Designing Visually Servoed Tracking to Augment Camera Teleoperators

A Thesis

Submitted to the Faculty

of

Drexel University

by

Ion Rares Stanciu

in partial fulfillment of the

requirements for the degree

of

Doctor of Philosophy

December 2004

Dedications

To my parents

Acknowledgements

I have been truly encouraged and blessed by several people along the way of my research. Professor Paul Oh never gave up on me. He brought out the thesis and ideas which were locked inside me. Thank you for all your patient, for your guidance and for mentoring me as I grew as a researcher. Dr. Oh, you challenged me to become a researcher and painted in me a picture of the next generation of robotics.

I would also like to thank my committee for the help, guidance and valuable suggestions you gave me. Professor Harry Kwatny, thank you for your big help in the controls field as well as in modelling my system. Without your expertise I would still be stuck on the boom model. Professor Sorin Siegler, thank you for your help regarding the dynamics of my boom. Professor Andrew Hicks, thank you for helping me better understand the mathematical aspects of vision. Professor Ajmal Yousuff, thank you for your being a model in teaching. You were always approachable and helpful. Professor Belta, you always advised me and you were always approachable. I would also like to thank Professor Jaydev Desai for allowing me to use the PRISM LAB facilities as well as the Mitsubishi robotic arm. I wish to thank Professor Lau as well for his constant encouragement and help along the way. I wish to thank Professor Choi for his constant support and help.

I am also grateful to the others members of the department: Kathie Donahue, Stephanie Delaney, Joanne Ferroni and Stephanos Karas for their loving and moral support.

Also I want to thank my lab colleagues and friends, Dr. Imhauser, Dr. Kennedy, Mr. Tie Hu, Mr. Gregory Tholey, Ms. Teeranoot Chantasopeephan, Mr. James Hing, Mr. Anand Pillarsetti, Mr. Keith Sevcik and Mr. Marius Kloetzer for all their support and encouragement. I would like to address special thanks to Mr. Suba Thomas for helping and advising me so many times. I wish to thank all of you for your constant availability, help and support. Special thanks to my colleague and friend Mr. William Green for much advice and support.

In the end, I would like to thank to my wife Adriana for her faith in me, her help, as

well as her constant encouragement. I also want to thank my family, my mother, my sister and my brother-in-law, for all their support.

Table of Contents

List of Tables

List of Figures

Abstract

Designing Visually Servoed Tracking to Augment Camera Teleoperators

Ion Rares Stanciu

Paul Y. Oh

Robots have now far more impact in humans life then ten years ago. Vacuum cleaning robots are already well known. Making today's robots to work unassisted requires appropriate visual servoing architecture. In the past, a lot of efforts were directed towards designing controllers that relies exclusively on image data. Still most robots are servoed kinematically using joint data.

Visual servoing architecture has applications not only in robotics. Video cameras are often mounted on platforms that can move like rovers, booms, gantries and aircrafts. People can operate such platforms to capture desired views of a scene or a target. To avoid collisions, with the environment and occlusions, such platforms demands much skill. Visual-servoing some degrees-of-freedom may reduce the operator burden and improve tracking. We call this concept *human-in-the-loop* visual servoing.

Human-in-the-loop systems involve an operator who manipulates a device for desired tasks based on feedback from the device and environment. For example, devices like rovers gantries and aircrafts possess a video camera. The task is to control maneuver the vehicle and position the camera to obtain desired fields of view. To overcome joint limits, avoid collisions and ensure occlusion-free views, these devices are typically equipped with redundant degrees-of-freedom. Tracking moving subjects with such systems is a challenging task and requires a well skilled operator. In this approach, we use computer vision techniques to visually servo the camera. The net effect is that the operator just focuses on safely manipulating the boom and dolly while computer-control automatically servos the camera.

1: Introduction

1.1 Preamble and Objective

A short time before the writing of this thesis, NASA's Spirit rover landed on Mars. This was another great achievement in robotics. Unlike Sojourner, which was completely teleoperated from Earth, Spirit had its own vision system. The step from Sojourner to Spirit was towards making robots work autonomously.

Almost all of today's robots operate in factories where the environment is contrived to suit the robot. Robots have less impact in applications where the object placement or environment is not accurately controlled. This limitation is a result of their lack of sensors. It is well known that sensor integration is extremely important for increasing the versatility and application domain of robots. This is basically the domain which provides a lot of research opportunities. One specific type of sensor that is subject to a lot of research, is vision.

Vision is a robotic sensor that mimics the human sense of vision. It allows for noncontact measurement of the environment. The first application of computer vision on robots was by Hill and Park in the 1970s [14]. Over the last three decades, vision has been used to demonstrate robots playing ping-pong, avoiding obstacles and automatically landing aircraft. Applications which include vision sensors in cars have also been studied. Despite a considerable knowledge base and accepted design methods, vision-based control in unstructured and dynamic environments remains an open problem. Visual servoing, the problem of servoing the robot based on *real-time* image data, is not new. Over the past fifteen years many systems have been developed, visually servoing robots to juggle [28], play ping-pong, pick up a moving toy train, play table-top soccer, harvest fruit and drive cars. Still, dynamic configurations and camera systems featuring a human in the control loop have not been considered in depth. This thesis explores such cases.

In the 1970s control designers focused their efforts toward overcoming large image

processing latencies associated with slow frame rates. Digital signal processing based vision boards became avaliable in the early 1990s. Still, image data was not very robust. So, effort was focused on developing image processing algorithms using stochastic prediction and estimation techniques and ”getting things to work”. With much of the latency issues having been overcome, vision research focus is on making robotic camera systems act more intelligently.

For the most part, visually-servoed robots rely exclusively on image data to servo its joints. By contrast, most factory robots are kinematically servoed; solely relying on their encoder data. People display interesting behaviors when performing vision-related tasks, suggesting that humans employ both image and kinematic data. In tracking moving objects humans tend to use several joints (eye, neck and torso) and their motions suggests a kinematic joint coupling. For example, both the eyes and the neck typically pan in the same direction. Also, eye movements tend to start before neck motions. This way, people track pretty well despite large variations in target motions.

Using of-the-shelf hardware, systems that track moving objects were studied. A robot was often used to move the camera which was attached to its end effector. Here, the manipulator updates the camera positions and orientations to maintain a desired field of view of the target. Given a target with known dimensions, such as a planar block, an image Jacobian, which maps differential changes between pixel and task spaces, can be computed. The net result is that the target’s 6 degrees-of-freedom motion can be computed and tracked by a 6 degrees-of-freedom eye-in-hand manipulator. *A problem that arises is what to do when the manipulator possess more than 6 degrees-of-freedom or if some of the manipulator’s degrees-of-freedom are human-controlled*. The former problem involves degrees-of-freedom redundancy. Here, there are additional joints that could be invoked to configure a camera pose. Such degrees-of-freedom can be exploited to overcome joint limitations.

The latter problem involves human-in-the-loop systems. For example, cameras are

often mounted on platforms like booms, rovers, aircrafts, and submersibles for applications such as broadcasting, exploration, and surveillance. A skilled operator maneuvers the platform to capture the desired fields-of-views. As such, the human becomes a component in the closed loop control of the camera. The objective of this thesis is to design a human operated robot vision system that helps the operator to track moving objects. Here, computer vision is involved to control the camera's two degrees-of-freedom.

1.2 Motivation and Research Platform

Human-in-the-loop systems involve an operator who manipulates a device for desired tasks based on feedback from the device and environment. For example, devices like rovers, gantries, and aircraft possess a video camera where the task is to maneuver the vehicle and position the camera to obtain desired fields-of-view. Such tasks have applications in areas like broadcasting, inspection and exploration. Such device-mounted camera systems often possess many degrees of freedom (DOF) because it is important to capture as many fields-of-view as possible. To overcome joint limits, avoid collisions and ensure occlusion-free views, these devices are typically equipped with redundant DOF. Tracking moving subjects with such systems is a challenging task because it requires a well skilled operator who must manually coordinate multiple joints. Tracking performance becomes limited to how quickly the operator can manipulate redundant DOF.

As examples, two such systems are presented in Figures 1.1 and 1.2. In the first picture, a traffic helicopter is shown. A gyrostabilized camera is attached to it. Such platforms are used for traffic surveillance. This system entails two people in normal operation. The pilot safely guides the helicopter to avoid hitting buildings or bridges while a copilot manually manipulates the camera to keep a ground target in focus.

The second system (shown in Figure 1.2) is a typical camera boom used in the broadcasting industry. The camera is shown in the the lower right corner of the image. This system entails two operators as well. The first rotates the boom and concentrates on

Figure 1.1: A traffic helicopter is often equipped with a camera. The pilot concentrates on safely flying the helicopter while an operator handles the camera.

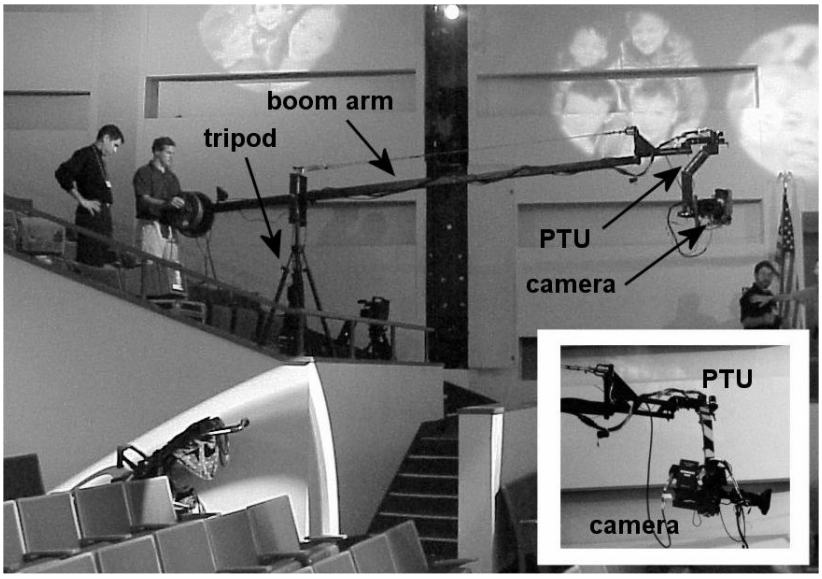

Figure 1.2: An operator can boom the arm horizontally and vertically to position the camera. The pan-tilt head (lower right inset) provides additional degrees-of-freedom.

avoiding collisions into people and objects. The second pans and tilts the camera.

The systems shown in Figures 1.1 and 1.2 are piloted by humans. The first operator ensures a safe coarse motion while the second coordinates fine camera motions to maintain the target in focus. These are examples of human-in-the-loop systems which involve an operator who manipulates a device for desired tasks based on feedback from the device and environment. Tracking performance becomes limited to how quickly the operator can manipulate redundant DOF.

Machine vision is a multi-disciplined problem. This thesis will not examine sub-problems like real-time image processing, image understanding or motion planning. They can present interesting challenges, but we have not found them to be particularly confounding with today's off-the-shelf vision hardware and software.

This thesis will present a way to deal with a human-in-the-loop system. Some preliminary experiments were done first. Using a very simple controller the camera was able to track slow moving objects. A feedforward controller with a feedback compensator is adopted to sidestep the challenges. Adding another controller to the system improves the performance even more.

1.3 System Description

Our particular interest in computer vision involves improving a camera operator's ability to track fast moving targets. *Visual-servoing* is used to control some DOF so that the operator has fewer joints to manipulate.

Our system is shown in Figure 1.3 and is a typical platform found in the broadcasting industry. The platform is composed of a 4-wheeled dolly, boom, motorized pan-tilt unit (PTU) and camera. The dolly can be pushed and steered. The 1.2 m long boom is linked to the dolly via a cylindrical pivot which allows the boom to sweep motions horizontally (pan) and vertically (tilt). Mounted on one end on the boom is a 2-DOF motorized PTU and video camera weighing 9.5 kg. The motors allow an operator to both pan and tilt the camera 360

Figure 1.3: An operator can boom the arm horizontally and vertically to position the camera. The pan-tilt head (lower left inset) provides additional degrees-of-freedom.

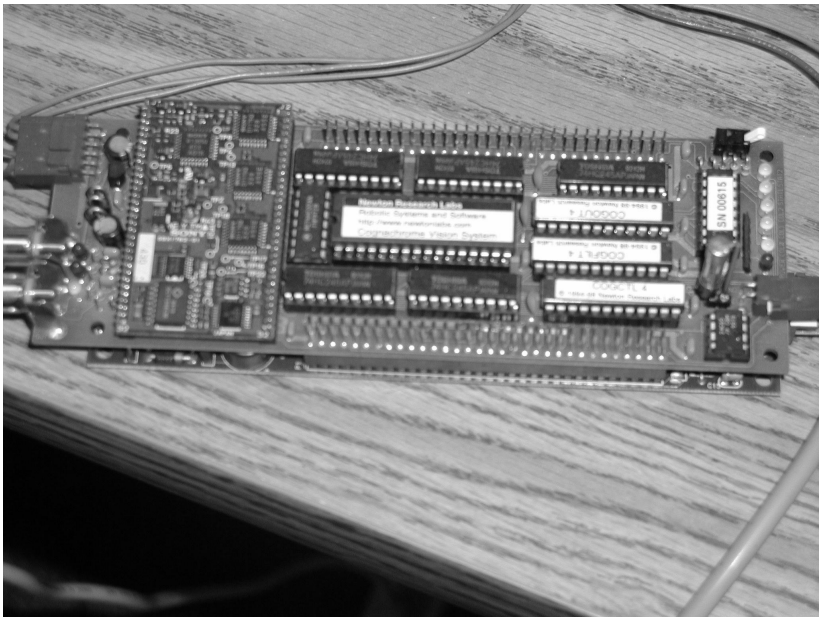

Figure 1.4: The Newton Board Color-Tracker. The board computes the target centroid. This value is sent to the PC via the serial port.

degrees at approximately 90 deg/sec. The PTU and camera are counterbalanced by 29.5 kg dumbbell plates mounted on the boom's opposite end.

Our particular interest is to apply *visual-servoing* to augment an operator's ability to track moving targets. Computer vision is used to control some DOF so that the operator has fewer DOF to manipulate.

Broadcast use of this boom-camera system normally entails one or more skilled personnel: (1) With a joystick, the operator servos the PTU to point the camera. A PC-104 small board computer and ISA bus motion control card allow for accurate and relatively fast camera rotations. (2) The operator physically pushes on the counterweighted end to boom the camera horizontally and vertically. This allows one to deliver a diverse range of camera views (e.g. shots looking down at the subject), overcome PTU joint limits, and capture occlusion-free views. (3) The operator can push and steer the dolly in case the boom and PTU are not enough to keep the target's image in the camera's field-of-view. The net effect is a human-in-the-loop camera system that possesses redundant degrees-of-freedom.

Tracking a moving object with a broadcast boom is a particularly challenging task. It requires one or more highly skilled operators who coordinate the platform's many degrees-of-freedom to keep the subject's image in the camera's field of view. Tracking performance is thus limited to how quickly the operator manipulates and coordinates multiple degrees-of-freedom. *The gap in the knowledge base is the absence of an analytical framework to design visually servoed eye-in-hand systems when redundant degrees-of-freedom or human operators are involved.*

This thesis applies computer vision to track a moving target with this broadcast boom-arm. The net effect is what we call *human-in-the-loop visual servoing* – the operator focuses on safely manipulating the boom and dolly while computer-controlled visual-servoing automatically points the pan-tilt camera at the target.

The first of our hypotheses was that the course and fine motions can be decoupled.

For image processing, a DSP board was used (see Figure 1.4). This board is able to

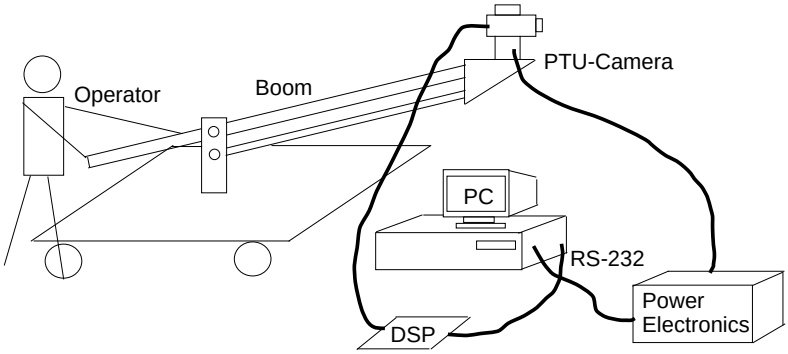

Figure 1.5: A schematic of our platform. The operator handles the boom while the vision system moves the PTU-camera and attempts to track the moving target.

perform color tracking. After training it for a specific color, this board is able to compute the target's centroid coordinates and send them to the PC using serial communication.

The second hypothesis was that using a feedforward control scheme with a feedback compensator will improve both the tracking performance and the stability of the system. The third hypothesis was that another feedback compensator improves the tracking performance.

A schematic of the system can be seen in Figure 1.5. While an operator rotates the boom the Cognachrome color-tracker processes the incoming image and sends the data to the computer, which performs a visually servoed camera rotation.

1.4 Thesis Significance and Outline

The vision setup is a working prototype which can track moving targets such objects and people without *a prior* knowledge of their trajectories or boom motion. The target is selected in the beginning of the program. Being able to track targets of diverse geometries over a wide range of target motions has many applications. Examples include surveillance (traffic helicopters), computer-assisted filming and mass media.

Since the camera is to be controlled, our setup falls into visual-servoing research.

Our approach may also encourage machine-vision researchers to revisit solutions that are well known in robotics for applications in the visual-servoing domain.

The rest of the thesis progresses as follows. Chapter 2 reviews related visual-servoing and human-in-the-loop literature, highlighting key design practices. Preliminary experiments with this system are described in Chapter 3. Conclusions are presented as well. The design of the feedforward controller is presented in Chapter 4. Experiment descriptions and conclusions are presented as well. A theoretical model for our boom is developed in Chapter 5. The experimental validation is described here as well. Chapter 6 describes the design of a new controller. Adding the new controller to the feedforward schematic improves the performance of the system. Chapter 7 shows a comparison between an unexperienced and an experienced operator with and without vision system. Chapter 8 presents some ways to improve this camera system.

2: Related Visual Servoing Research

Our goal is an automated vision-system to track moving targets like robotic heads, tools, persons and objects which move over a large space. In this endeavor, the literature reveals the multi-disciplinary nature of visually-servoed robotics. Section 2.1 presents three categories which are important.

