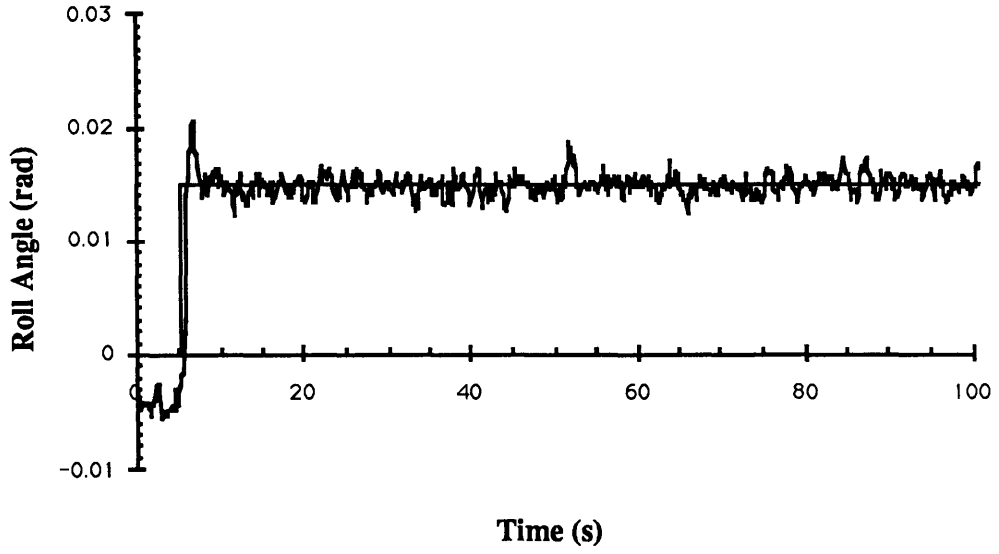


Figure 4-11: **Roll Step Response**

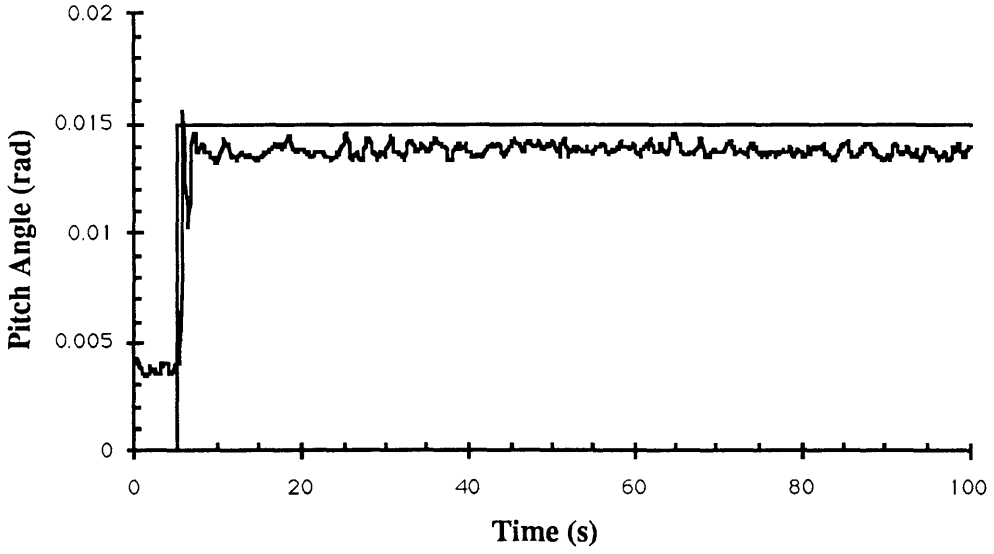


Figure 4-12: **Pitch Step Response**

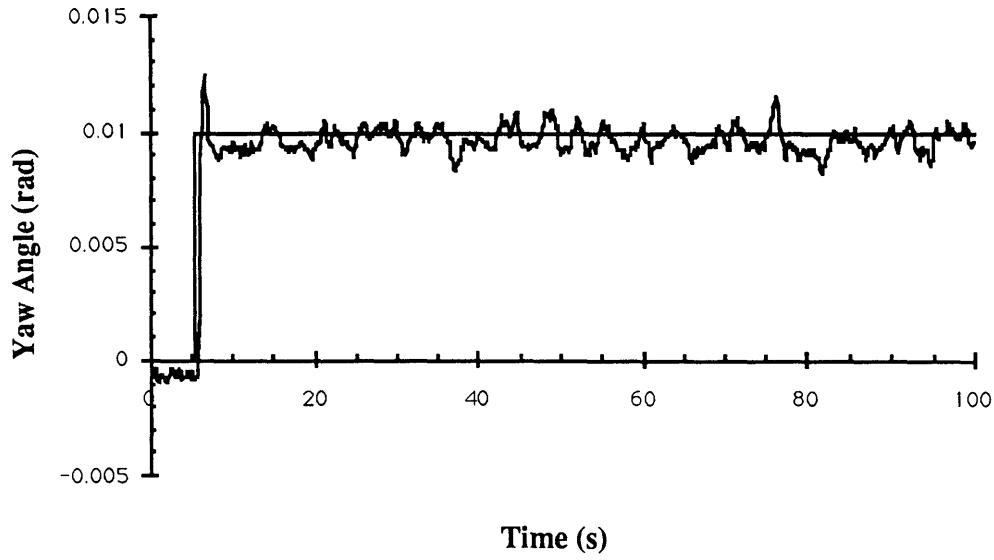


Figure 4-13: **Yaw Step Response**

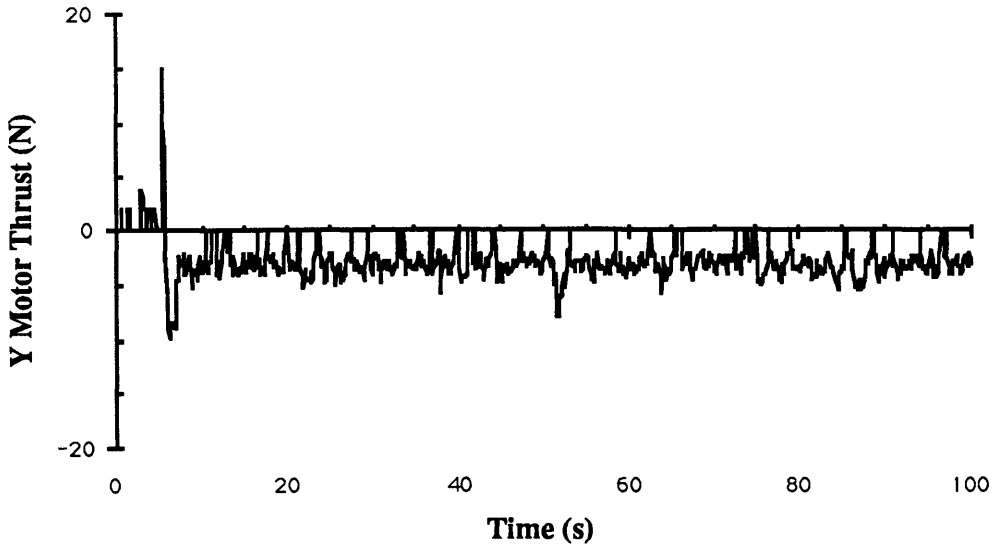


Figure 4-14: **Thrust Commands for Roll Step Input**

4.3 System Performance Observations

This section describes some interesting observations of the vision-based robot control system's performance and capabilities.

4.3.1 Effect of Camera Roll on State Measurement Noise

An important observation of the vision-based sensing system's performance is the increased accuracy of the state measurements when the camera is at a small roll angle. This is dramatically illustrated in Figure 4-15 for the Z state; the measurement noise is greatly reduced when a roll step is commanded 14 sec within the run. This reduction in measurement noise was observed for the other degrees of freedom as well.

This effect is a result of pixel quantization with the frame grabber and the navigation target. With no roll, the edges of the target's squares are horizontal and vertical. An example is illustrated in Figure 4-16. For a vertical motion, all the points along an horizontal edge will "jump" from one pixel row to the next creating a large apparent

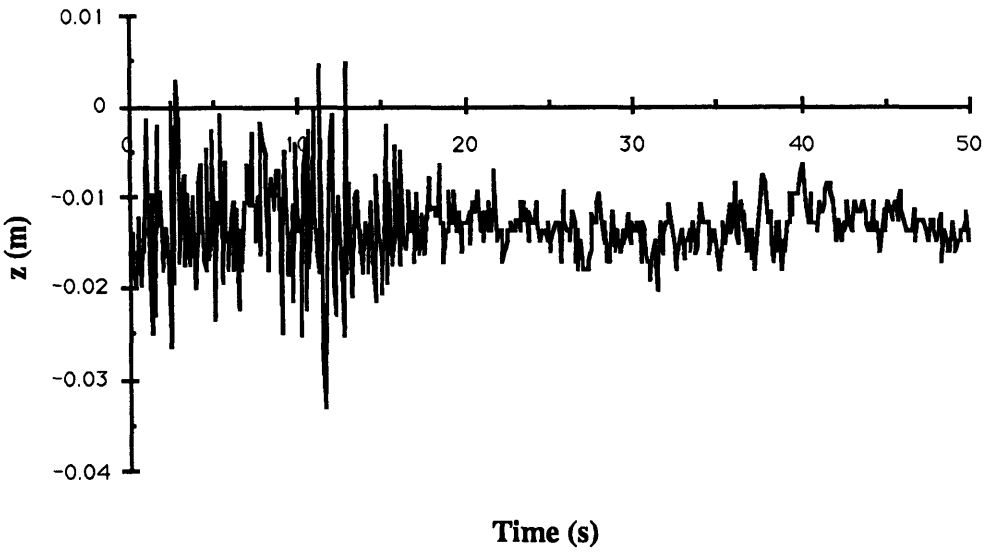


Figure 4-15: **Decrease in Z Measurement Noise with Roll Offset at T=14.3(s)**

change of the camera's position. Once a roll angle is introduced, these edges are no longer horizontal so edge locations will gradually change from one pixel row to the next along the edge. The noise is effectively cancelled out along an edge's samples. The analogous situation applies to vertical edges and the pixel columns.

4.3.2 Performance with Closed-Loop Thruster Control

Closed-loop control of the thrusters angular velocity plays a vital role in the vehicle position and attitude control system. Only a small amount of thrust is required for fine vehicle positioning and attitude control so the actual thrust commands from the controller are usually relatively small (as shown by Figures 4-3 and 4-14). With simple open-loop thrust control, the control system's commands must also provide enough torque to overcome internal stiction in the thruster motors which leads to sluggish motor response for small command inputs. It is also very difficult to accurately model the effects of stiction for a particular motor.