2.1 Broad Overview

Adding vision to a robot is intuitively attractive. It is not a surprise that the first experiments using this idea began in 1970s [24]. But, vision and computer hardware were prohibitively expensive and slow. Real-time visually servoed robots began their debut in the late 1980s. After off-the-shelf vision hardware become accessible, this equipment helped researchers to define this very exciting area. What happened in the past three decades are elements that constitute a robot-camera system.

2.1.1 Camera Setup

Our particular vision interest required robotically positioning a camera. This is somewhat close to what is called an eye-in-hand setup, where the camera is mounted on a robot's end-effector. In such a setup images are acquired and transformed into a robotic motion that servos the camera into a desired pose. Eye-in-hand configurations were commonly used to regulate a desired camera pose relative to its target. A common problem in this configuration is robot endpoint vibrations that induce noise and corrupt the image data. This problem is typically handled with appropriate filters [2].

By contrast, a fixed camera setup was also used. In this case, the camera is mounted on a tripod to have a full view of the entire robot's workspace. One disadvantage is limited pose and view. An alternative is the use of a pan-tilt system. Often, calibration must be done each time the tripod and camera are repositioned.

2.1.2 Image Processing

Image data is typically acquired using a charge-coupled device (CCD) camera and digitized using a framegrabber. Real-time rates (30 frames/sec) are common with today's off-the-shelf hardware. Processing the data was, until recently, the bottleneck in visual-servoing. *Image features* are characteristics of the target (lines, points, etc.) that can be extracted from an image. Typically, they correspond to the projection of the target's physical features (edges) onto the camera's image plane. Real-time constraints often forced using target with fiducial points. Often the image background had to be prepared to allow noiseless and unambiguous image feature extraction. For example, gray-scale thresholding can be done using a black background and white target or vice versa. Thresholding is used to construct edges and corners.

Images can also be extracted using a small size window. These were called region-based trackers. Off-line, a region in the entire image is defined. This template is searched for and tracked in the image acquisition process. The best window is found using sum-squared-difference (SSD) minimization.

To minimize the search effort, it is important to reduce the image dimensions. Using this method, the computation effort is reduced. Gaussian and Laplacian Pyramid techniques were used to achieve this goal [3]. This technique is used to search in the reduced image space.

2.1.3 General Control Approaches

In visual servoing, the task is to control the *pose* of the robot's end effector using visual information (*features* extracted from the image). This information is often used in a closed loop control scheme. Camera-to-target distance is often referred to as *depth* or *range* [7]. In 1980, Sanderson and Weiss introduced a taxonomy of visual servo systems. There are four major categories. If there is a hierarchical control architecture that uses vision to provide inputs to the joint controllers, thus making use of feedback to stabilize

the robot, then it is referred to as *dynamic look-and-move* system. On the contrary, *direct visual servoing* entirely eliminates the robot controller and replaces it with a visual servo controller, which computes the joint inputs directly and uses vision alone to stabilize the mechanism. For several reasons, which will be described in the following, many systems adopt the first approach.

First, vision systems offer low sampling rates. This makes direct control of a robot with complex dynamics a challenging control problem. Second, usually systems that accept Cartesian velocity commands are used. Third, the look-and-move approach separates the visual controller from singularities of the mechanism. This allows the robot to be considered as a Cartesian device. Another major classification is position and image based control. In *position-based* control, the scene features are extracted from the image. They are then used in conjunction with a camera and target model which is *a prior* known. The information is used to determine the pose of the target relative to the camera. In *image-based* servoing, control values are computed on the basis of the target's image features directly. The major disadvantage of position-based methods is that its success depends on camera calibrations. Image based methods are less sensitive to such calibrations. They have become the preferred method for visual servoing.

In 1993, Papanikolopoulos used sum-squared-differences (SSD) optic flow to compute the vector of discrete displacements at each time instant [26]. This approach was used to design and contrast PI, pole assignment, and LQG controllers, as well as steady-state Kalman filters.

In 1996, Corke and Good [6] made a clear distinction between visual kinematic and visual dynamic control. Kinematic control is concerned with how the manipulator should move in response to perceived visual features. The latter is concerned with manipulator dynamics and image processing delays. Corke and Good applied *feedforward control* to improve the visually servoed control. Based on the observation that vision introduces a one step delay and by taking advantage of the possibility of target prediction in the image

plane, feedforward control improves tracking performance.

In 2001, Oh and Allen [25] designed a unique controller that they called *partitioning* that exploits any fast bandwidth degrees-of-freedom in the manipulator to improve tracking performance. Frequency domain design is used to visually or kinematically servo fast and slow degrees-of-freedom respectively. The net result is a multi-input-multi-output control scheme that defines an underlying joint coupling that coordinates camera motions. The net result is that rapidly moving targets can be tracked.

2.2 Image-Based Visually-Servoed Tracking

As mentioned in the previous section image-based methods are preferred for visually-servoed robots and used in many eye-in-hand systems [8], [10] [20]. Camera pose is robotically servoed to maintain its field of view centered on the target and at a fixed distance away from the target. This is associated with what literature calls the *pose estimation* problem.

Pose regulation works as follows: first, a reference image of the target is defined as a template of the desired camera-to-target pose. During tracking, each image frame from the video stream is compared to the reference image and an error is determined. This error is then mapped from the camera's image space to the robot's task space. This will be the input to a control scheme. The control prescribes the necessary camera rotations to maintain the pose. The manipulator Jacobian transforms these camera pose motion commands into robot joint commands. The image-to-task space mapping is ensured by the *image Jacobian*.

If target trajectory is *a prior* known, the optic flow can be used for Jacobian estimation. This approach was used by Allen [1]. A fixed camera setup was used to guide a robotic arm to pick up a moving object. A circular motion of a model train was characterized by an elliptical optic flow. This was used to predict the target position.

A deterministic target behavior associated with image understanding may lead to a very good vision system. Such a system is described in [28]. Their system even worked when there was no visual contact, because the system could predict the ball position.

2.2.1 Human Operated Systems

The technological explosion of the twentieth century has produced an endless variety of machines. There are a lot of tools and systems operated by humans. Examples includes cars, trains, airplanes, PCs, as well as industrial machines. There are tools for people to use in manufacturing, transporting, building, farming, exploring and computing. As man has demanded greater performance from machines, they have demanded more of him. Higher speed, closer tolerance, more complex dynamics as well as integration and interaction with other systems and machines require of the operator predictable, repeatable, on-time performance. The design of such machines and systems that make full use of human capabilities without demanding too much is an engineering problem whose solution requires understanding of how people behave in the normally well-defined tasks required for operation and control.

In the 1950s Paul Fitts conducted experiments with various taping tasks applying principles of information theory to human movement [13]. He derived what is called Fitts law which is a robust model of human psychomotor behavior. Fitts law enables prediction of human motion based on rapid, aimed ballistic movement times.

Canon [4] developed a target-threshold control theory model for predicting human-machine movement time. This theory allows the previously empirical parameters of Fitts' speed and accuracy law to be determined before system construction. Fitts' law may now serve a role as a predictive design engineering tool for new systems. This model successfully characterized human control movement times, before system construction in experiments involving camera pointing for a new class of point-and-direct telerobotics.

Radix, Robinson and Nurse present a generic man-machine interface performance model based on a combination of Fitts law an classical control theory [27]. This paper presents a model of motion time derived from classical control system theory, using the target-threshold method derived by Canon [4] to account for gain effects.

Sheridan and Ferrell [30] described five categories of systems in the case of human

operation. A *compensatory* system is one in which the human operator has a single input, the error, the difference between the actual response y and ideal response (the reference input). A *pursuit* system is one in which the instantaneous reference input r and instantaneous controlled process output y are both displayed to the human operator separately and independently. The operator may distinguish individual properties of these signals by direct observation. Here, the human controller must be considered as a function of two input variables. A *preview* system is similar to a pursuit system except that the operator has available a true display of $r(t)$ from the present time until some time in the future. Preview control is more characteristic of everyday tasks than pursuit or compensatory control. A *precognitive* system is one in which the operator has foreknowledge of the input in terms other than a direct and true view.

In deriving a model for the operator, Sheridan and Ferrell suggest three parameters. The first one is called *reaction-time delay*. Simple reaction-time experiments reveal a minimum *reaction-time* delay (or *refractory period*) t_r of about $0.15sec$. This includes neural synaptic delays, nerve conduction and central processing time as well as the time necessary to make a measurable response. The second parameter is the *gain*. Any feedback control loop would have a gain K as large as possible consistent with stability. The gain is dimensionless and usually varies between 2 and 20 at low frequencies. The third parameter is *neuro-muscular lag*. Once a muscle is commanded to move, the muscle's inherent inertia combined with the asynchrony of muscle fiber might be expected to result in exponential response. Combining these parameters, a transfer function model of the operator is

$$Y_H(s) = \frac{K \cdot e^{-s \cdot t_r}}{1 + s \cdot t_n} \tag{2.1}$$

The teleoperation literature, however, has a long history and formulating an analytical framework for visual servoing is promising. Continuous teleoperation in earth orbit or deep in the ocean by operators on earth surface is impeded by transmission delays. These delays appear due to the limits of the light speed, computer processing at the sending and

receiving information as well as satellite relay stations. In case of vehicles in low earth orbit, the time from sending a discrete signal until the receipt of any feedback are minimally $0.4sec$ (Sheridan [31]). A similar problem appears in the remote control of vehicles deep in the ocean. This delay affects the stability of the system.

Ferrell [11] showed that the operator can adapt to avoid instability. He makes a discrete control movement then stops, waiting for confirmation that the control loop action has been followed by the remote vehicle. His experiments also showed that teleoperation task performance is a predictable function of the delay.

2.3 Applications to Our Vision System

The systems described in the beginning of this chapter have advanced the synthesis and understanding of visual servoing. Still, tracking geometrically complex targets (like people) with non-deterministic motion trajectories remains an open problem. If the camera is not mounted on a system operated by humans, its motion is not precise.

3: Preliminary Experiments

A couple of experiments were performed to validate the first of our hypotheses (section 1.3). A simple proportional controller was implemented. Using color tracking techniques, the system was used to track objects and people. In the following, a brief description of the color tracker is provided. A brief description of the proportional controller as well as conclusions follows. In an attempt to improve the system performance and stability a coupling controller was designed. Experiments with this algorithm are described as well.

3.1 The Cognachrome 2000 Color Tracker

The Cognachrome 2000 (see Figure 1.4) color vision system is a video-processing computer system. Its hardware allows multiple object tracking at 60 Hz frame rate. The resolution is 200 x 250 pixels. It is also capable of low-resolution 24-bit RGB frame grabs at 64 x 48 pixels. The input is a standard National Television System Committee (NTSC) signal. This way, real-time image processing is ensured.

The Cognachrome 2000 system allows for several modes of use:

1) Using pre-programmed algorithms, it can be set up to output tracking data over serial ports to another computer. There are two software packages available for this mode - *stand alone* and *Pioneer*.

2) The vision board itself can be programmed in C to control various actuators. This code can run as a completely separate code or can be called after each frame. The board has free input ports which may be used for interfacing to other devices.

3) The system allows one to compile and place their own processing algorithms on the board as well as make data available to a remote machine or to other processes operating on the board.

There are two boards. The top one is the color processing board which digitizes the incoming NTSC into 24 bit RGB. This RGB signal is then sent to through a look-up table.

This board can be trained to recognize up to three color at once. Objects are tracked or recognized using the color look-up table. This runs at 60 Hz and allows multiple object tracking at once.

The bottom board contains the processor. It is a standard Motorola 68332-based computer with 256K of RAM. This quantity is expandable to 1MB on board. The board has input/output ports left over after the interface to the vision processing board. This includes digital input/output lines, a bus with software-definable chip selects and two asynchronous and one synchronous serial ports. It is therefore possible to directly control a robot or to interface to a network computer.

There are two serial ports. Port A is considered to be the primary port, as it is connected directly to the 68832 processor's hardware asynchronous serial port, and thus takes minimal overhead for data transmission and reception. Port B is connected to the general-purpose Timer Processor unit. This runs microcode to implement asynchronous serial protocol. Port B is only reliable at 9600 baud or slower. The board can be programmed to automatically send the image centroid data every frame (60 Hz is about 16.66 ms) or the computer can periodically "ask" for data. The communication uses an RS232 protocol and data is sent as a string of characters. This string of characters is processed in our program for conversion into decimal numbers.

The board has a vision output as well. This can be connected to a video monitor. The output signal is always black and white so no color monitor is needed.

3.2 The Pan-Tilt Unit

The boom-camera system uses a commercially-available, motorized pan-tilt unit. This is actuated by two DC motors to rotate the camera with respect to a horizontal and a vertical axis. The characteristics of the two DC motors can be seen in Table 4.1.
They are equipped with digital encoders which offer position information. The power electronics are driven by an Industry Standard Architecture (ISA) Digital Signal Processing

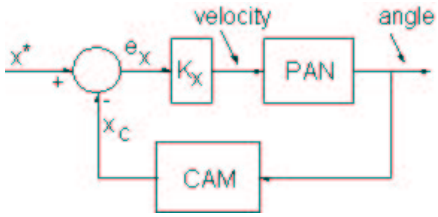

Figure 3.1: The schematic block diagram for a simple proportional controller. Tracking with such a controller is successful.

(DSP) board which ensures PC interface. The DSP board is accompanied by a set of libraries available to the developer. Using functions from these libraries, a C program can rotate the camera by prescribing the position or the velocity. Time, current position and velocity information are also available.

3.3 A Simple Proportional Controller

As stated before, the first hypothesis was that using a simple proportional controller and simple image processing technique, we can track moving targets even while the boom moves. We first implemented a simple proportional controller.

The controller schematic diagram can be seen in Figure 3.1. The actual position of the target is compared with the desired one. The result is sent to the controller which computes the desired camera velocity. This value is then sent to the driver unit.

3.4 Experiments Description

Several experiments were performed in the lab. The goal of these experiments was to confirm (or infirm) our first hypothesis. The first of our experiments was to track a toy-truck (see Figure 3.2). An artificial white background was used to help the system to better detect the target. The camera-target distance was 3 m. The toy was moving back and forth while the camera was tracking. Camera motion data, booming data and tracking error were

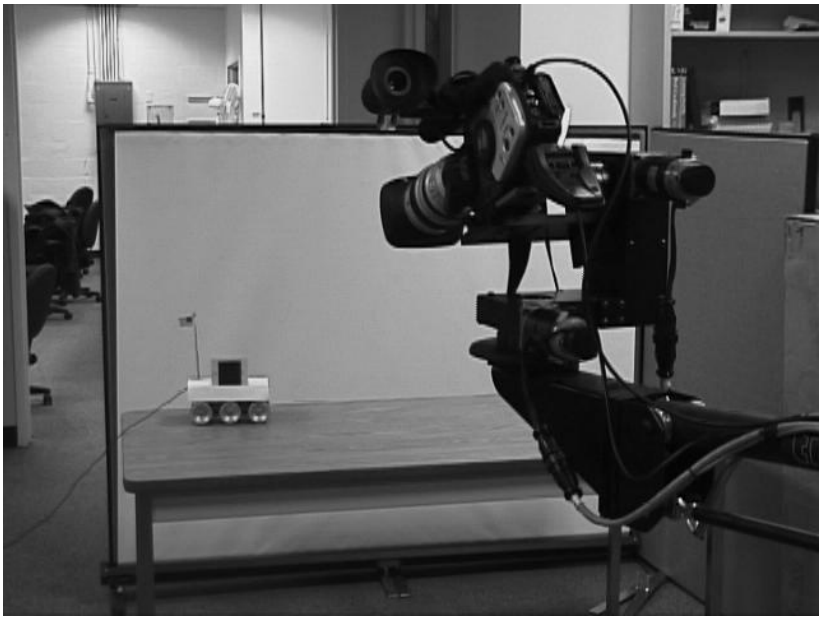

Figure 3.2: Tracking a toy truck. The truck is moving back and forth in an “artificial” environment. The proportional controller is used to track this target while an operator is booming. Tracking error, booming and camera data are recorded.

recorded. The plots can be seen in Figure 3.3.

It can be seen that as the operator is booming and the target is moving the controller performs a visually servoed counterrotation. The system is able to track the moving target. Still there were two challenges: system stability and tracking performance.

The work demonstrated that the key design parameter when visually servoing redundant degrees-of-freedom systems is stability, especially when the target and the boom move 180 degrees out of phase. If boom motion data is not included, camera pose cannot be determined explicitly because there are redundant degrees-of-freedom. As a result, the system could track a slow moving target rather well, but would be unstable when the target or boom moves quickly.

The second issue was the system performance. The operator was booming very slowly (less the 1 deg/sec). The target moved slow as well (about 10 cm/sec). Any attempt to increase the booming or target speed resulted in tracking failure.

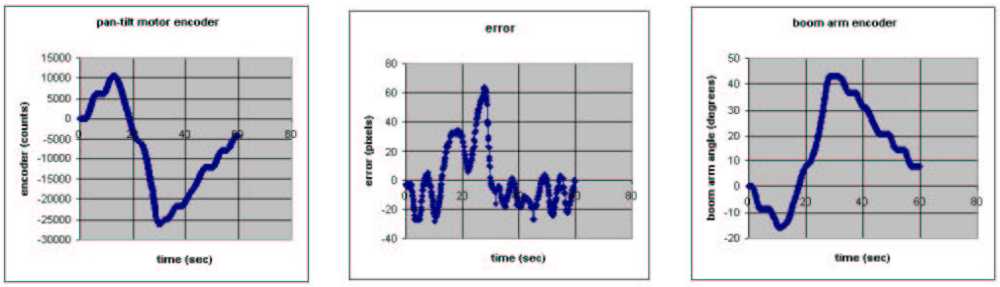

Figure 3.3: Experiment with the Proportional Controller $K_x = 100$, a) encoder b)pixel-error c)boom-arm encoder.

Still, another experiment was performed. This time people tracking was attempted. A person wearing a red coat was asked to move back and forth in the lab. The color-tracker was trained for red. The task was to keep the red coat in the camera's field of view while an operator boomed. The camera-target distance was about $5m$. Sequential images can be seen in Figure 3.4.

The boom camera's point of view can be seen in the top row. In the middle row, the three images show the operator booming and the target moving. It can be seen that both the target and the boom are moving. The bottom row shows a view taken from another camera.