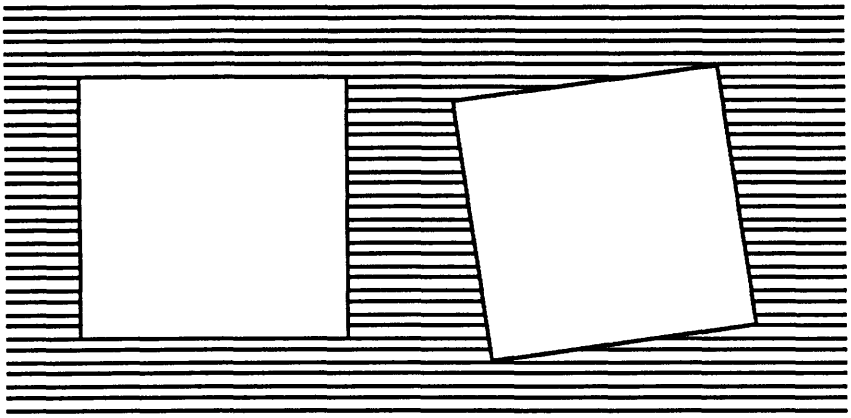


Figure 4-16: **Pixel Quantization and Edge Location Noise**

Before optical encoders were installed in all of STAR's thrusters, a six degree of freedom station-keeping vision-based control system was run with open-loop thrust control except for translation along the Z_b axis (the two motors along Z_b were equipped with encoders at the time of the test). In an attempt to overcome stiction, the open-loop motor commands were increased to a minimum value which ensured enough motor torque to spin the propellor. The control system was able to stably control the robot's position and attitude but with significantly poorer performance than the final system presented in Section 4.2. Figure 4-17 shows the vehicle's yaw and pitch angles during the automatic station-keeping. Unlike the control system with full closed-loop thrust control, the yaw and pitch angles are noticeably "twitching" back and forth during the control run; this was noticeably apparent while watching the vehicle under control. This *limit-cycling* is caused by the open-loop thrust commands which prevent fine control.

The solution to this problem (which is described in detail in Appendix B) is to implement closed-loop motor shaft velocity control. By adding feedback of the current motor shaft velocity with an optical encoder, a separate inner-loop control system ensures that the motor spins at the desired rate; all the vehicle's thrusters therefore respond quickly and uniformly to commands from the vision-based vehicle control system. This inner-loop control gives the fine control needed to precisely

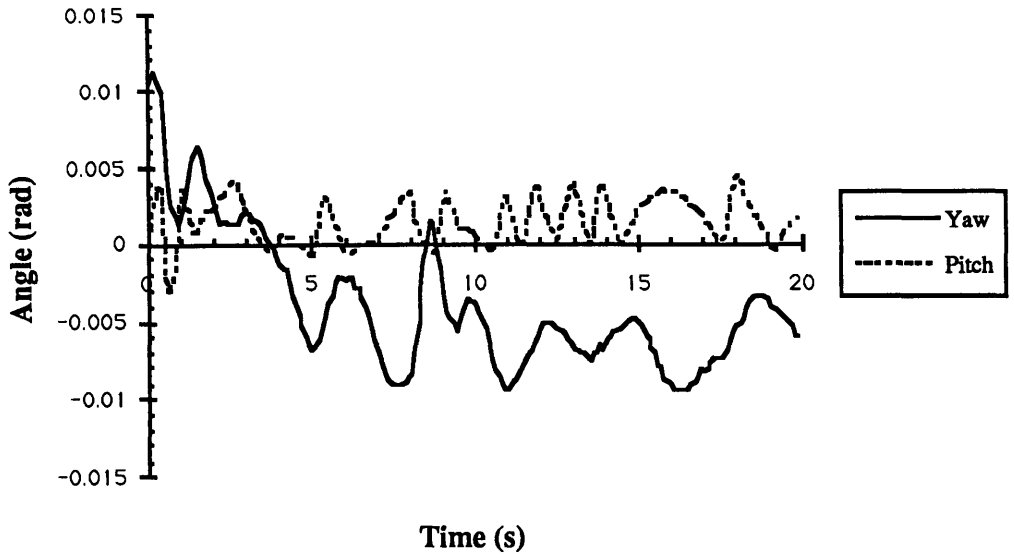


Figure 4-17: **Limitcycling with Open-Loop Thrust Commands**

position and orient the vehicle.

4.3.3 Disturbance Rejection

As desired, the vision-based position and orientation control system can reliably reject disturbance forces and torques applied to the vehicle. For one qualitative test, 12 lb of scuba weights were added to a corner of the vehicle while the vision system remained locked on the vision target (when balancing the vehicle, just .5 lb of weight dramatically changes the robot's buoyancy and rotational balance). Without integral control to correct the steady-state error, the vehicle did drift from the nominal position by a few centimeters while the vision sensing system remained locked on the navigation target.