As the images show, the system is able to perform people tracking. Still, the same challenges as in the previous experiment apply. It is important to underline that the vision had no information about booming. So, introducing booming information can improve tracking performance as well as stability.

3.5 The Coupling Controller

As can be seen, the system is able to track using a very simple proportional controller. Many times it almost lost the target. Stability was also an issue. To sidestep the challenges previously mentioned, a better controller is needed. More than that, as stated before, the vision system has no booming information. By introducing this information

Figure 3.4: Three sequential images from videotaping the experiment. Top row: camera field-of-view shows target is tracked. Middle row: boom manually controlled. Bottom row: view from another camcorder.

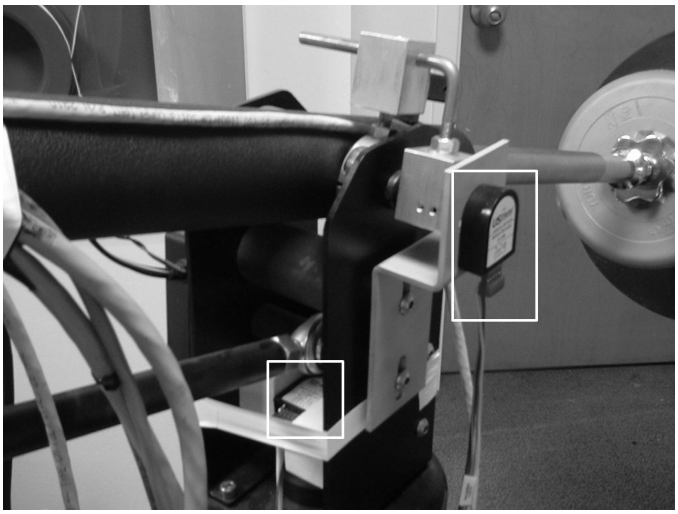

Figure 3.5: Two US Digital encoders were mounted on the boom to determine the angle of rotation in the horizontal and vertical plane. An ISA PC board is used to read them.

both the performance and the stability were expected to increase.

Two encoders were mounted on the boom. A US Digital ISA board is used to read these encoders (Figure 3.5). The booming velocity is determined by derivation followed by filtering.

Inspired by the work of Oh and Allen [25] and dividing the system into two parts - the human-in-the-loop and the pan-tilt unit - serves to overcome instability challenges and lay the foundation for designing a couple algorithm. By taking advantage of the fact that the motion of the human-in-the-loop is much slower than the motion of the camera-pan-tilt-head, this algorithm is able to counterrotate the camera and keep a stationary target in the camera's field of view when the operator booms. The vision system compensates only for target motion.

A schematic depicting the top view of the boom is shown in Figure 3.6. Assuming a stationary target, one can calculate the camera velocity needed to compensate for boom rotation. For example, in Figure 3.6, one assumed that the motions start when the boom, camera and target are aligned. A "worst case" scenario is presented here: the boom rotates

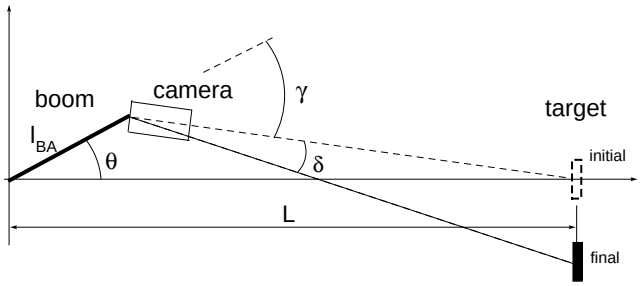

Figure 3.6: Boom top view; due to the coupling algorithm, the vision system has to compensate only for angle δ.

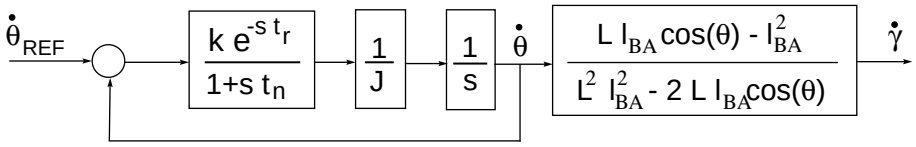

Figure 3.7: The coupling block diagram. Based on the boom's angular velocity, the camera velocity is calculated.

towards left while the target moves towards right. The program compensates for the angle γ while the vision system will compensate only for angle δ. It can be seen that the angle which is to be compensated for by the vision system is reduced. Here, L is the distance between the scene and the boom's pivot, l_{BA} is the length of the boom and γ is the angle the pan-tilt head rotates with respect to the boom. Assuming that the operator will boom such that Equation (3.1) is satisfied, then the value of the pan-tilt head velocity which will compensate for booming is given by

$$\dot{\gamma}(t) = \left(\frac{Ll_{BA}\cos(\theta) - l_{BA}^2}{L^2 + l_{BA}^2 - 2Ll_{BA}\cos(\theta)} + 1\right)\dot{\theta}(t) \tag{3.1}$$

where θ is the angle of the boom with respect to its initial position. A block diagram of this

is shown in Figure 3.7. In this figure, the first block represents the operator transfer function as it was derived by Sherridan and Ferrell (Section 2.2.1 equation 2.1). The second block contains the transfer function of the boom. Assuming a rigid, frictionless boom structure, the boom transfer function relating the torque acting on boom and its angular acceleration is given to be

$$G_{BA} = \frac{\ddot{\theta}(s)}{M(s)} = \frac{1}{J} \tag{3.2}$$

The angular velocity is obtained by integration. The input is the boom's angular velocity and the output is the camera angular velocity. Figure 3.7 represents the coupling controller. The output represents the desired camera velocity. This value is sent to the PTU electronic driver as reference speed.

3.5.1 Experiments with the Coupling Controller

The coupling controller was implemented in C. A set of experiments was performed to assess its performance. As these experiments only attempt to test the coupling controller algorithm, this time a fixed target was used. In these experiments the target was 4.5 *m* away from the boom's pivot.

The operator boomed while the controller attempted to keep the target in camera's field of view. Booming data as well as the error in both the *x* and *y* directions were recorded. Some results can be seen in Figure 3.8. It can be seen that as the operator boomed, the coupling controller attempted to counterrotate the camera. The plots of the error in *x* and *y* directions can be seen as well. The plots show successful tracking. Still, this algorithm depends on camera-target distance. When this value is not precisely known, the errors are not acceptable.

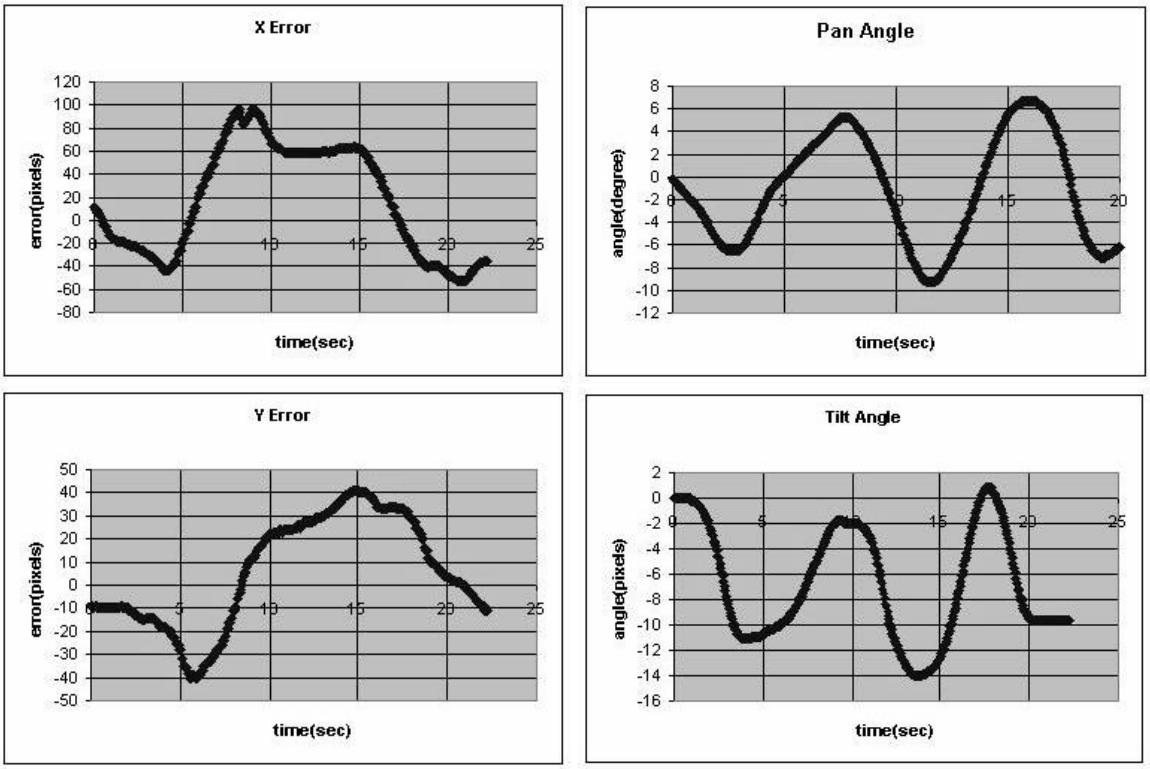

Figure 3.8: Coupling Algorithm Experiment Results: Pan and Tilt errors and angles. The errors are still too big.

4: The Feedforward Controller

The term *visual servoing* has somehow replaced the term *visual feedback*. Almost all reported visual servoed systems are based on a feedback control scheme. In this chapter, a feedforward controller with a feedback compensator is described.

The coupling algorithm we presented above had several problems. The most inconvenient one was that the camera-to-target distance has to be known precisely. If any error appears, this induces errors in our algorithm. To sidestep this, a feedforward control scheme with a fedback compensator was adopted. In the following, for simplicity, this control scheme is referred to as feedforward controller.

4.1 Visual Feedforward Control

To overcome instabilities and increase the performance, a feedforward controller was designed. This provides target motion estimation [6]. The schematic block diagram is shown in Figure 4.1. The transfer function can be written as

$$\frac{{}^{i}X(z)}{X_t(z)} = \frac{V(z)(1 - G_p(z) \cdot D_F(z))}{1 + V(z) \cdot G_p(z) \cdot D(z)} \tag{4.1}$$

where ${}^{i}X(z)$ is the position of the target in the image, $X_t(z)$ is the target actual position, and $V(z)$ and $G_p(z)$, are respectively the transfer functions for the vision system and PTU. $D_F(z)$ and $D(z)$ are respectively the transfer functions for the feedforward and feedback controllers.

Clearly, if $D_F(z) = G_p^{-1}(z)$ the tracking error will be zero, but this requires knowledge of the target position which is not directly measurable. Consequently, the target position and velocity are estimated. For a horizontally translating target, its centroid in the image plane is given by the relative angle between the camera and the target.

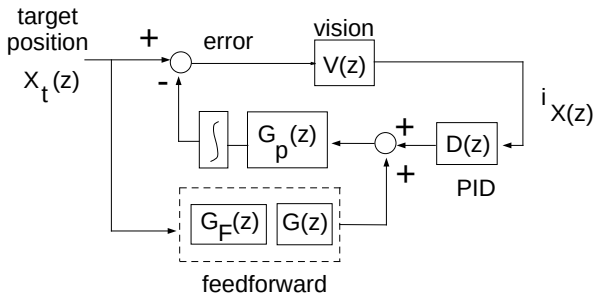

Figure 4.1: The Feedforward Controller with Feedback Compensation. This bloc diagram is not usable in this form.

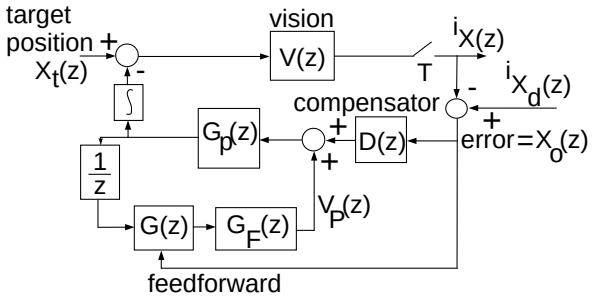

Figure 4.2: The Feedforward Controller with Feedback Compensation as it was implemented.

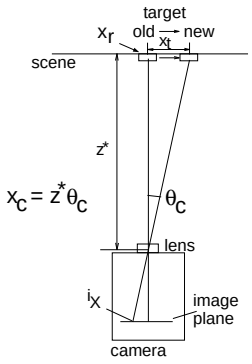

Figure 4.3: A schematic of camera-scene. Using the information coming from encoders the controller can estimate the target's position.

4.2 Target Position Estimation

Without having target motion information, the schematic block diagram depicted in Figure 4.1 cannot be used. Still, based on the successive target positions, its position and velocity can be estimated (Figure 4.2). The camera senses target position error directly:

$$^{i}X(z) = K_{lens}(X_t(z) - X_r(z)) \tag{4.2}$$

where $^{i}X(z)$ and $X_t(z)$ are the target position in the image plane and world frame respectively. $X_r(z)$ is the position of the point which is in camera's focus (due to the booming and camera rotation) and K_{lens} is the lens zoom value. The target position prediction can be

obtained from the boom and PTU as seen in Figure 4.3. Rearranging this equation yields

$$\hat{X}_t(z) = \frac{^{i}\tilde{X}(z)}{K_{lens}} + X_r(z) \tag{4.3}$$

where $\hat{X}_t$ is predicted target position.

It is known that the vision system introduces a unit delay, so the algorithm should be delayed. This will lead to a delay in target position estimation. The estimates must be

based on some assumption of target motion. Tracking performance will be evaluated for a standard sinusoidal test motion.

4.2.1 Simple Differentiation

A way to determine the target's velocity is simply to differentiate its position. However, this method requires a filter to eliminate the introduced noise. Also, it is possible to compute the optical flow rather than differentiate the position with respect to time. In literature it is suggested that the human eye's fixation reflex is driven by retinal slip or optical flow.

4.2.2 The α - β tracking filter

Numerous applications such as air-traffic handling, missile interception and anti-submarine warfare require the use of discrete-time data to predict the kinematics of a moving object. The use of passive sonobuoys which have limited power capacity constrain us to implement target-trackers which are computationally inexpensive. With these considerations in mind, an analysis of an α - β filter is presented in this paragraph.

α - β filters were developed in the mid 1950s [32] for radar target tracking, position and velocity estimation from noisy measurements of range and bearing. In the simplest form they are fixed gain filters based on the assumption that the target acceleration is small enough (or even zero).

The target is periodically observed and its position is recorded. Based on the recorded position, an estimated position is made by the filter. These algorithms work in two steps. The first step is *prediction*, in which the position and the velocity of the target is predicted for the new iteration:

$$x_p(k+1) = x_s(k) + Tv_s(k) \tag{4.4}$$

where T is the sample time and $x_p(k+1)$ is the prediction for position at iteration $k+1$.

$x_s(k)$ and $v_s(s)$ are the corrected values of iteration k for position and velocity respectively.

The second step is to make corrections

$$x_s(k) = x_p(k) + \alpha(x_o(k) - x_p(k)) \tag{4.5}$$

$$v_s(k) = v_s(k) + (\beta/T)(x_o(k) - x_p(k)) \tag{4.6}$$

where $x_o(k)$ is the observed (sampled) position at iteration k. The above equations can be arranged into a transfer function form

$$\frac{\hat{V}_s(k)}{X_o(z)} = \frac{\beta}{T}\frac{z(z-1)}{z^2 + (\alpha+\beta-2)z + 1 - \alpha} \tag{4.7}$$

It is then clear that the poles are manipulated by the values of α and β. The filter is treated as having only one free parameter (α) while β is computed for critical damping.

$$\beta_{CD} = 2 - \alpha - 2\sqrt{1-\alpha} \tag{4.8}$$

4.2.3 The α - β - γ tracking filter

The α - β - γ filter is a higher order extension. This algorithm also estimates acceleration and should improve target tracking performance. The filter works similar to the previous one. Its equations are

$$x_p(k+1) = x_s(k) + Tv_s(k) + T^2 a_s(k)/2 \tag{4.9}$$

$$v_p(k+1) = v_s(k) + Ta_s(k) \tag{4.10}$$

where T is the sample time and $x_p(k+1)$ and $v_p(k+1)$ are respectively the predictions for position and velocity at iteration $k+1$. $x_s(k)$, $v_s(k)$ and $a_s(s)$ are the corrected values at iteration k for position, velocity and acceleration respectively.

The second step is to make corrections

$$x_s(k) = x_p(k) + \alpha(x_o(k) - x_p(k)) \tag{4.11}$$

$$v_s(k) = v_p(k) + (\beta/T)(x_o(k) - x_p(k)) \tag{4.12}$$

$$a_s(k) = a_s(k-1) + (\gamma/2T^2)(x_o(k) - x_p(k)) \tag{4.13}$$

where $x_o(k)$ is the observed (sampled) position at iteration k. The appropriate selection of gains α, β and γ will determine the performance and stability of the filter [36].

4.2.4 Kalman filter

This filter, proposed by Kalman in 1960 [19], is an optimal estimator of system state where input and the measurement noise are zero-mean Gaussian signals and the covariance is known.

Literature shows that a Kalman filter was oftentimes used. Still, this algorithm is relativelly complex and time consuming to execute. However, its gains converge to stable values.

4.3 The Pan-Tilt Unit Transfer Function

As mentioned before the camera is mounted on a 2 degree-of-freedom pan-tilt unit (PTU) which, is actuated by two electric motors. Two DC motors are driven by a motion card installed in a PC. Like many commercial motion cards, the PID control gains are factory set, balancing transient response with minimal overshoot. Using a standard DC motor transfer function, one has

$$\begin{aligned} G_m(s) &= \frac{\dot{\theta}_m(s)}{E_a(s)} \\ &= \frac{K_t}{K_v K_t + (sJ_m + D_m)(R_a + sL_a)} \end{aligned} \tag{4.14}$$

where $\dot{\theta}_m$ is motor speed, E_a is the applied voltage, K_t is the motor torque constant, K_v is

Table 4.1: The parameters of the PTU motors.