4.3.4 Closed-Loop Teleoperated Control

During one pool test session, a test was run to qualitatively determine the precision of the vision-based control system for a teleoperated docking task. A mockup of a manipulator arm about 3 ft long (made of two fixed links with a 2.5 inch slot at the end as an end effector) was attached to the front of STAR in the field of view of the vision navigation camera. The pilot's goal for the test was to engage a fixed 1 inch diameter rod attached to the bottom of the navigation target with the "arm's" end effector. Manually flying the robot, it was impossible to grasp the rod without crashing into it. However, with closed-loop position and orientation control through the vision navigation system, the test was extremely successful. The rod was easily "grasped" since with the vision system the pilot was able to position the end effector with better than a half a centimeter precision. Once the rod was engaged, the end effector could be precisely slid along the length of the rod, disengaged, and re-engaged. Figures 4-18 and 4-19 are photographs of STAR, with the camera in its aluminum housing mounted on top of the vehicle, during these tests.

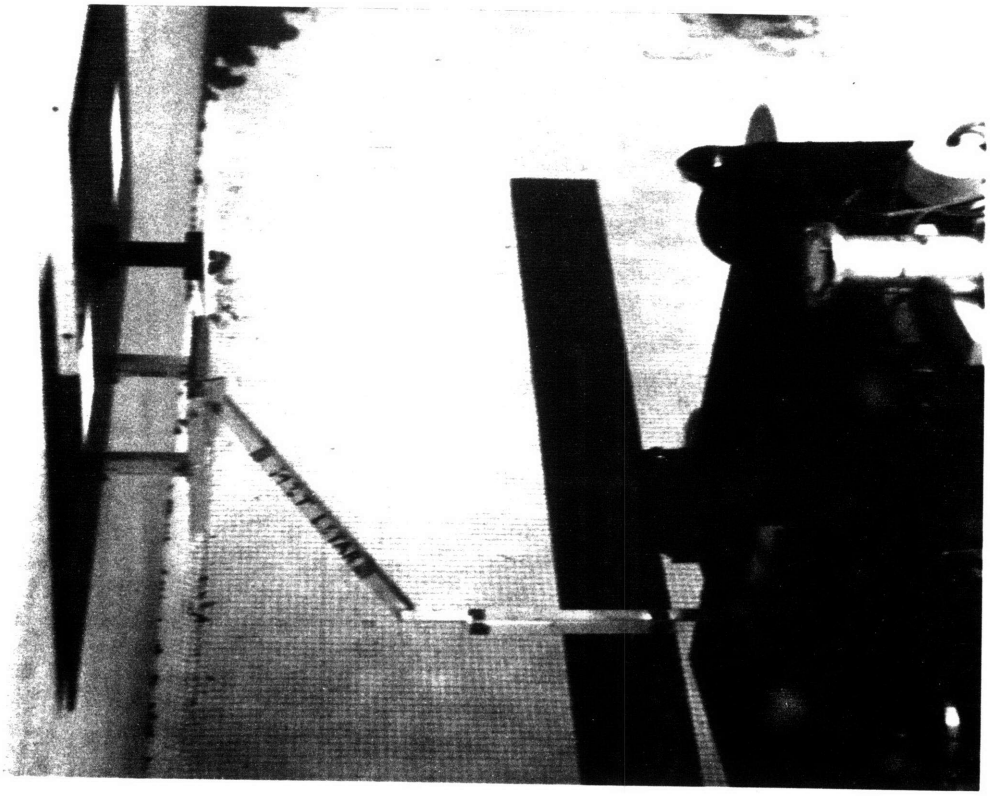


Figure 4-18: **STAR with Fixed Manipulator**

Figure 4-19: **Manipulator End Effector Positioning**

Chapter 5

Conclusions

The development of STAR's real-time navigation and control system, based on the linearized vision sensing technique, has led to a number of important results. The linearized vision-based position and orientation sensing system has the necessary accuracy and speed to be incorporated in a real-time robot control system. The vision-based control system provides accurate, precise, and robust six degree of freedom control, as shown in Chapter 4, with relatively basic hardware. The feasibility of using vision-based position and orientation sensing for a variety of free-flying robot navigation and control applications has been proven.

5.1 Real World Applications

The major accomplishment of the current vision-based navigation and control system is that it provides the necessary position and attitude control for developing the advanced capabilities needed for space and underwater robots. With the vision system, STAR is an ideal test-bed which can be extended to general underwater and space free-flying robot research. The current system can be directly applied to six degree of freedom control in the vicinity of a special navigation target. Accurate vision-based control can also be extended to such free-flying robotic tasks as flying along predetermined paths, obstacle avoidance, and serving as a free-flying platform for a manipulator arm (effectively adding an additional six controllable degrees of freedom

to the manipulator).

This technology can easily be developed for autonomous control with all the vision processing and control hardware on-board the vehicle. Such a vision system can also be used for certain tasks to provide closed-loop teleoperated position and orientation control as with the mock-up docking described in Section 4.3.4. Another exciting possibility is combining automatic and teleoperated control to relieve the operator's workload during teleoperation. For example, a human operator may manually control a manipulator end effector as the vehicle, under closed-loop vision-based control, is automatically repositioned and reoriented to provide the most convenient workspace for the operator.

5.2 Recommendations for Further Development

The major drawback of the current system is the fixed scanning method used to detect the current edge positions; this severely limits the range of motion of the vehicle under vision-based control. LSTAR is currently successfully developing and testing a scanning algorithm which tracks the corners of the target's squares in the camera's field of view. To reduce the search time, the corner locations from the previous image are used to find the current set of locations. The current vehicle state is then computed with the corresponding pseudoinverse of the linearized sensitivity matrix of the corner locations as discussed in Chapter 2. The speed of the enhanced tracking algorithm is comparable to the original simple system so that the sampling period of the vehicle control system is unchanged.

The current system relies on a friendly navigation target with easily detectable features and sensitivity to six degree of freedom motion. Although it may not be practical or possible to add such an artificial target for all robot tasks in space and underwater, vision-sensing can still be applied to many key robot navigation tasks by tailoring the vision system to the particular task. For example, assume a space robot must fly along a truss beam on a space station; the range to the beam, perpendicular distance from the beam, and relative roll angle can be computed with a vision sensing

system tracking the edges of the beam with the appropriate sensitivity matrix. With additional sensing, such as inertial sensors and another camera to track other features in the robot's environment, for measuring the vehicle's orientation and position along the beam, a vision-based navigation system can conceivably still be effectively used for this type of complicated task (such a task could easily be simulated by having STAR fly along the swimming lane stripes on the MIT pool floor).

For robot control within a small range to the worksite, there may be enough easily detectable natural features for six degree of freedom vision-based control. For example, LSTAR is also currently testing the linearized vision sensing system with a much simpler navigation target whereby the nominal state is defined relative to a single target square viewed from a large angle. When viewed from an angle, the relative positions and sizes of the front and back edges provide the necessary sensitivity to six degree of freedom motion. This system is directly applicable to flying to and capturing a box with a free-flying robot's manipulator.

There are several other issues which must also be considered for the development of vision-based space and underwater robot control. Lighting plays a significant role in space and deep underwater; for example, a vision navigation system for a space robot operating in Earth's orbit will have to contend with severe changes between darkness and brightness. A realistic system will also need to be able to incorporate several camera views which would allow continuous tracking of various navigation features in the environment as the view of the current feature is blocked by the vehicle's motion.

Another interesting area of research is the use of artificial intelligence (AI) in vision-based free-flying robot navigation and control. AI may be able to handle a number of foreseeable situations which would be catastrophic with a simple system:

- What if all or part of the navigation target is briefly blocked, for example by a manipulator arm or cloudy water?
- Which arbitrary features in the robot's environment are usable navigation features?
- How can the vehicle best recover from an tumble caused by a system failure

(for example a brief power failure which disables the computers)?

Many of the advanced capabilities that will be needed for future free-flying space and underwater robots can be handled with the development of more powerful vision-based navigation and control systems. STAR's real-time navigation and control system, based on the linearized vision algorithm, proves the feasibility of obtaining accurate, precise, and robust performance from vision sensing with relatively basic hardware. The door is open for exciting advances in the use of machine vision with free-flying robots.

References

[1] Akin, David L., "Neutral Buoyancy Simulation of Space Telerobotic Operations," Proceedings of the SPIE Technical Conference on Cooperative Intelligent Robotics in Space II, SPIE Intelligent Robotic Systems Conference, Boston, MA, November 11-15, 1991, SPIE Paper No. 1612-42.

[2] Alexander, H.L., "Experiments in Teleoperator and Autonomous Control of Space Robotic Vehicles," Proceedings of the International Symposium on Artificial Intelligence, Robotics, and Automation in Space (i-SAIRAS '90), November 18-20, 1990, Kobe, Japan.

[3] Alexander, H.L., Azarbayejani, A.J., and Weigl, H., "Kalman Filter-Based Machine Vision for Control of Free-Flying Unmanned Remote Vehicles," American Control Conference, Chicago, IL, June 24-26, 1992.

[4] Alexander, H.L., Weigl, H. and Eberly, K., "Vision-Based Control of a Neutrally-Buoyant Space Robot Simulator Vehicle," Proceedings of the SPIE Technical Conference on Cooperative Intelligent Robotics in Space II SPIE Intelligent Robotic Systems Conference, Boston, MA, November 11-15, 1991, SPIE Paper No. 1612-25.

[5] Atkins, Ella M., "Design and Implementation of a Multiprocessor System for Position and Attitude Control of an Underwater Robotic Vehicle", S.M. Thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, May 1990.

[6] Azarbayejani, Ali, "Model-based Vision Navigation for a Free-Flying Robot,"

SM Thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, August 1991.

[7] Azarbayejani, A.J. and Alexander, H.L., "Vision Navigator for Free-Flying Robots," Proceedings of the SPIE Technical Conference on Cooperative Intelligent Robotics in Space II, SPIE Intelligent Robotic Systems Conference, Boston, MA, November 11-15, 1991, SPIE Paper No. 1607-65.

[8] Craig, John J., *Introduction to Robotics: Mechanics and Control*, Second Edition, Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.

[9] 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.

[10] Franklin, Gene, et. al., *Digital Control of Dynamic Systems*, Second Edition, Addison-Wesley Publishing Company, Reading, Massachusetts, 1990.

[11] Halliday, D. and Resnick, R., *Physics, Part II*, Third Edition, John Wiley and Sons, Inc., New York, New York, 1978.

[12] Horn, Berthold K.P., *Robot Vision*, The MIT Press, Cambridge, MA, 1986.

[13] Machlis, Matthew, "Investigation of Visual Interface Issues in Teleoperation Using a Virtual Teleoperator," SM Thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, August 1991.

[14] St. John-Olcayto, E., "Machine Vision for Space Robotics Applications," SM Thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, May 1990.