Motor Parameters	Value and Units
R_a, rotor resistance	1.15Ω
L_a, rotor inductance	$1.4mH$
K_t, torque constant	$0.055Nm/A$
K_v, back EMF constant	$5.8V/krpm$
J_a rotor moment of inertia	$1.33 \cdot 10^{-5}$

the back EMF constant, R_a and L_a are the rotor resistance and inductance respectively and D_m is the armature viscous damping. Values for these parameters are given in Table 4.1. J_m is the motor shaft's moment of inertia.

$$J_m = J_a + J_L(\frac{N_1}{N_2})^2 \tag{4.15}$$

where, J_L is load moment of inertia, J_a is the rotor moment of inertia and $\frac{N_1}{N_2}$ is the gear ratio. The PTU's gear ratio and D_m are both small and were set to zero. As such, Equation(4.15) with values from Table 4.1 results in

$$G_m(s) = \frac{\dot{\theta}_m(s)}{E_a(s)} = \frac{5500}{0.001862s^2 + 1.295s + 31.9} \tag{4.16}$$

Using a zero-order-hold to model a digital-to-analog converter, the discrete form of the transfer function can be calculated. Figure 4.4 gives the plant block diagram that combines the motor and the PID controller $D(z)$. Here v_{ref} is the command reference velocity, E is the error between the command and actual motor velocities and $K_e = 2000$ counts/rev is the encoder constant. The sampling time T was set at 1.25 msec. $D(z)$ is the factory tuned PID controller with proportional, integral and derivative gains set at $K_P = 15000$, $K_I = 40$ and $K_D = 20000$ respectively for the PTU pan motor. PID gains for the tilt motor were factory set at $K_P = 15000$, $K_I = 20$ and $K_D = 32000$. With $G_m(s)$ given by

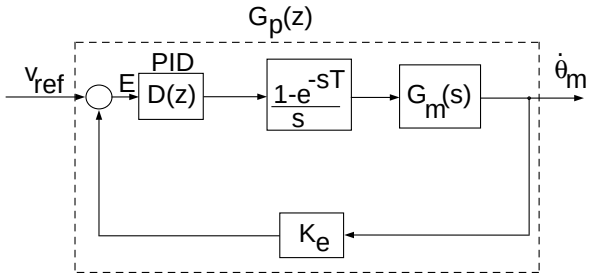

Figure 4.4: The PTU Controller Block Diagram. This represents the controller of each degree-of-freedom of camera.

Equation 4.16, the plant discrete transfer function $G_P(z)$ relating the command and actual velocities is given as

$$G_p(z) = \frac{0.704 - 0.787z^{-1} + 0.439z^{-2} - 0.055z^{-3} + 0.035z^{-4}}{1409.3 - 1575.36z^{-1} + 878z^{-2} - 11.02z^{-3} + 70z^{-4}} \tag{4.17}$$

4.4 Feedforward Controller Implementation

To implement the controller, a method for target position and velocity had to be chosen. To combine the advantages of simplicity of implementation as well as tracking performance the α - β - γ filter was chosen. The next section discusses the filter stability.

4.4.1 Stability of α - β - γ filter

As mentioned previously, the stability and performance of this filter are determined by the values of α, β and γ. Prior to implementation, a stability study was done.

By applying the Z-transform to equations 4.9 and 4.10,

$$zX_p(z) - zx_p(0) = X_s(z) + TV_z + \frac{1}{2}T^2A_s(z) \tag{4.18}$$

$$zV_p(z) - zv_p(0) = V_s(z) + TA_s(z) \tag{4.19}$$

By applying the Z-transform to equations 4.11, 4.12, and 4.13,

$$X_s(z) = X_p(z) + \alpha(X_o(z) - X_p(z)) \tag{4.20}$$

$$V_s(z) = V_p(z) + (\beta/T)(X_o(z) - X_p(z)) \tag{4.21}$$

$$A_s(z) = \frac{z}{z-1}(\gamma/2T^2)(X_o(z) - X_p(z)) \tag{4.22}$$

Therefore,

$$zX_p(z) = zx_p(0) + X_p(z) + \alpha[X_o(z) - X_p(z)] + T[V_p + \frac{\beta}{T}[X_o(z) - X_p(z)] + \frac{1}{4}\frac{z}{z-1}\gamma[X_o(z) - X_p(z)] \tag{4.23}$$

Similarly,

$$(z-1)V_p(z) = zv_p(0) + \frac{1}{T}[X_o(z) - X_p(z)][\beta + \frac{\gamma}{2}\frac{z}{z-1}] \tag{4.24}$$

Assuming $v_p(0) = 0$, Equation 4.24 becomes

$$V_p(z) = \frac{1}{T}[X_o(z) - X_p(z)][\frac{2\beta(z-1) + \gamma z}{2(z-1)^2}] \tag{4.25}$$

If $V_p z$ is substituted, the 4.23 becomes

$$(z-1)X_p(z) = zx_p(0) + [X_o(z) - X_p(z)][\alpha + \frac{2\beta(z-1) + \gamma z}{2(z-1)^2} + \beta + \frac{1}{4}\frac{z}{z-1}\gamma] \tag{4.26}$$

Assuming $x_p(0) = 0$ this time, we obtain an equation which depends only on $X_p(z)$, $X_o(z)$ and has parameters α, β and γ. The transfer function relating $X_p(z)$ and $X_o(z)$ is then

$$G_{P\alpha\beta\gamma} = \frac{X_p(z)}{X_o(z)} = \frac{(\alpha + \beta + \frac{\gamma}{4})z^2 + (-2\alpha - \beta + \frac{\gamma}{4})z + \alpha}{z^3 + (\alpha + \beta + \frac{\gamma}{4} - 3)z^2 + (-2\alpha - \beta + \frac{\gamma}{4} + 3)z + \alpha} \tag{4.27}$$

Table 4.2: The Jury's Stability Table of the α - β - γ filter.

z^0	z^1	z^2	z^3
$\alpha-1$	$-2\alpha+\beta+\frac{\gamma}{4}+3$	$\alpha+\beta+\frac{\gamma}{4}-3$	1
1	$\alpha+\beta+\frac{\gamma}{4}-3$	$-2\alpha+\beta+\frac{\gamma}{4}+3$	$\alpha-1$
$\alpha(\alpha-2)$	$\alpha(4-2\alpha-\beta+\frac{\gamma}{4})-\frac{1}{2}$	$\alpha(\alpha+\beta-2+\frac{\gamma}{4})-\frac{1}{2}$	

where "P" stands for position. A similar transfer function which relates the predicted velocity to the observed position can be derived as well. Jury's stability test is used to determine the stability region. Writing the coefficients of the characteristic polynomial in Jury's table and calculating the determinants yields Table 4.2.

The condition $a_0 > 0$ is satisfied since $a_0 = 1$. To satisfy the constraints $|a_n| < a_0$ the coefficients require $|\alpha - 1| < 1$, which is equivalent to

$$0 < \alpha < 2 \tag{4.28}$$

Substituting $z = 1$ and applying the constraint $P(z)|_{z=1} > 0$ requires satisfaction of the inequality

$$1 + (\alpha + \beta + \frac{\gamma}{4} - 3) + (-2\alpha - \beta + \frac{\gamma}{4} + 3) + \alpha - 1 > 0 \tag{4.29}$$

which can be rewritten as

$$\gamma > 0 \tag{4.30}$$

Satisfying the constraint $P(z)|_{z=-1} < 0$ for odd n yields

$$2\alpha + \beta < 4 \tag{4.31}$$

which is the same constraint for α and β as for the $\alpha-\beta$ tracker. The final condition $|b_2| > |b_0|$ requires

$$|\alpha(\alpha-2)| > |\alpha(\alpha-2)+\alpha(\beta+\frac{\gamma}{4})-\frac{\gamma}{2}| \tag{4.32}$$

Observing that the term $\alpha(\alpha-2)$ is always negative, we have

$$\alpha(\beta+\frac{\gamma}{4})-\frac{\gamma}{2} > 0 \tag{4.33}$$

This statement leads to the constraint on γ for which the $\alpha-\beta-\gamma$ tracker is stable, which is

$$\gamma < \frac{4\alpha\beta}{2-\alpha} \tag{4.34}$$

The adopted values were then: $\alpha = 0.75$, $\beta = 0.8$ and $\gamma = 0.25$.

The $\alpha-\beta-\gamma$ filter was then implemented. To gain confidence regarding implementation, an experiment was performed. The target's actual and predicted position were recorded versus time. The plot can be seen in Figure 4.5.

Excluding the moments where the target changes direction with respect to the camera, the errors are negligible (less then 10 *pixels*).

4.4.2 The Condensation Algorithm

To detect the pre-selected target, the condensation algorithm was employed. A brief description of how this algorithm works is given in this paragraph. This algorithm is a tracking framework combining trained motion models and factored sampling to accomplish the task of propagating the state density of a tracked object over time [16].

The trained motion models are used to get the best possible prediction of the movement of the object between frames. This is essential for the performance of the tracker.

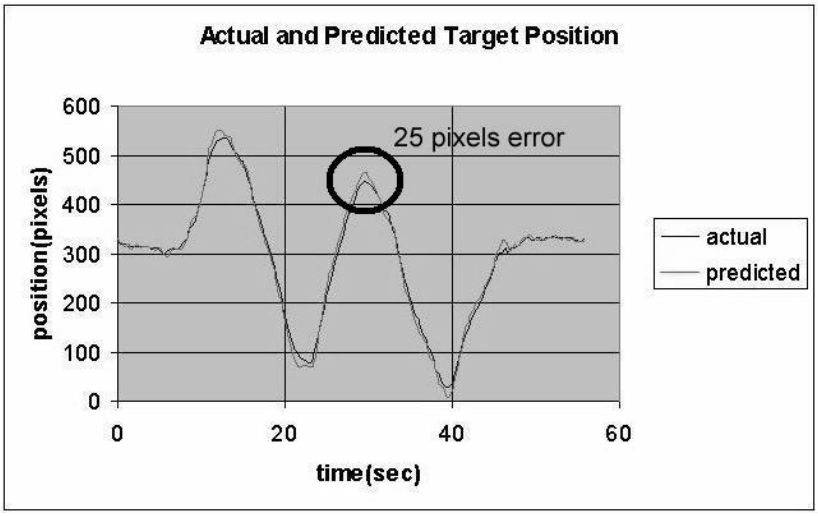

Figure 4.5: The actual and predicted target position data were recorded versus time. It can be seen that the filter predicts the targets position within ± 25 *pixels*. This maximum error occur when camera-target relative velocity changes.

The task of the state based tracker is to determine the configuration (spatial motion, possibly 3D orientation and internal degrees of freedom) of an object from consecutive frames of video. One way to accomplish this is by testing every possible configuration and then selecting the one that best fits the current frame.

Instead of comparing every possible configuration of the object with each video frame, the Condensation algorithm makes a set of informed guesses of the configuration, compare these guesses with the current frame and uses the result of this comparison as the basis for a new set of guesses when the next frame arrives. These new guesses are made by selecting the best guesses from the last frame and applying a model of the movement of the object from one frame to the next. The set of guesses will converge around the correct state of the object frame by frame.

To be able to perform tracking using this method, two models must be available: 1) A motion model encompassing knowledge of how the state of the object evolves over time. This is the basis for dynamic propagation of the guesses from one frame to the next.

In addition to a deterministic part based on mechanical properties of the object, this model generally contains a stochastic element modelling the uncertainties in the model and in the data. 2) A measurement model describing how physical properties of the object and background produce the features used for comparing particles with image data. This model

forms the basis for a likelihood function that expresses the confidence, that a particular guess corresponds to the actual state of the object tracked.

These models must be constructed based on *a priori* knowledge of the specific tracking problem. This knowledge can either be deduced from known physical properties of the tracked object, the background, and the lighting and camera setup, or it can be learned from video sequences similar to what the final system is expected to handle. The better these models are, the better the performance of the tracking system will be. The description can be divided in two parts: initialization and tracking.

Initialization is performed when the tracker is started to provide an initial estimate of the position of the target. The tracker is initialized by creating a set of guesses of the configuration of the tracked object. Once the initialization is accomplished, tracking is started.

4.5 Experiments With the Feedforward Controller

4.5.1 People Tracking Experiment

With the system equipped with the feedforward controller, a couple of experiments were performed. Again, the first was a people tracking experiment. A subject was asked to move back and fourth in the laboratory environment. While the operator boomed, the camera tracked the subject. This experiment was videotaped as well. Sequential images can be seen in Figure 4.6. The first row shows the boom camera view. It can be seen that the system tracks much better in this case in comparison to the proportional controller. The second row shows the operator booming while the third row shows the program working.

Figure 4.6: Three sequential images from videotaping the experiment using the feedforward controller. Top row: camera field-of-view shows target is tracked. Middle row: boom manually controlled. Bottom row show the working program.

Figure 4.7: Three sequential images from videotaping the Cye robot tracking experiment using the feedforward controller. Top row: camera field-of-view shows target is tracked. Middle row: boom manually controlled. Bottom row: the working program.

It can be seen that target is well detected.

4.5.2 Tracking the Cye Robot

The feedforward controller was tested to track the Cye robot which was moving on a specific, pre-programmed path. The large path was set in the lab environment such that the camera-target distance strongly varied along the path (from $1m$ to $5m$). During tracking, the operator was booming at the same time the robot was in motion. The maximum robot speed was about $0.5m/sec$.

The experiment was videotaped as well. Several pictures can be seen in Figure 4.7.

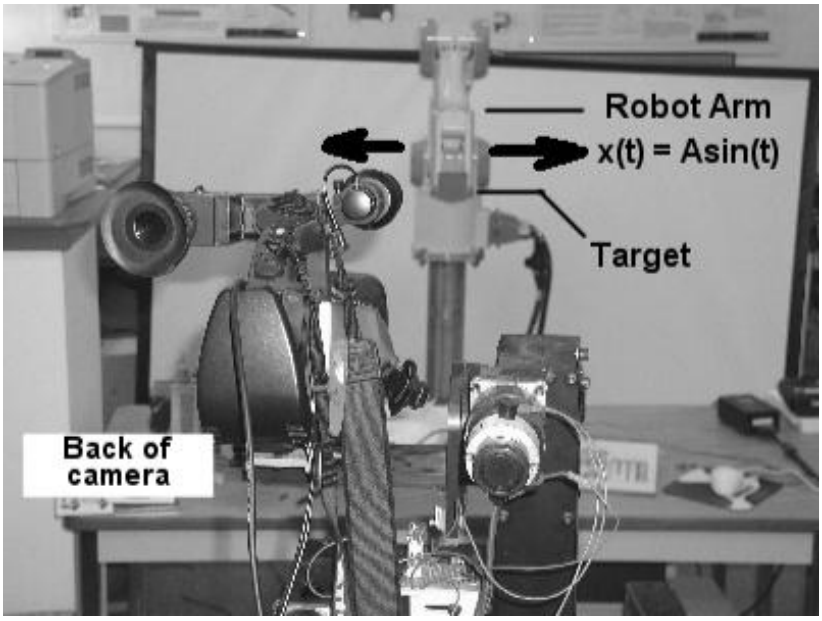

Figure 4.8: A wooden block target was mounted on the end-effector of a Mitsubishi robot arm (background). The boom-camera system (foreground) attempts to keep the target's image centered in the camera's field-of-view.

4.5.3 A Comparison between the Feedforward and the Proportional Controller

To assess the performance of the new controller, another experiment was set up. A Mitsubishi robotic arm was programmed to sinusoidally move its end effector back and forth. The setup can be seen in Figure 4.8. The target was attached to the end effector and was tracked using both the proportional and the feedforward controller. An operator was panning the boom at the same time. Data regarding Mitsubishi motion, booming and error were recorded. The performance is assessed by comparing the tracking error.

The experiment was set up in the lab. The camera-target distance was $3.15m$. The target dimensions were $8.9 \times 8.25cm^2$. The robotic arm moved the target sinusoidally with a frequency of about $0.08Hz$ and a magnitude of $0.5m$. The condensation algorithm was used for target detection. As this algorithm is noisy, the target image should be kept small. The target dimensions in the image plane were $34 \times 32 pixels$. While both controllers at-

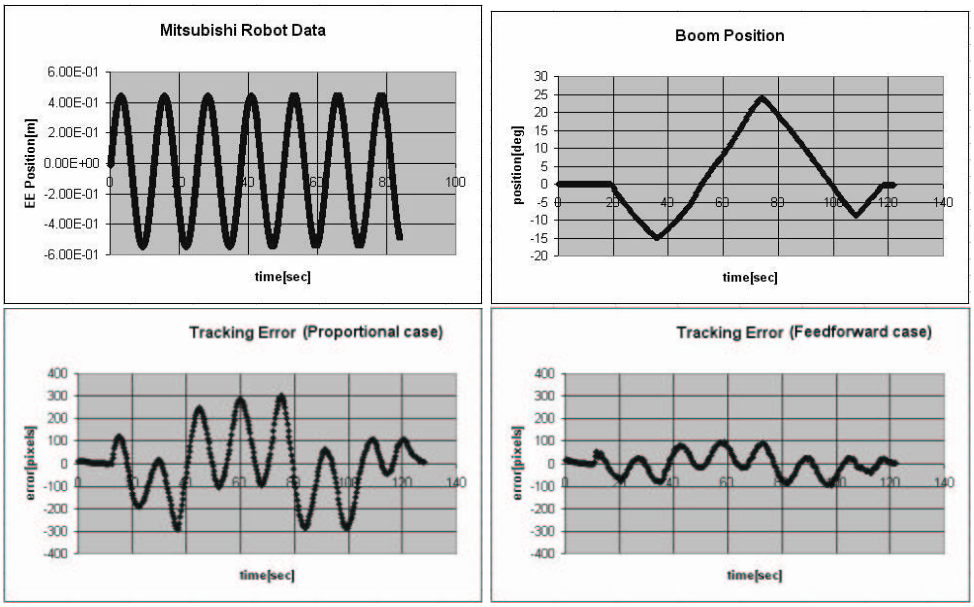

Figure 4.9: Tracking errors comparing feedforward and proportional control in *human-in-the-loop visual-servoing*. Top row: target sinusoidal motion and booming. It can be seen that the operator moved the boom real slow (about 1 deg/sec). Bottom row: the tracking error using a proportional control (left hand side) and a feedforward control (right hand side).

tempted to track, the boom was manually moved from $-15deg$ to $+25deg$. The plots can be seen in Figure 4.9.

In the top row, the target motion and the booming plot (both versus time) can be seen. The operator moved the boom really slow (approximately 1 deg/sec). This rate was used because of the proportional controller. The tracking errors are shown in the bottom row. The left hand side image (bottom row) shows the error when using the proportional controller for tracking. The right hand side image (bottom row) shows the error when using the feedforward controller. When using the feedforward controller, the peak-to-peak error was about 100 *pixels*, while with the proportional controller the error was more then 300 *pixels*. By comparing the error in the same conditions, the conclusion was that the feedforward controller is “much better” then the proportional one.

5: A Model for the Boom-Camera System

5.1 Introduction

In the previous chapter it is shown that by using a feedforward control scheme, the tracking performance as well as system stability increases.

In [33] we underlined the system's stability challenge, especially when the target and the boom move 180 degrees out of phase. If the system has no booming information, camera pose cannot be determined explicitly because there are redundant degrees-of-freedom. As a result, the system could track a slow moving target rather well, but would be unstable when the target or boom moves quickly. The feedforward controller described in [34] improved performance. In comparison with the proportional controller, the system was able to track a sinusoidally moving target with an error three times less that the proportional controller.

At this step, a model for the boom was desired. Based on the fact that, despite simplicity, the boom is nonlinear, our next hypothesis is that a better control scheme can offer even better performance. But to design and implement such a controller, a model of the boom is needed. To apply control design methods to this relatively complex model can be computationally expensive.

The first part of this chapter describes derivation of a dynamical model for the boom-camera system. The second part shows the validation of this model by matching the simulation results against experimental ones. Some conclusions are presented as well.

5.2 How to model the system

Sometimes, multibody mechanical systems have a chain structure. In such case, rigid bodies are connected together by joints. Considering dimensional relationships between links it is possible to write the kinematic equations between the two ends. Still, cases when a mechanical system do not have a chain structure exists (i.e. when they con-

tain loops). In such cases it is convenient to treat them in the same manner and add the necessary constraints (which will re-establish the loops).

The boom-camera system is composed of rigid bodies connected together by joints. Because it contains about seven links it is convenient to exploit modern computer algebra tools like *Mathematica* and *TSi ProPac* to build the model. The boom is treated as a chain structure with constraints.

There are certain ways to model a system such as the one we are dealing with. The *analytical* way uses Lagrange's mechanics. By writing the potential and kinetic energy of the system and applying Lagrange's formula, a system of equations is found. The second way is using mathematical packages such as *Mathematica* and *ProPac*. These software packages make use of Poincaré's equations. Describing the mathematical theory behind these equations is beyond the goal of this thesis. For additional readings we recommend Chapter 4 from ([22].

5.3 Creating the Symbolic Model

A nonlinear mathematical model of the boom and the corresponding simulation model were developed using *Mathematica* and *Tsi ProPac* ([22] and [21]). The mathematical model enables us to evaluate the properties of the boom and design either a linear or nonlinear controller. The simulation model is in the form of a C-code that can be complied as an S-function in SIMULINK. Together, these models of the highly involved boom dynamics facilitate the design and testing of the controller before its actual implementation.

The boom, shown in Figure 5.1, is comprised of 7 bodies and 8 joints. The bodies and joints are denoted by boxes and circles, respectively. The degrees of freedom of the various joints are detailed in Table 5.1, while the physical data are given in Table 5.2. They give the position/Euler angles of the 'joint body'(JB) with respect to the 'reference body'(RB). At the origin, which corresponds to a stable equilibrium, the boom and the camera are perfectly aligned. The characteristic of the boom to always position the camera's

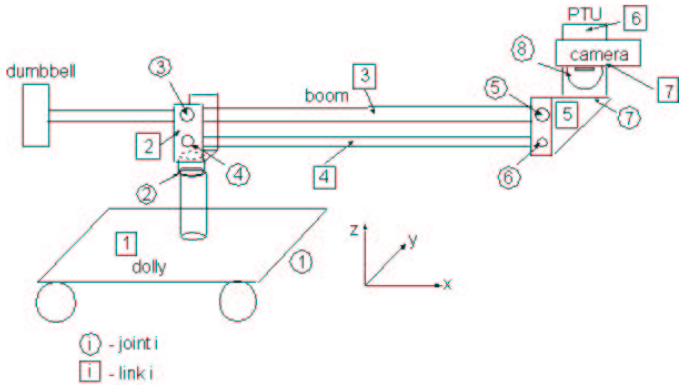

Figure 5.1: A number was assigned to every link and joint. Circled numbers represent joints while a number in a rectangle represents a link.

Table 5.1: Types of motion for links.

Joint #	RB	JB	x	y	R_x	R_y	R_z
1		1	x	y			
2	1	2					ψ_b
3	2	3				θ_{bt1}	
4	2	4				θ_{bb1}	
5	3	5				θ_{bt2}	
6	4	5				θ_{bb2}	
7	5	6					ψ_c
8	6	7				θ_c	

Table 5.2: Boom link, masses and moments of inertia.

Object link	Mass $[kg]$	Moment of Inertia $[kg \cdot m^2]$
Dolly (link1)	25kg	$I_{xx} = 2.48$, $I_{yy} = 0.97$, $I_{zz} = 3.465$
Link 2	0.6254	$I_{xx} = 0.000907$ $I_{yy} = 0.000907$, $I_{zz} = 0.00181$
Boom (link 3)	29.5	$I_{xx} = 0$ $I_{yy} = 16.904$, $I_{zz} = 16.904$
Link 4	0.879	$I_{xx} = 0$ $I_{yy} = 0.02379$, $I_{zz} = 0.02379$
Link 5	3.624	$I_{xx} = 0.08204$ $I_{yy} = 0.00119$, $I_{zz} = 0.00701$
PTU (link 6)	12.684	$I_{xx} = 0.276$ $I_{yy} = 0.234$, $I_{zz} = 0.0690$
Camera (link 7)	0.185	$I_{xx} = 0$ $I_{yy} = 1.33\dot{1}0^{-5}$, $I_{zz} = 1.33\dot{1}0^{-5}$

platform horizontally comes from the fact that bodies 3 and 4 are part of a parallelogram. There are two constraints for the system. These can be seen in Equations 5.1. Thus the system effectively has only 6 degrees of freedom.

$$\begin{aligned} \theta_{bb1} - \theta_{bt1} &= 0 \\ \theta_{bt1} + \theta_{bt2} &= 0 \end{aligned} \tag{5.1}$$

The inputs acting on the system are the torques Q1 (about y) and Q2 (about z) exerted by the operator, and the torques Q3 and Q4 applied by the pan and tilt motors of

the camera, i.e., $\mathbf{u} = \{Q1, Q2, Q3, Q4\}$. The operator operates the dumbbell at the end of body 3 to facilitate the tracking of the target by the camera. In this analysis, it is assumed that he does not move the cart, although it is straightforward to incorporate that motion as well. The pan and tilt motors correspond to the rotations ψ_c and θ_c, respectively.

The model is obtained in the form of Poincaré's equations (see [22] and [21] for details)

$$\dot{\mathbf{q}} = V(\mathbf{q})\mathbf{p}$$
$$M(\mathbf{q})\dot{\mathbf{p}} + C(\mathbf{q})\mathbf{p} + \mathbf{Q}(\mathbf{p}, \mathbf{q}, \mathbf{u}) = \mathbf{0} \tag{5.2}$$

The generalized coordinate vector $\mathbf{q}$, is given by $\mathbf{q} = [x, y, \psi_b, \theta_{bt1}, \theta_{bt2}, \theta_{bb1}, \theta_{bb2}, \psi_c, \theta_c]^T$. The vector $\mathbf{p}$ is the 7×1 vector of quasi-velocities. The first set of equations are the kinematics and the second are the dynamics of the system.

5.4 Model Validation

In [34] we described an experiment were the camera was tracking a target which moved sinusoidally while booming. The setup can be seen in Chapter 4, Figure 4.8. The Mitsubishi robotic arm was instructed to sinusoidally move the target in the horizontal direction. During this motion, the camera was tracking while the operator was booming. The displacement magnitude was about 0.5 *m* and the frequency was about 0.08 *Hz*. Tracking error, boom angle and target motion information were recorded. To validate this model, we performed a simulation of this experiment using this data. The results matched the experimental data.

To run the simulation we first compiled the Mathematica file. The output is a standard C file which can be edited/compiled using any C compiler. We used Matlab MEX function to create a Dynamic Link Library (dll) file which defines the Simulink S-function. The file was compiled using the command *mex*. The boom model was then imported into

Simulink. Using two LOOK-UP tables, data from the Mitsubishi robot and data from booming were imported into Simulink. A first order model was assumed for the pan-tilt-unit. The loop was closed in Simulink using the proportional controller. A torque that generates a boom motion similar to the experimental one was generated. The target was programmed to move according to the motion of the Mitsubishi robotic arm (Figure 5.3 - top).

After running the simulation, the tracking error was compared to the one obtained experimentally (Figure 5.3 - middle and bottom).

In Figure 5.2, the booming data are plotted. The booming angle in radians is plotted versus time. In the top figure, the experimental booming is shown. In the bottom one, the simulation can be seen. During the experiment, there was no information about the force acting on the boom. In Simulink a torque was applied to the boom. The corresponding booming angle can be seen in the bottom of Figure 5.2. It can be seen that the two curves shown in this figure have the same shape.

The simulation results are plotted in Figure 5.3. In the top figure, the target position can be seen. The tracking error was plotted in pixels for comparison with the experimental case. The last plot shows the experimental tracking error in pixels. The value of the error was about 350 pixels versus 300 pixels in the experiment. One source of error was the friction that appeared in the real boom, which was not modelled. Another source was the torque that was applied to the boom in the simulation. The boom angle in the simulation does not match exactly the experimental boom angle. Since the values of the tracking errors are similar, this suggests that the model is reasonably accurate.

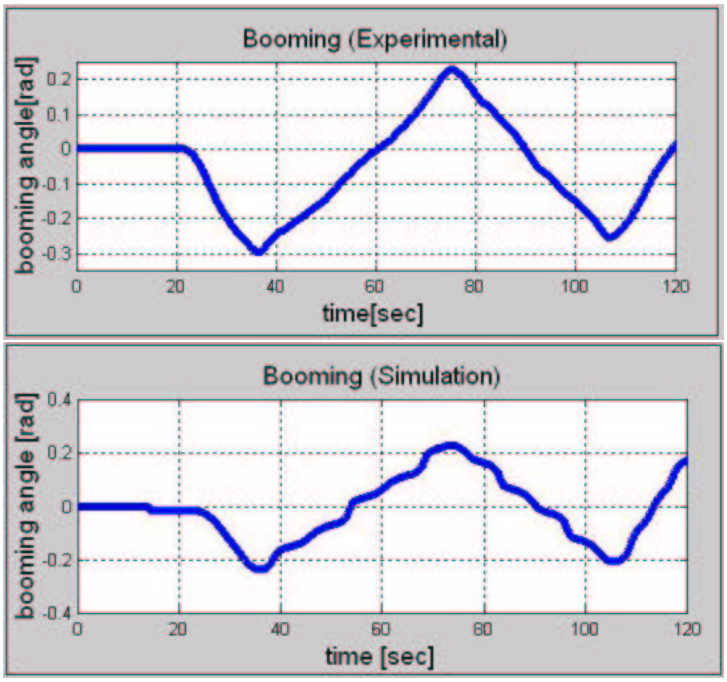

Figure 5.2: Booming - Experiment (top) and simulation (bottom). The two curves are similar.

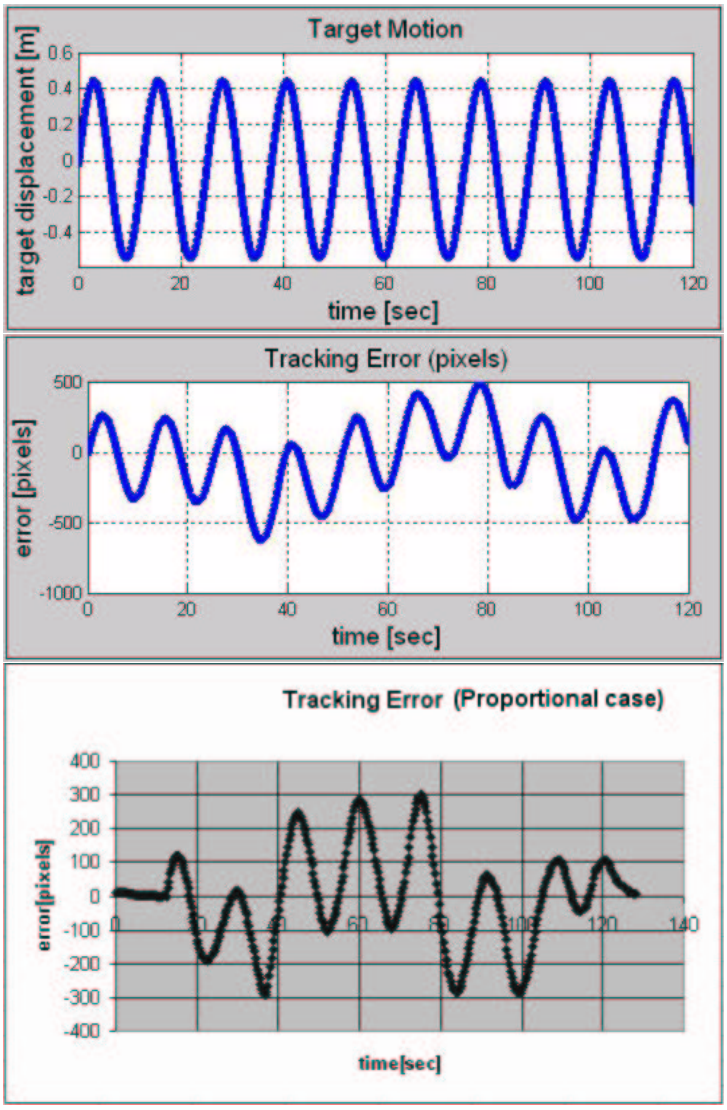

Figure 5.3: Target motion (top figure), error in pixels for simulation (middle) and experimental (bottom).

6: The Output Tracking Regulation Controller

6.1 Introduction

Output tracking regulation is a classic problem. Basically, the goal is to design a feedback law for the purpose of imposing a prescribed steady state response at every external command (in a prescribed family). The difficulty lies in imposing the *tracking error* (which is the difference between *reference* and actual output) to be a function of time which decays to zero as time tends to infinity. This is to be expected for every reference output and every undesired disturbance ranging over prespecified families of functions.

6.2 Theoretical Background

This section briefly presents some theoretical aspects of the output tracking regulation problem. This problem is discussed in detail in [23] [17].

Consider a system modelled by the equations of the form

$$\begin{aligned} \dot{x} &= f(x,w,u) \\ e &= h(x,w) \end{aligned} \tag{6.1}$$

In 6.1, the first equation describes the dynamics of the plant. The state x is defined in a neighborhood of the origin of R^n and the control input $u \in R^m$. This system is subject to input variables $w \in R^r$. The second equation in 6.1 defines the error $e \in R^m$ expressed as a function of the state x and input w. The objective is to design a control law that ensures that the error e decays to zero as the time tends to infinity.

Two cases are presented in [17]. The first one is when the set of measured variables includes all the components of the state x and of the input w. In this case it is said that the controller is provided with *full information*. The second case presents the situation when only the components of the error e are available for measurement. In this case it is said that

the controller is provided with *error feedback*. In case of *full information* the solution is ensured by the theorem below.

Theorem. *The full information output tracking regulation problem is solvable if and only if the pair (A, B) is stabilizable and there exists mappings* $x = \pi(w)$ *and* $u = c(w)$ *with* $\pi(0) = 0$ *and* $c(0) = 0$, *both defined in a neighborhood* $W^o \subset W$ *of the origin, satisfying the conditions*

$$\begin{aligned} \frac{\delta \pi}{\delta w} s(w) &= f(\pi(w), w, c(w)) \\ 0 &= h(\pi(w), w) \end{aligned} \tag{6.2}$$

for all $w \in W^o$.

In [17] it is considered that the input $w(t)$ is generated by a "generator" according to

$$\dot{w} = s(w) \tag{6.3}$$

6.3 Design of the Output Tracking Regulation Controller

If the system is linear, the conditions 6.2 reduce to a linear matrix equation (6.4).

$$\begin{aligned} \dot{x} &= Ax + Pw + Bu \\ \dot{w} &= Sx \\ e &= Cx + Qw \end{aligned} \tag{6.4}$$

The regulator problem is solvable if and only if the linear matrix equations 6.5 are solved by Π and Γ [17].

$$\Pi S = A\Pi + P + B\Gamma$$

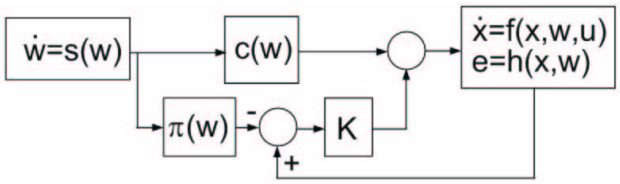

Figure 6.1: The Output Tracking Regulation Controller as it was implemented.

$$0 = C\Pi + Q \tag{6.5}$$

A regulating control can then be constructed as

$$u = \Gamma w + K(x - \Pi w) \tag{6.6}$$

where K is chosen so that the matrix $(A+BK)$ has desired eigenvalues. These eigenvalues determine the quality of the response. The controller structure is shown in Figure 6.1.