[15] Strang, Gilbert, *Linear Algebra and Its Applications*, Third Edition, Harcourt Brace Jovanovich Publishing Company, Orlando, Florida, 1988.

Appendix A

Direct Sensing of Vehicle State Changes from Edge Motion

A very simple algorithm was developed to detect changes in the vehicle position and orientation directly from the location of the navigation target's edges in the video image. This method is computationally fast and extremely simple; it was developed as a preliminary vision sensing method which is intuitively simpler to debug than using the pseudoinverse of the linearized sensitivity matrix as discussed in Chapter 2.

To determine by how many pixels an edge has moved from its nominal position, n measurements are taken in the camera reference frame along each of the target's squares. In Figure A-1 the search scans, e_i which determine the number of pixels an edge has moved, are labelled.

A.1 Orientation Sensing

To determine the changes in the vehicle's orientation, the angles are defined as the pointing angles at the center square. Each angle change is computed from the average of the pixel locations along the appropriate edges. For example, a positive pitch vehicle rotation is detected by the motion of the center square's horizontal edges in the positive Y_c direction (Section 2.4.1 describes the orientation of the navigation reference frames). Similarly, positive yaw rotation is detected from the average motion of the

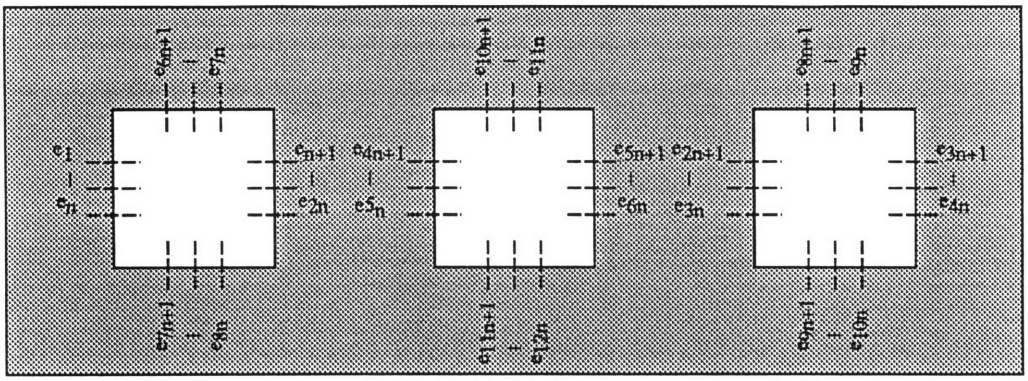


Figure A-1: **Edge Location Scan Labels**

center square's vertical edges in the negative X_c direction. Positive roll is detected by motion of the outer squares' horizontal edges; the edges move in the positive Y_c direction on the left square and in the negative Y_c direction for the right square. This is shown in the equations below:

$$\alpha_b \propto \left(\sum_{i=6n+1}^{7n} e_i + \sum_{i=7n+1}^{8n} e_i - \sum_{i=8n+1}^{9n} e_i - \sum_{i=9n+1}^{10n} e_i \right) /4n \tag{A.1}$$

$$\beta_b \propto \left(\sum_{i=10n+1}^{11n} e_i + \sum_{i=11n+1}^{12n} e_i \right) /2n \tag{A.2}$$

$$\gamma_b \propto \left(- \sum_{i=4n+1}^{5n} e_i - \sum_{i=5n+1}^{6n} e_i \right) /2n \tag{A.3}$$

A.2 Position Sensing

Similar averages are taken to determine position changes from the target's edge locations. Motion along the X_b axis is determined by the changing size of the target's three squares; as the vehicle moves closer to the target along the X_b axis, the three squares will become larger in the image. Motion along the Y_b and Z_b directions is decoupled from yaw and pitch rotations by determining the difference in the average location changes of the center square's edges and the outer square's edges. The

position sensing equations are given below:

$$
\begin{aligned}
x_b \;\propto\; & -\left(-\sum_{i=1}^{n} e_i - \sum_{i=6n+1}^{7n} e_i + \sum_{i=n+1}^{2n} e_i + \sum_{i=7n+1}^{8n} e_i\right)/4n + \\
& -\left(-\sum_{i=2n+1}^{3n} e_i - \sum_{i=8n+1}^{9n} e_i + \sum_{i=3n+1}^{4n} e_i + \sum_{i=9n+1}^{10n} e_i\right)/4n + \\
& \left(-\sum_{i=4n+1}^{5n} e_i - \sum_{i=10n+1}^{11n} e_i + \sum_{i=5n+1}^{6n} e_i + \sum_{i=11n+1}^{12n} e_i\right)/4n \qquad \text{(A.4)} \\
y_b \;\propto\; & -\left(-\sum_{i=1}^{n} e_i - \sum_{i=n+1}^{2n} e_i - \sum_{i=2n+1}^{3n} e_i - \sum_{i=3n+1}^{4n} e_i\right)/4n + \\
& \left(-\sum_{i=4n+1}^{5n} e_i - \sum_{i=5n+1}^{6n} e_i\right)/2n \qquad \text{(A.5)} \\
z_b \;\propto\; & -\left(-\sum_{i=6n+1}^{7n} e_i - \sum_{i=7n+1}^{8n} e_i - \sum_{i=8n+1}^{9n} e_i - \sum_{i=9n+1}^{10n} e_i\right)/4n + \\
& \left(-\sum_{i=10n+1}^{11n} e_i - \sum_{i=11n+1}^{12n} e_i\right)/2n \qquad \text{(A.6)}
\end{aligned}
$$

Appendix B

Closed-Loop Thruster Angular Velocity Control

In order to ensure fast, controllable, and uniform response from all eight of STAR's propellor motors, closed-loop angular velocity control was added to each of the thrusters. As shown by experimental results in Chapter 4, this inner-loop control of the thrusters greatly improves the performance of the vehicle's position and attitude control system. The development of the thruster control system was aided by Paul Stach, an undergraduate student in LSTAR who has been heavily involved in the design and construction of STAR's propulsion system.