A simplified model was used to design this controller. This model has the transfer function

$$\frac{\theta(s)}{V_a(s)} = \frac{0.01175}{1.3s^2 + 32s} \tag{6.7}$$

where the output is the camera angle measured in degrees. In this case, the state space description of the system is given by matrices A, B and C

$$A = \begin{vmatrix} -24.61 & 0 \\ 1 & 0 \end{vmatrix}$$

$$B = \begin{vmatrix} 0.0088 \\ 0 \end{vmatrix}$$

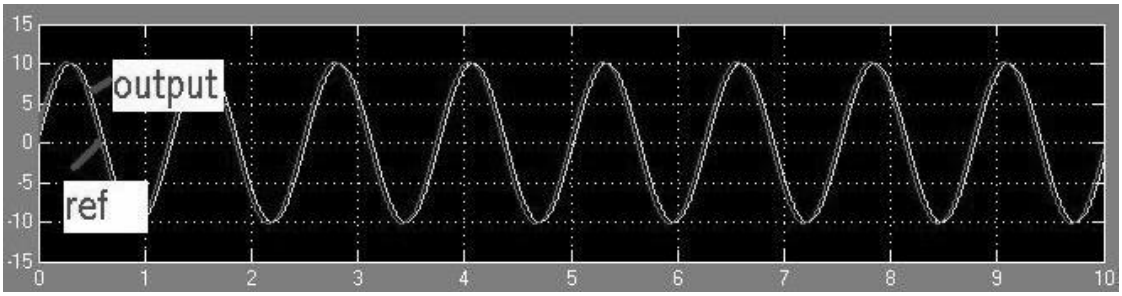

Figure 6.2: The reference as well as the output of the pan-tilt unit using the new controller.

$$C = \begin{vmatrix} 0 & 1 \end{vmatrix}$$

From Equation 6.5

$$\Pi = \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix}$$

$$\Gamma = \begin{vmatrix} -113.6 & 2796.6 \end{vmatrix}$$

The matrix K was

$$K = \begin{vmatrix} -10000 & -380 \end{vmatrix}$$

6.4 Output Tracking Regulation Controller: Simulation Results

Prior to implementation, the new controller was simulated using Matlab-Simulink. The simplified camera model (Equation 6.7) used to design this controller was used for simulation. A sinusoidal reference signal corresponding to a frequency of 1 rad/sec was applied to the controller. Both the reference and the output of the system were plotted on the same axes frame. The plot can be seen in Figure 6.2.

The output of the boom-camera system was simulated in *Matlab* using the model developed in Chapter 5 and the data recorded in Mitsubishi robot experiment. The simulated

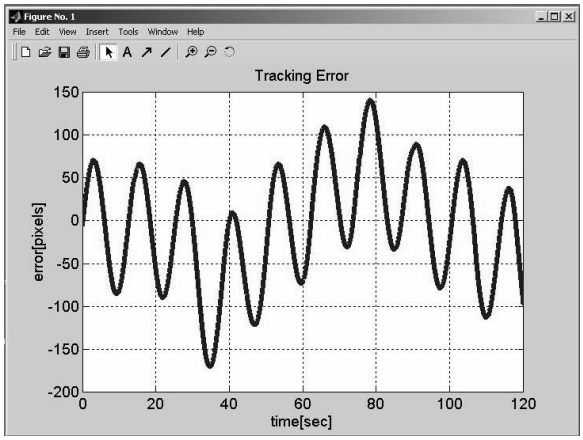

Figure 6.3: The reference as well as the output of the pan-tilt unit using the new controller.

tracking error can be seen in Figure 6.3.

The error is close to ± 50 *pixels* but greater. This controller was implemented. The controller was added as a feedback compensator to the feedforward control scheme. The structure can be seen in Figure 6.4.

6.5 Experiments using feedforward with OTR controller

6.5.1 Mitsubishi Robot Experiment

A couple of experiments were designed to assess the performance of this controller. First, the system tracked the target moved by the Mitsubishi robot. Again, the robotic arm was instructed to sinusoidally move the target. The frequency was approximately 0.08 *Hz* and the magnitude was 0.5 *m*. The camera was followed the target while an operator boomed. Booming data and tracking error were recorded. The plots can be seen in Figure 6.5. In this figure, the top plot represents the target motion. The second plot shows the operator booming. The third plot is the horizontal error when using the OTR controller.

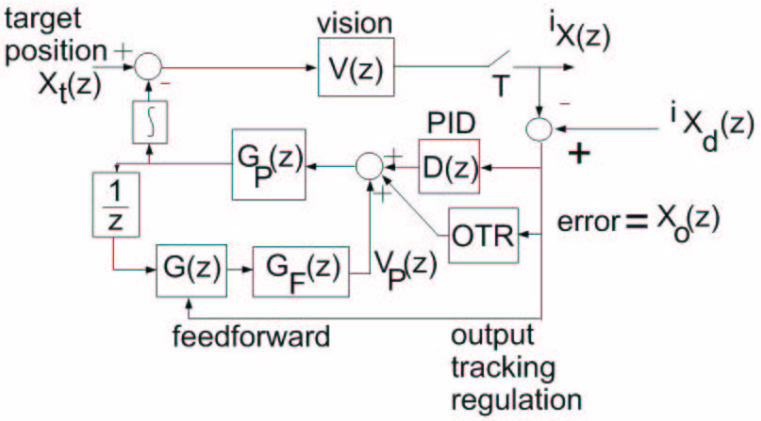

Figure 6.4: The Feedforward Controller with the Feedback Compensation as it was implemented. The output of tracking regulation controller (OTR) is added to the scheme.

6.5.2 Ball tracking Experiment

The output tracking regulation controller was added to the Visual C program for both axes. This controller was tested to track a ball moving between two players. The experiment was set in the lab. The task was to track the ball while booming.

The experiment was videotaped using three cameras. Sequential pictures can be seen in Figure 6.6. The top row shows the camera tracking the ball. The second row shows the boom camera point of view. The third row shows the program working.

It can be seen that the target is precisely detected and tracked.

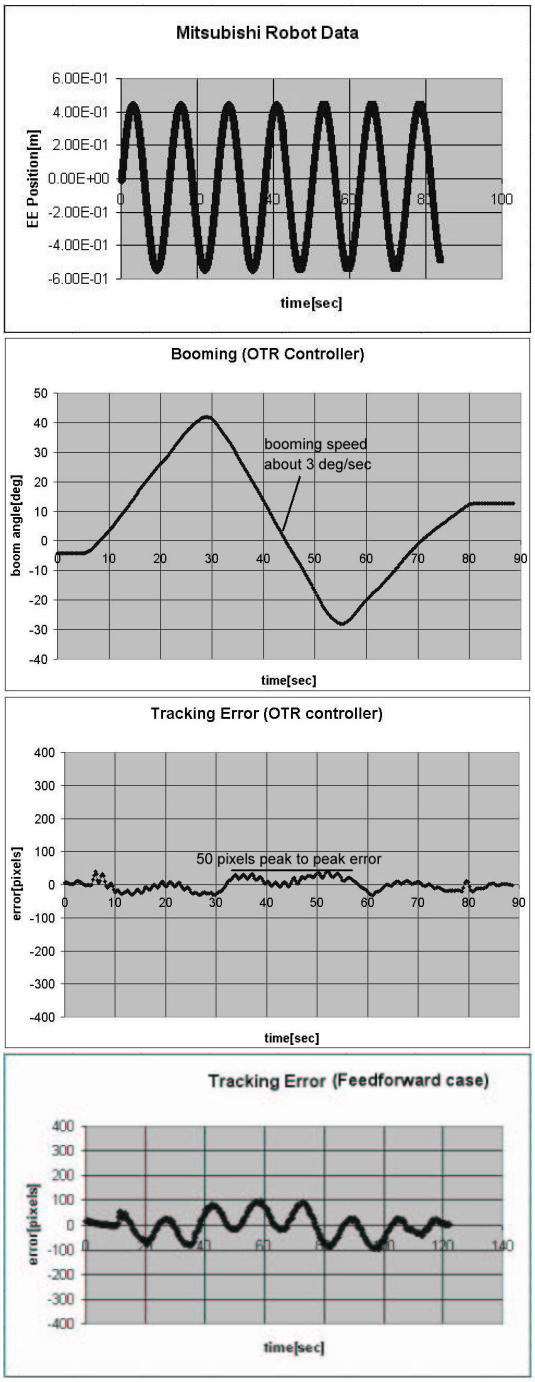

Figure 6.5: The Mitsubishi experiment using the OTR controller. The first figure shows the target moving. The second figure shows the boom motion. The third figure shows the tracking error in case of the output tracking controller. The fourth figure shows the error using the feeedforward controller. It can be seen that by using the OTR controller, the error is less then ± 50 *pixels*.

Figure 6.6: Ball tracking Experiment.

7: Comparison Operator + Vision System versus Operator Only

7.1 Introduction

The experiments described previously show that the system successfully tracks a selected target. In the people tracking experiment, the camera was able to track at much higher booming speeds than before (around $5 deg/sec$).

The ball tracking experiment shows that the target is well detected even while booming. Still, more is needed. The following experiments try to answer the questions, "is this system preferred over an inexperienced operator? Or, is it preferred over an experienced operator?" To answer these questions, two sets of experiments were performed. The target was moved by the Mitsubishi robot. An experienced and an inexperienced operator were asked to track this target with and without the vision system. In the case of manual manipulation of the camera, a joystick was used. The program was modified to move the PTU according to the joystick. In the first set, the boom motion was not restricted.

7.2 Unrestricted Boom Path

Again, the Mitsubishi robotic arm was used. The target was attached to its end-effector. The robot was moving the target in a vertical plane describing a trajectory corresponding to a figure eight. An experienced operator as well as an inexperienced operator were invited to handle the boom and manually coordinate the camera to keep a moving target in the camera's field of view. The target was about $4m$ away from the camera. The maximum speed of the robot's end effector was about $0.2 m/sec$. Data regarding tracking was recorded.

The experiment was set up in the lab. A top-view schematic of the experiment can be seen in Figure 7.1.

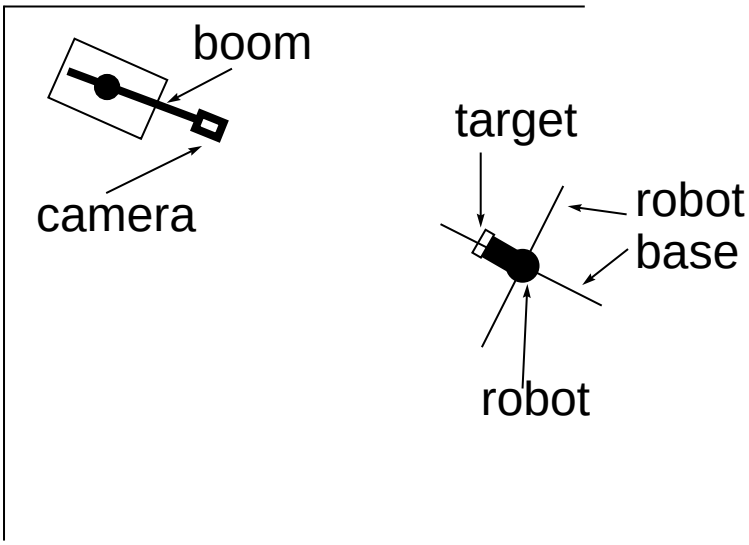

Figure 7.1: The setup using the Mitsubishi robotic arm (top view). The target is moved in a vertical plane. The is motion of the boom is not restricted.

7.2.1 Inexperienced Operator Without Vision

First, an inexperienced operator handled the boom and manually manipulated the camera. The experiment was videotaped using three cameras. Several pictures taken from the experiment are shown in Figure 7.2.

The first observation to be made is that the target is often lost. Tracking such a motion by an inexperienced operator is difficult. As the top row of Figure 7.2 shows, the boom is almost not moving. By concentrating on tracking the target, the operator does not move the boom much.

7.2.2 Inexperienced Operator With Vision

Figure 7.3 shows the images of an inexperienced operator tracking an object with the aid of the vision. It can be seen that the target is never lost. More so, the operator concentrates on booming while the vision system takes care of tracking.

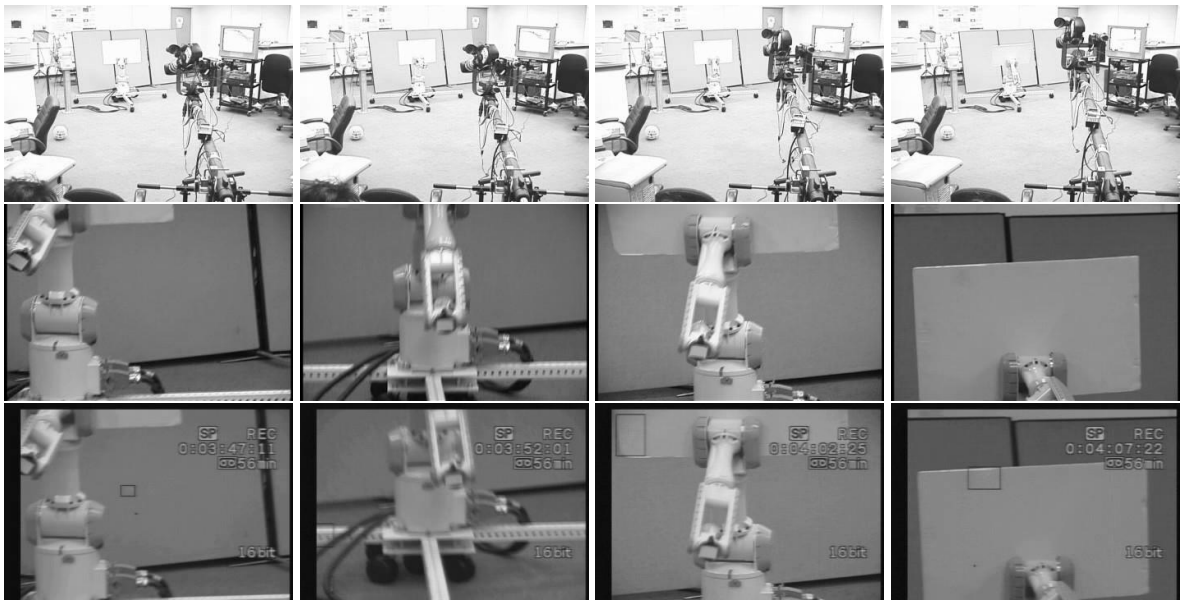

Figure 7.2: Inexperienced operator without the vision system. The boom path was not restricted. The target was sometimes lost. Also, it can be seen that the operator concentrates on moving the camera rather than on booming.

Figure 7.3: Inexperienced operator with the vision system. The boom path was not restricted. The vision system never loses the target. Also, it can be seen that the operator moves the boom.

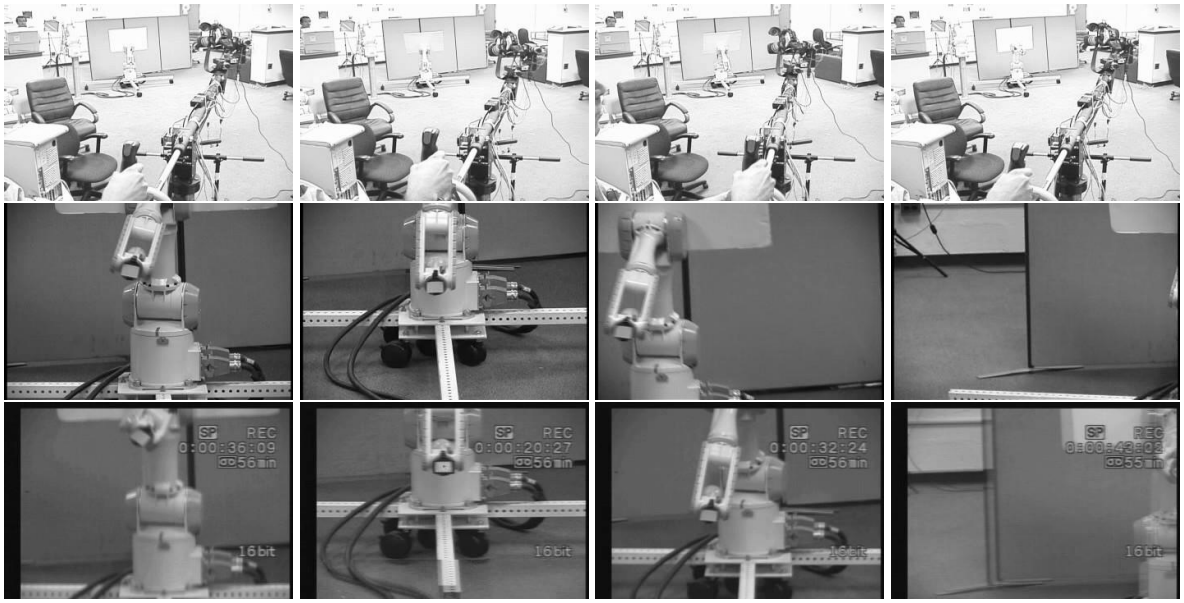

Figure 7.4: Experienced operator without the vision system. The boom path was not restricted. The target was sometimes lost. Also, it can be seen that the operator rather concentrates on moving the camera than on booming.

7.2.3 Experienced Operator Without Vision

An experienced operator boomed this time. The pictures can be seen in Figure 7.4. It can be seen that, despite his experience, there are moments when the target is lost. As expected, the experienced operator is able to track the target better then the inexperienced one. Same as before, the experienced operator concentrates on camera tracking rather than booming.

7.2.4 Conclusion in the Case of Unrestricted Boom Path

From the picture sequence it can be seen that, despite his experience, the experienced operator lost the target just as the inexperienced operator did. Still, this did not happen as often as the inexperienced operator, which lost the target many times. When vision was employed, the target was precisely detected and tracked.

The experiments showed that in the case of a difficult tracking job without the use of

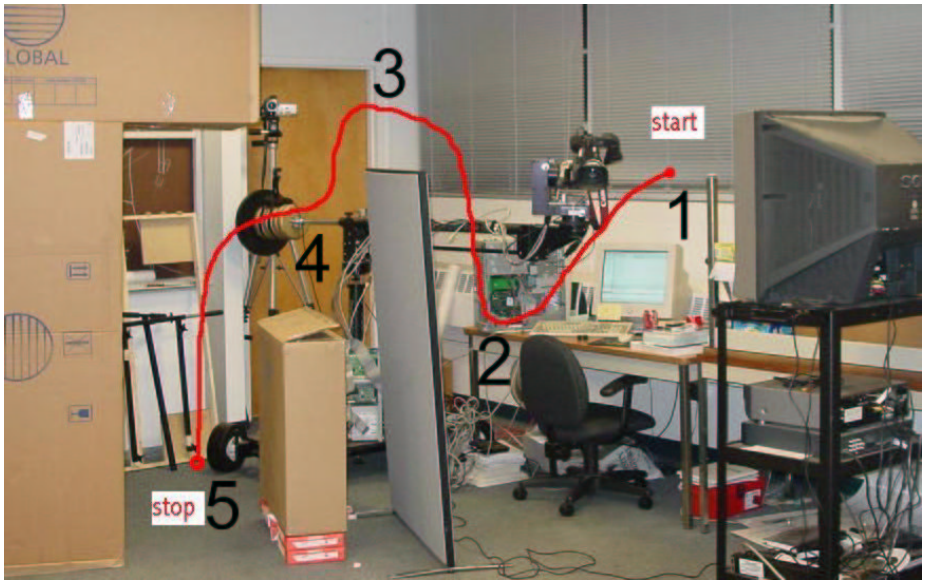

Figure 7.5: The operator will move the camera along the path. He has to avoid hitting the objects as well.

vision, the operators concentrate more on handling the camera than on booming. Because of this matter, the experiment with the experienced operator booming with vision was not performed.