The two main components of each thruster's control system are an optical encoder mounted in the motor and a National Semiconductor LM629 Motion Controller interfaced to the on-board computer. The LM629 controller actively changes the current applied to the motor to ensure that the motor spins at a desired velocity which in turn corresponds to a desired thrust. The optical encoder measures the current angular velocity of the motor shaft which is used in the feed-back path of the LM629 control system.

B.1 LM629 Compensator

The LM629 Motion Controller uses a simple proportional/derivative (PD) compensator to control the rotational velocity of the motor shaft. For the Minn Kota trolling motors, suitable compensator control gains were experimentally found to be $K_p = 30$ for the proportional gain and $K_d = 100$ for the derivative gain.

It is important to limit the maximum thruster speed commanded by the LM629. The maximum rotational speed that can be achieved with the propellor motors underwater is significantly slower than the maximum speed out of water. The control system must be limited to the maximum underwater speed. In velocity mode, the LM629 is actually continuously controlling the angular shaft position to correspond to the desired velocity. Once too large a velocity has been commanded, the LM629 will continue to update the shaft position in an attempt to catch up to the commanded value even when a new velocity has been commanded. This has potentially disasterous effects for control of the robot. The maximum velocity commands of the LM629 were therefore limited to 3200 RPM for underwater operation.

B.2 Optical Encoder

There are two main components of an optical encoder: the encoder wheel and the sensing module. The encoder wheel is mounted on the motor shaft; it has a series of equally sized and spaced slits along its outer edge. The edge of the wheel is then slid into a gap in the encoder module. The module contains a light-emmitting-diode (LED) and a photo transistor which are located on opposite sides of the encoder wheel slits. As the wheel spins, the light and dark regions detected by the optical system produces a square wave; the duty cycle of the square wave determines the shaft rotational speed. There is also a second series of slits (and optical detectors) which are slightly offset from the outer set of slits. With two square waves, the resolution is increased and direction of motion can be determined. The two optical encoder square waves are fed directly back to the LM629 motion control chip; this is very convenient

since the decoding of the shaft angular velocity is handled by the LM629 without any additional logic. It is therefore important to have "clean" square wave signals from the optical encoder with sharp, right angles and little noise.

B.2.1 Choice of Optical Encoder

There are several factors which determined the choice of optical encoders to be used for STAR's thruster motors. There is very limited space available in the prefabricated Minn Kota trolling motors. The main requirements were therefore small size of the encoder module and a compatible encoder wheel shaft diameter. It is also important that the quality of the encoder signals is not affected by the axial shaft play of the motors and the noisy electrical environment of the small motors. Low cost was also a major concern.

These requirements are best met by the Hewlett Packard HEDS-9140-A00 encoder module and HEDS-5140-A13 encoder wheel. The module fits well into the trolling motor (see Section B.2.2 below); this model wheel has an 8 mm shaft diameter which is very close to the 5/16 inch motor shaft diameter. The 9140/5140 series also has the advantage of a small number of slits on the encoder wheel (500 counts/rev for the models used). Unlike for angular position control, a lower resolution is desirable for velocity control since there are usually many revolutions within a sampling period (T) (the LM629 has a limit for the maximum number of counts for T) and finer slit patterns are more susceptible to signal corruption from misalignment of the encoder module and shaft play.

B.2.2 Encoder Module Mounting

The only space large enough for mounting the optical encoder in the trolling motors is the end-bell of the motor (Figure B-1 labels the main motor sections); the cable to the optical encoder is passed through a waterproof seal in the end-bell to avoid running the sensitive signals along the motor magnets and coils. A mounting plate is attached to the end-bell which rests on the thrust bearing housing in the center of

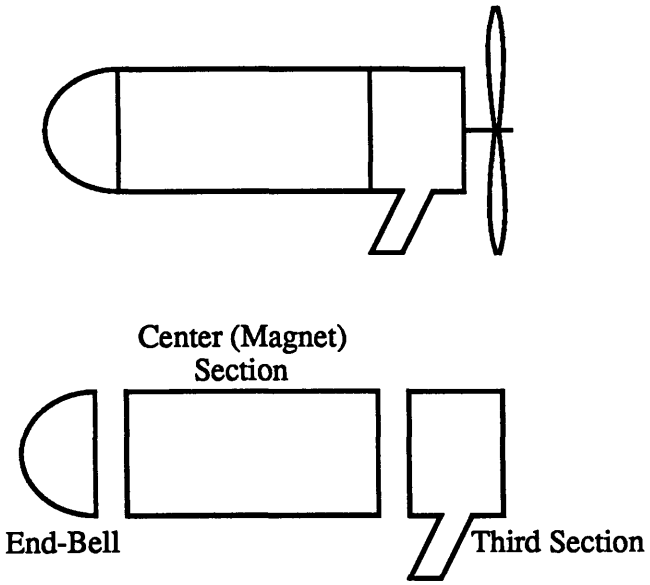


Figure B-1: **Trolling Motor Sections**

the end-bell as shown in Figure B-2. Four 4-40 thread holes are tapped into the posts in the corners of the end-bell to hold the mounting plate. The encoder module must then be carefully aligned with the motor shaft according to specifications to ensure good quality signals.