To force the operator to move the boom, in the next set of experiments a booming path was set. Just as before, an experienced as well as an inexperienced operator handled the boom with and without vision.

7.3 Restricted Boom Path

For the next experiment, the boom path is restricted, meaning that the operator had to follow a specific path. This path can be seen in Figure 7.5. Several positions were marked along this path.

The experiment was videotaped. Sequential images can be seen in Figures 7.6, 7.8, 7.11 and 7.13. The images are taken when the camera is in one of the positions marked in Figure 7.5.

Again, the Mitsubishi robot was used. Its end-effector was instructed to move the target on a trajectory corresponding to a figure “8” for about 60 *seconds*. The operator was asked to boom along the path in the same amount of time. An experienced operator as well as a beginner were asked to boom on a predefined path. Each operator boomed two times, first using the vision system and then, using a joystick. The objective was to keep the target in the camera’s field of view while both the target and boom move.

7.3.1 Inexperienced Operator Without Vision

As in the previous experiment, first an inexperienced operator handled the boom and coordinated the camera. The experiment was videotaped using three cameras. A sequence of images can be seen in Figure 7.6. The top row shows the boom in positions “2”, “3”, “4” and “5” (see Figure 7.5). As before, the middle row shows the boom camera point of view. The bottom row shows the program tracking. It can be seen that sometimes the target is lost. When this happens, the program focuses on some other objects in the image. For this reason, the error plot in this case is irrelevant.

It can be seen that the operator is now forced to boom on a specific path. Again, it can be seen that the target was sometimes lost. When this happens, the program focuses on something else, and therefore, the error plot has no relevance in this case. The booming plots are shown in Figure 7.7. It can be seen that, despite the fact that the operator is following the path, the handling of the boom is not smooth in without vision. The states “1”, “2”, “3”, “4” and “5” are marked on both plots. They correspond to the states marked in Figure 7.5.

7.3.2 Inexperienced Operator With Vision

The inexperienced operator then used the vision system to track the same target. The operator had to move the boom on the same path while the vision system was tracked. The experiment was videotaped using three cameras. A sequence of images can be seen in

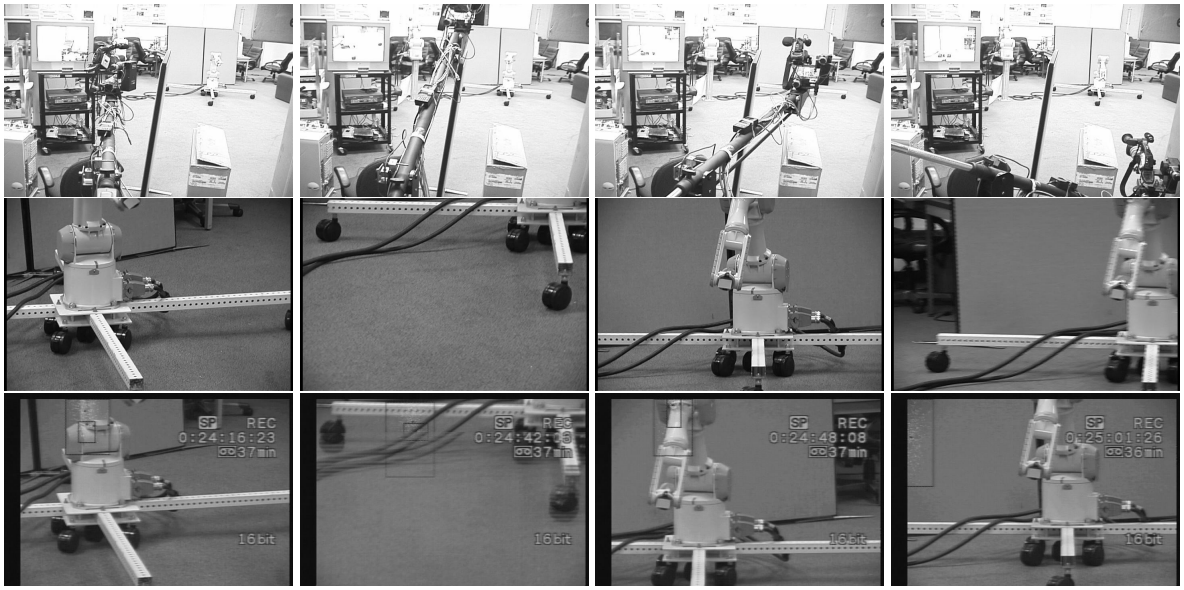

Figure 7.6: Inexperienced operator without vision system. The top row shows the boom in positions “2”, “3”, “4” and “5” (see Figure 7.5). The middle row, shows the boom camera point of view. The bottom row shows the program tracking. The operator is able to boom along the path, but sometimes the target is lost. In this case, the program focuses on other objects in the image.

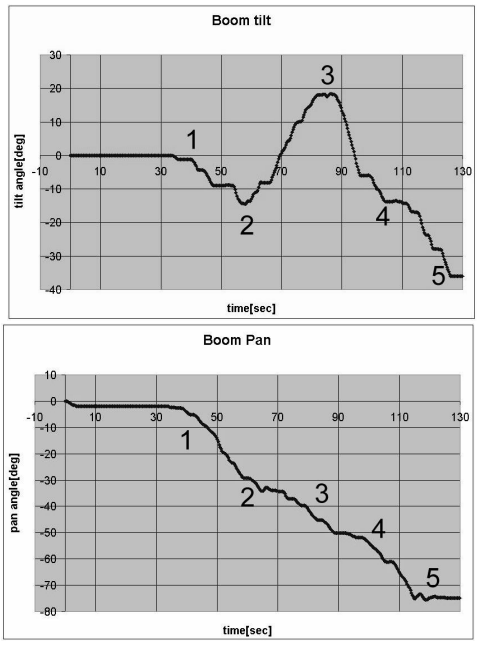

Figure 7.7: Boom tilt (top figure) and pan (bottom figure) angle in the case of an inexperienced operator without vision. It can be seen that booming is not smooth.

Figure 7.8. The top row shows the boom in positions “2”, “3”, “4” and “5” (see Figure 7.5). As before, the middle row shows the boom camera point of view. The bottom row shows the program tracking. It can be seen that the target is never lost. For this reason the error plot in this case is relevant.

Also, the error in the horizontal direction was around ± 50 *pixels* (see Figure 7.9). More so, the output regulation controller (which is implemented for the pan motion) maintains the target very close to the image center. Due to the parallelogram mechanism, tracking regulation is much easier along the vertical direction. Despite this fact, it can be seen that along the horizontal direction the target is much closer to the image center in comparison with a vertical line.

An interesting fact was that in the case of controlling both the boom as well and the camera, the operator needed more than 60 seconds. The reason is that while controlling the camera he moved the boom slower.

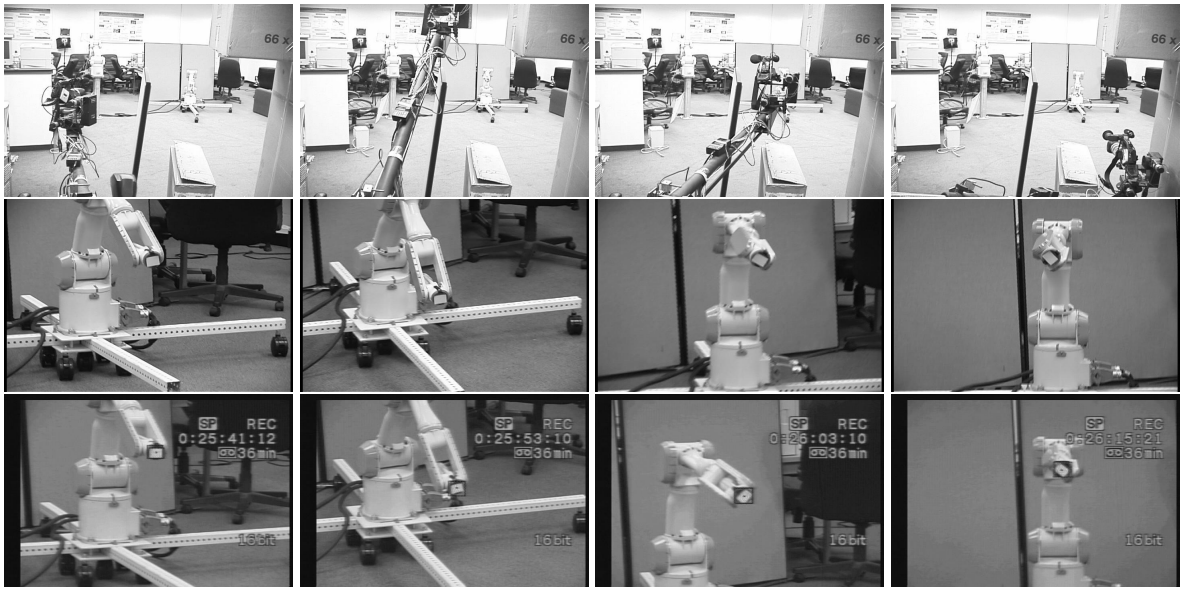

Figure 7.8: Inexperienced operator with vision system. The target is never lost. The top row shows the boom in positions “2”, “3”, “4” and “5” (see Figure 7.5). The middle row shows the boom camera point of view. The bottom row shows the program tracking. Wit the help from vision, the operator is able to boom along the path and the target is never lost.

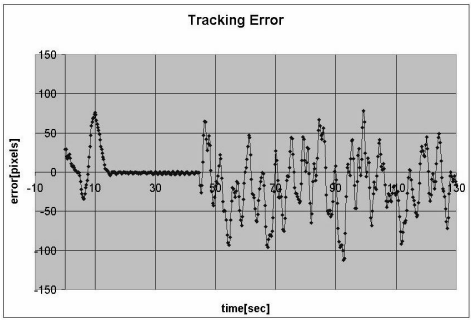

Figure 7.9: The Horizontal Tracking Error of Inexperienced operator using vision.

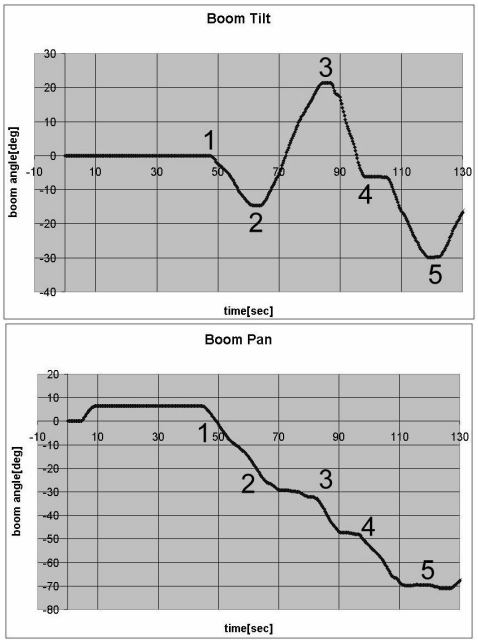

Figure 7.10: Boom tilt (top figure) and pan (bottom figure) angle in the case of inexperienced operator with vision. It can be seen that booming is smoother than when not using vision (Figure 7.7)).

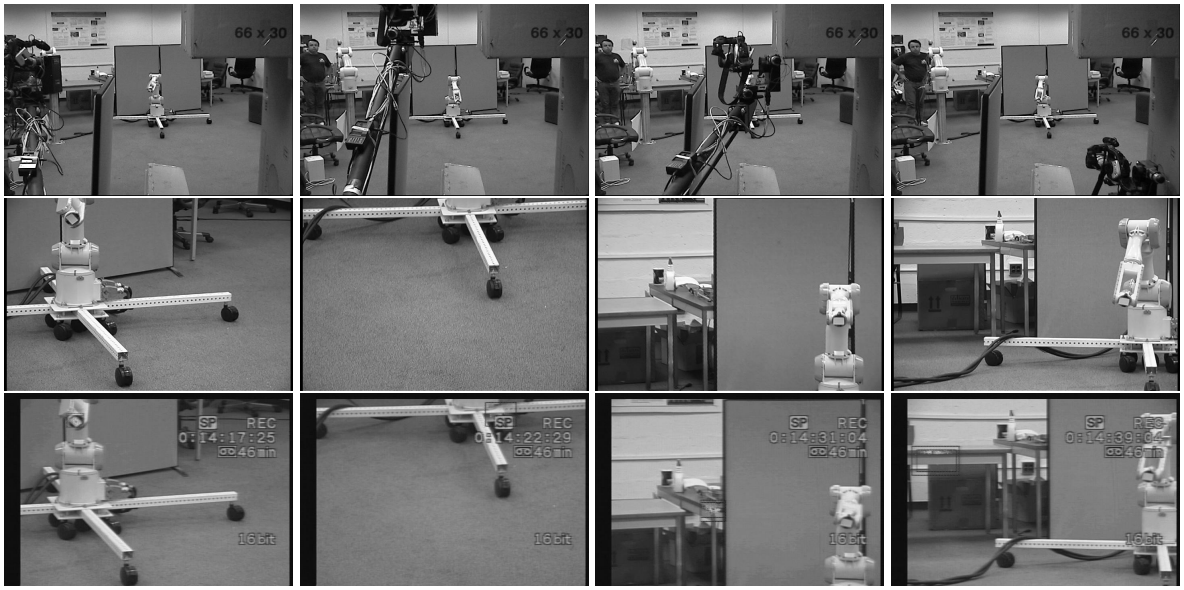

Figure 7.11: Experienced operator without vision. It can be seen that sometimes the target is lost.

7.3.3 Experienced Operator Without Vision

As in the previous experiment, an experienced operator handled the boom and co-ordinated the camera. The experiment was videotaped using three cameras. A sequence of images can be seen in Figure 7.11. The top row shows the boom in positions “2”, “3”, “4” and “5” (see Figure 7.5). As before, the middle row shows the boom camera point of view. The bottom row shows the program tracking. It can be seen that sometimes the target is lost. When this happens, the program focuses on other objects in the image. For this reason, the error plot in this case is irrelevant. the booming plots are shown in Figure7.12. It can be seen that the experienced operator boomed smoothly.

7.3.4 Experienced Operator With Vision

The experienced operator then used the vision system to track the same target. The operator had to move the boom on the same path while the vision system was tracking. The experiment was videotaped using three cameras. A sequence of images can be seen

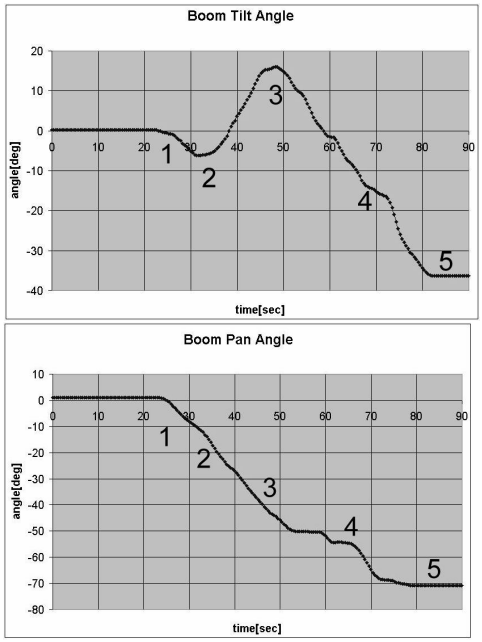

Figure 7.12: Boom tilt (top figure) and pan (bottom figure) angle in case of experienced operator without vision. It can be seen that booming is smoother then in case of inexperienced operator without vision.

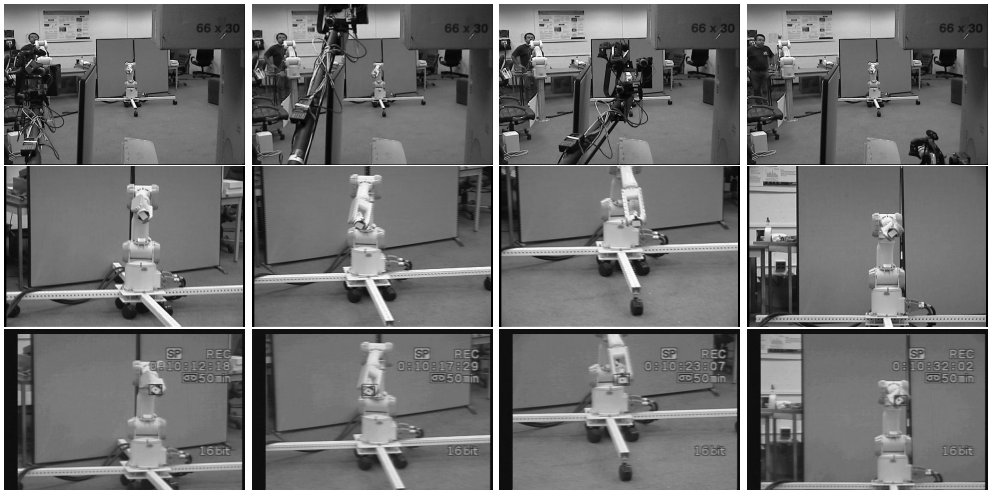

Figure 7.13: Experienced operator with vision system.

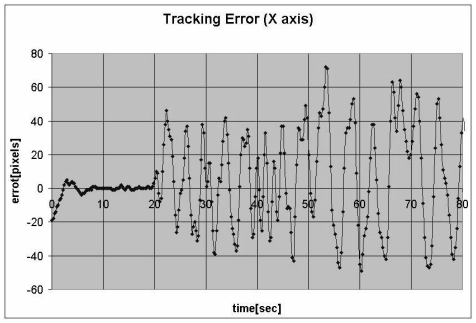

Figure 7.14: Experienced operator using vision. The Horizontal Tracking Error.

in Figures 7.13. The top row shows the boom in positions “2”, “3”, “4” and “5” (see Figure 7.5). As before, the middle row shows the boom camera point of view. The bottom row shows the program tracking. It can be seen that the target is never lost. For this reason the error plot in this case is relevant.