Axial shaft play must be considered when mounting the encoder wheel within the gap of the module. To ensure that the wheel remains within the specified height limits of the module gap, but still allow for thermal expansion, the shaft play of the thrusters has to be reduced to 0.03 inch with spacer washers. When mounting the encoder wheel, a 0.235 inch thick spacer is used between the mounting plate and wheel. The spacer ensures that the encoder wheel can move by 0.015 inch in both directions from the center of the module gap (as discussed below, the motor is assembled with the motor shaft firmly against the thrust bearing of the end-bell keeping the wheel at the lower height limit).

The following procedure was developed by trial and error for assembling and disassembling the trolling motors once the encoder module has been mounted in the motor end-bell. When assembling the motor with the optical encoder, it is important

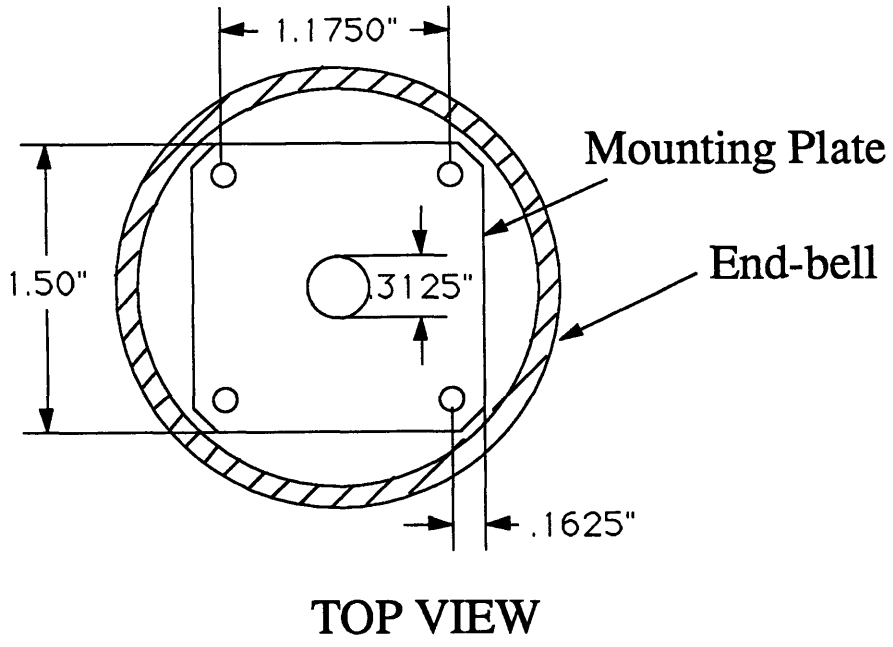


Figure B-2: **Encoder Module Mounting Plate**

to have a perpendicular motor shaft; the shaft is therefore mounted in the chuck of a drill-press and the motor assembled on the drill-press table. Also since the magnets in the center motor section constantly pull the shaft up from the end-bell, the wheel would be damaged once it is in the gap of the encoder module. To avoid this problem, the motor shaft is constantly held against the thrust bearing of the end-bell throughout the assembly process. The motor assembly and disassembly procedures are given below.

Motor Assembly with Encoder Module and Wheel

1. Tie back motor brushes with stiff wire providing a large enough gap for the shaft and spacer washers.

2. Mark locations of the screw holes on the outside of the motor end-bell with a marker.

3. Mount the top of the motor shaft in drill press chuck. Slip the motor's center section (with the magnets) over the shaft.

4. Check all o-rings and add vacuum grease if necessary.

5. Rest the encoder wheel with the proper orientation on the spacer. Slip the end of the shaft through the encoder wheel and firmly against the thrust bearing of the end-bell which rests on the drill press table. Lock the drill press into position to eliminate *ALL* shaft play.

6. Tighten encoder wheel set-screw and remove spacer.

7. Lower motor's center section and align with screw holes in end-bell.

8. While firmly holding shaft against end-bell (as perpendicularly as possible), raise chuck, add the necessary spacer washers, and place the third motor section with screws onto the shaft.

9. Re-lock the drill press with shaft again firmly against the end-bell's thrust bearing.

10. Untie brushes and seal motor by tightening motor screws.

Motor and Optical Encoder Disassembly

1. Mount motor shaft in chuck and lock drill press with the motor shaft firmly against the end-bell which rests on the drill press table.

2. Unscrew motor screws.

3. Raise third motor section and tie back motor brushes with stiff wire so that the shaft with spacer washers can be slipped past the brushes.

4. While firmly holding shaft against end-bell (as perpendicularly as possible), raise chuck and remove third motor section.

5. Re-lock the drill press chuck with the shaft firmly against end-bell thrust bearing.

6. Raise center motor section from the end-bell.

7. Carefully loosen the encoder wheel set screw.

8. Release chuck and gently slip shaft through the encoder wheel and remove wheel from encoder module.