Also, the error in the horizontal direction was around ± 50 *pixels* (see Figure 7.14). More so the output regulation controller (which is implemented for the pan motion) maintains the target very close to the image center. Due to the parallelogram mechanism, tracking regulation is much easier along the vertical direction. Despite this fact, it can be seen

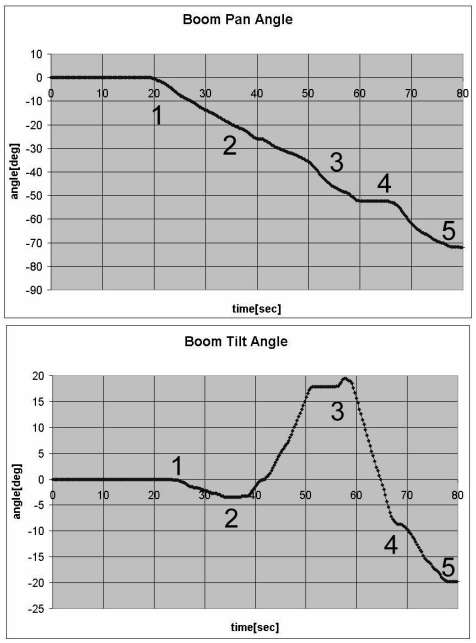

Figure 7.15: Boom tilt (top figure) and pan (bottom figure) angle in case of experienced operator with vision.

that along the horizontal direction the target is much closer to the image center in comparison with vertical line.

7.4 Experiments with OTR controller implemented on both axis

To better approximate the vision system performance, the output tracking regulation controller was implemented on Y direction as well. The experiment was performed once again with both operators. In both cases vision system was used. Again, the Mitsubishi robot is used. Its end-effector was instructed to move the target on a trajectory corresponding to figure “8” for about 60 *seconds*. The operator was asked to boom along the path in the same amount of time. An experienced operator and a beginner were asked to boom on a predefined path. Each operator boomed two times, first using the vision system and then, the camera was rotated by the operator using a joystick. The objective was to keep the target in camera’s field of view while both, target and boom moves.

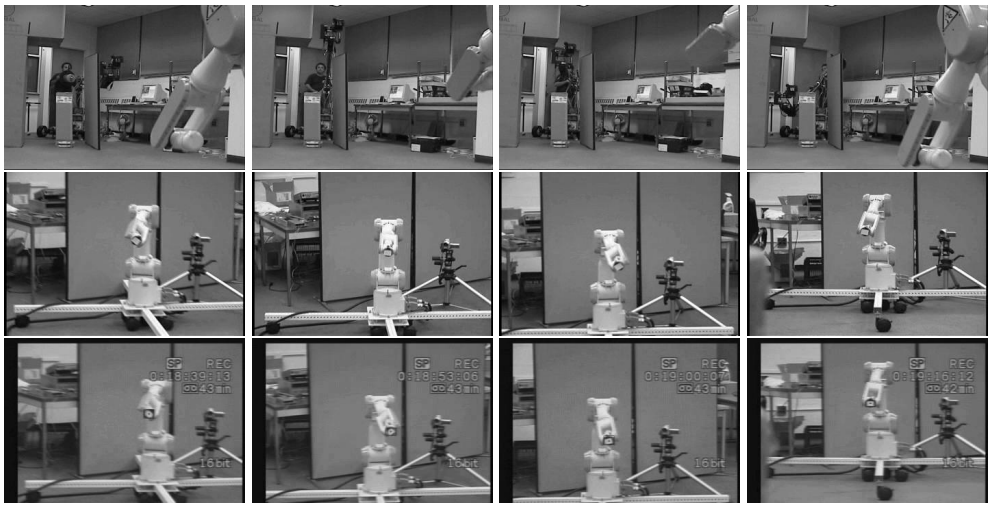

Figure 7.16: Inexperienced operator with vision system. It can be seen that target is much closer to the image center in comparison to any other experiment.

7.4.1 Inexperienced Operator

An inexperienced operator handed the boom first. This experiment was videotaped as well. Sequential images can be seen in Figure 7.16. From the images, it can be seen that the target is very close to the image center. In this case, the error is computed as a squared sum of the horizontal and vertical errors. The error plot is shown in Figure7.18 (bottom).

7.4.2 Experienced Operator

An experienced operator handled the boom second time. This experiment was videotaped as well. Sequential images can be seen in Figure 7.17.

The absolute error is shown in Figure 7.18 (top). One can see that in this case the error is in the same range with the error obtained when inexperienced operator handled the boom.

The boom angles can be seen in Figure 7.19.

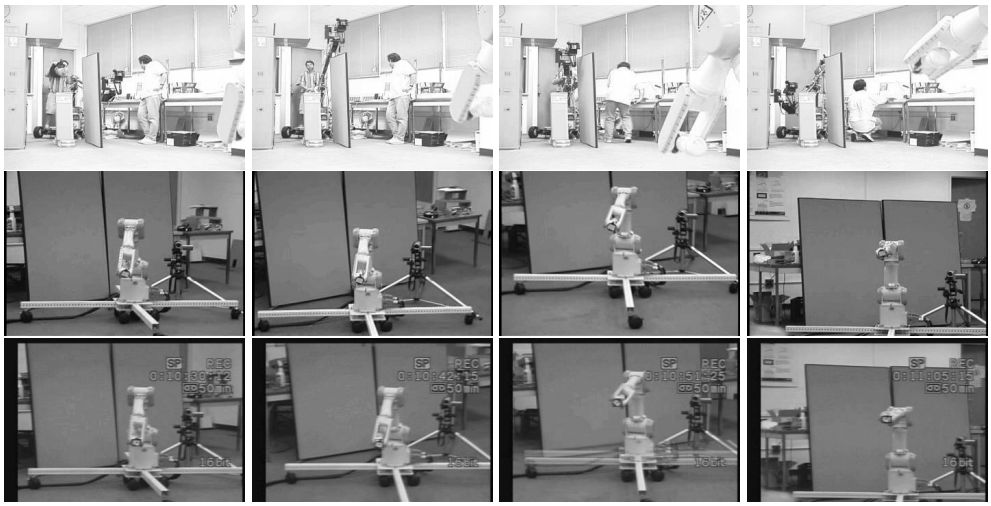

Figure 7.17: Experienced operator with vision system. It can be seen that target is much closer to the image center in comparison to any other experiment.

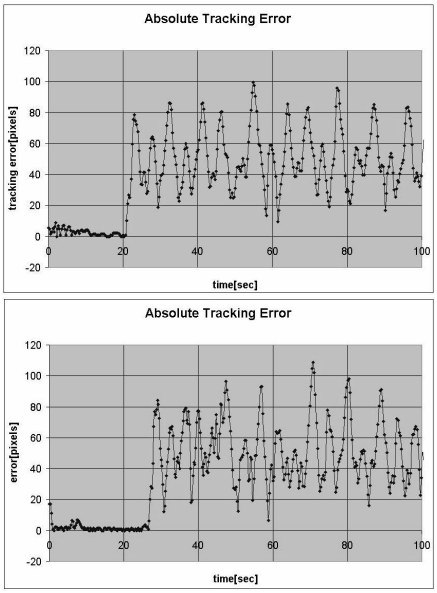

Figure 7.18: Tracking error in case of experienced operator (top) as well as in case of inexperienced operator (bottom). The booming path was restricted. It can be seen that there is no difference between error plots.

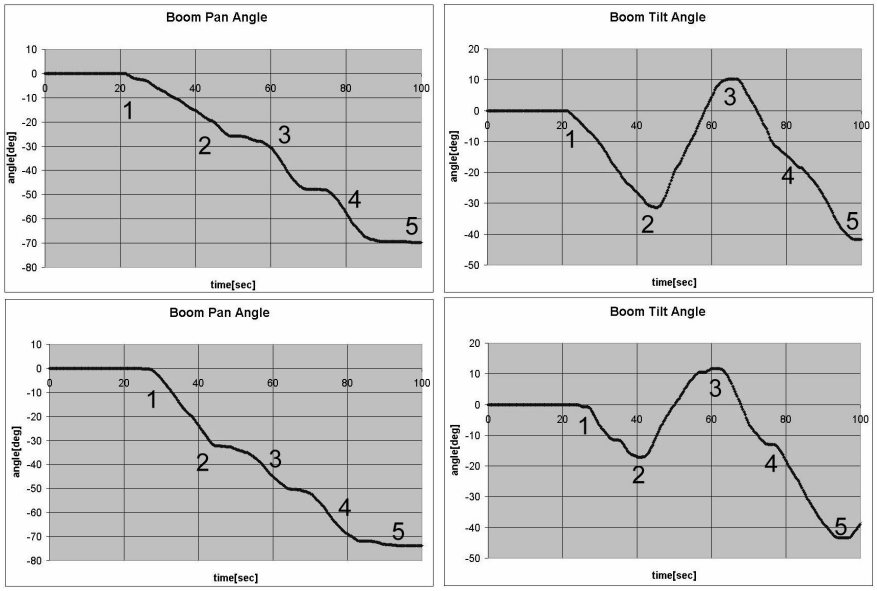

Figure 7.19: Booming angles in case of experienced operator (top) as well as in case of inexperienced operator (bottom). The booming path was restricted. It can be seen that there is not much of a difference between plots.

7.5 Conclusions

This chapter describes some experiments done by both, experienced and inexperienced operators using vision or manually manipulate the PTU-camera. The target was moved by a Mitsubishi robotic arm. Its end-effector trajectory was corresponding to figure “8”. Data regarding booming as well as tracking error was recorded in all experiments. In case of not vision, the target was lost a couple of times. In case of loosing the target, because the program focuses on something else, the tracking error is not relevant.

Also, the use of vision system definitely helps the inexperienced operator (the target was precisely detected and tracked). More then that, using the vision system, an inexperienced operator can achieve similar performance as a skilled operator. The experiments also, shows that there are situations when vision is helpful for a skilled operator as well.

8: Conclusions and Future Work

8.1 Conclusions

In chapter 7 a set of experiments was performed to answer the question wether the vision system is a help for the operator. For this purpose, an inexperienced as well as an experienced operator were asked to move the arm. Vision proved to be a big help for especially in case of small booming experience.

In the situation described in chapter 7 the inexperienced operator, barely could track the target. With vision the situation was completely different. The target was precisely detected and tracked.

Despite the fact that due to its experience the second operator was able to track the target. Still there were situations when the target was lost. This was not happened in case of using vision. So, at least, there are situations when computer vision can help even an experienced operator.

8.2 FutureWork

There still is a lot of work until this setup becomes an off-the-shelf product. Perhaps the next step is to add zooming to this program. Zooming is helpful in situations when the system is almost to loose the target. In this case, the program can automatically command zooming out. After zooming out, the target gets closer to the image center. Also its speed is reduced. Whenever is possible, the program can zoom back in. Today cameras have what is called *LANC Protocol* which allows remote zooming control. This communication protocol was developed by Sony. Using an appropriate hardware the computer can zoom in or out.

The second improvement can consider multi-target acquisition before tracking is launched. During tracking, target switching is possible. Automatic as well as human controller switching can be considered. In such case, at one moment the program can stop

Figure 8.1: A boom fully actuated. The motion is precisely prescribed and repeatable.

tracking one target and start tracking another one. This might be useful in case of having more actors in the scene.

The third improvement can consider several pre-implemented scenario situations. Automated actuation of boom can be done. Using two electric motors, the boom can be precisely panned and tilted. More than that, these motions can be prescribed and they are also repeatable. In the same time, the boom's angular position and velocity is precisely known. Even more, the dolly can be electrically actuated. Such a boom, can be seen in Figure 8.1. It was preprogrammed to fill a glass of juice. A sequence of images can be seen here.

The fourth direction of improvement can consider the boom itself. As the boom is gimballed its top describes o sphere in the space. So, only circular moves can be considered. There are certain situations when a linear camera move is required. To achieve this situation, the boom can be made extensible. By panning up in the same time as extending, the camera can be made to move linear with respect to the target. This type of motions cannot be achieved with the actual structure.

Another improvement that can be done is to use a laptop instead of a desktop. The only reason for this is that laptops have a battery. If PTU motors can be powered using a battery as well, then the whole ensemble will not need power network outlets.

Bibliography

[1] Allen, P.K., Timchenko, A., Yoshimi, B., Michelman, P., “Real-time Visual Servoing”, *IEEE Int Conf on Robotics and Automation*, Vol. 4 pp. 851 1991.

[2] Bensalah, F., Chaumette, F., “Compensation of Abrupt Motion Changes in Target Tracking by Visual Servoing”, *Proc. IROS ’95*, p.181, Pittsburgh, 8/95.

[3] Burt, P.J., Adelson, E.H., “The Laplacian Pyramid as a Compact image Code” *IEEE Transactions on Communcations*, Vol. COM-31 No. 4, April 1983.

[4] Canon, D.J., “Experiments With a Target-Threshold Control Theory Model for Deriving Fitts Law Parameters for Human-Machine Systems”, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 24 No. 8, pp. 1089-1098 August 1994.

[5] Chaumette F., Rives P., Espiau B. “Positioning of a robot with Respect to an Object, Track it and Estimating its Velocity by Visual Servoing” *Proceedings of the 1991 IEEE International Conference Robotics and Automation*, Sacramento, CA, April 1991.

[6] Corke P.I., Good M.C.“Dynamic Effects in Visual Closed-Loop Systems” *IEEE Transactions on Robotics and Automation* Vol. 12 No. 5 October 1996.

[7] Corke, P.I., “Visual Control of Robots high-performance visual servoing” Research Studies Press LTD, Taunton, Somerset, England 1997.

[8] Chaumette, F., Santos, A., “Tracking a Moving Object by Visual Servoing” *Proc. of 12th World Congress IFAC*, Vol. 9 pp. 409-414, Sydney, July 1993.

[9] Dzialo K.A., Schalkoff R.J. “Control Implications in Tracking Moving Objects Using Time-Varying Perspective-Projective Imagery”, *IEEE Transactions on Industrial Electronics* Vol. IE-33, No. 3, August 1986.

[10] Feddema, J.T., Lee, G.C.S., “Weighted Selection of Image Features for Resolved Rate Visual Feedback Control” *IEEE Trans Robotics and Automation* V7 N1 2/91

[11] Ferrell W.R., “Remote manipulative control with transmission delay”, *IEEE Trans. Human Factors in Electronics* HFE-6, No. 1, 1965.

[12] Ferrier N. “Achieving a Fitts Law Relationship for Visual Guided Reaching”, *Int. Conference Computer Vision (ICCV)* Bombay, India, pp. 903-910, January 1998.

[13] Fitts P. M., “The Information capacity of the human motor system in controlling the amplitude of movement”, *J. of Experimental Psychology* Vol. 47 No. 6, pp. 381-391, 1954.

[14] Hill J., Park W.T., “Real time control of a robot by visual feedback in assembling tasks” *Pattern Recognition* Vol. 5 pp. 99-108, 1973.

[15] Hutchinson S., Hager G.D., Corke P.I., "A Tutorial on Visual Servo Control", *IEEE Transactions on Robotics and Automation* Vol. 12 No. 5, pp. 651-670 October 1996.

[16] Isard, M., Blake, A., "CONDENSATION – Conditional Density Propagation for Visual Tracking", *Int. J. Computer Vision*, V29, N1, pp. 5-28, 1998.

[17] Isidori, A., "Nonlinear Control Systems. 3rd ed." *Springer Verlag 1995*

[18] Kalata, P.R., Murphy, K.M., '$\alpha - \beta$ Target Tracking with Track Rate Variations", *Proceedings of the Twenty-Ninth Southeastern Symposium on System Theory*, pp. 70-74, March 1997.

[19] Kalman R., "A new approach to linear filteringand prediction problems. *Journal of Basic Engineering* pp. 35-45, March 1960.

[20] Koivo, A.J., "Real-time Vision Feedback for Srvoing Robotic Manipulator with Self-Tuning Control", *IEEE Trans Syst Man Cybern* V21 N1 january 1991.

[21] Kwatny H.G., Blankenship G.L."Symbolic control of models for multibody dynamics", *IEEE Transaction on Robotics and Automation*, Vol. 11 No. 2 pp. April 1995.

[22] Kwatny H.G., Blankenship G.L., "Nonlinear Control and Analytical Mechanics: a computational approach", Boston: Birkhäuser, 2000.

[23] Kwatny H.G., Kalnitsky K.C., "On Alternative Methodologies for the Design of Robust Linear Multivariable Regulators" IEEE Transactions on Automatic Control, Vol. AC-23 No. 5, October 1978.

[24] Hill, J., Park, W.T. "Real Time Control Of A Robot With A Mobile Camera" *Proc 9th ISIR*, Washington D.C. Mar 1979, pp. 233-246.

[25] Oh, P.Y. Allen P.K. *"Visual Servoing by Partitioning Degrees of Freedom"*, IEEE Transactions on Robotics Automation Vol. 17 No. I, February 2001.

[26] Papanikolopoulos N.P. Khosla P.K. Kanade T., *"Visual Tracking of a Moving Target by a Camera Mounted on a Robot: A Combination of Vision and Control"*, IEEE Transactions on Robotics and Automation Vol. 9 No. 1 February 1993.

[27] Radix, C.L., Robinson, P., Nurse P., "Extension of Fitts' Law to Modeling motion Performance in Man-Machine Interfaces" IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans, vol. 29, No. 2 March 1999

[28] Rizzi, A.A., Koditschek, D.E., "A Dynamic Sensor for robot Juggling" *Visual Servoing*, K. Hashimoto, Ed. Singapore: World Scientific, pp. 1 1993.

[29] Sanderson A.C., Weiss L.E., "Image-based visual servo control using relational graph error signals" *Proc IEEE*, pp. 1074-1077, 1980.

[30] Sheridan T.B., Ferrell W.R., "Man-Machine Systems: Information, Control, and Decision Models of Human Performance", MIT Press, Cambridge, Massachusetts, 1994.

[31] Sheridan T.B., “Space Teleoperation Through Time Delay: Review and Prognosis” *IEEE Transaction on Robotics and Automation*, Vol. 9, No. 5 pp. 592-606, 1993.

[32] Sklansky J., “Optimizing the dynamic parameter of a track-while-scan system” *RCA Laboratories, Princeton*, NJ June 1957.

[33] Stanciu R., Oh P.Y., “Designing Visually Servoed Tracking to Augment Camera Teleoperators” *IEEE Intelligent Robots and System (IROS)*, Lausanne, Switzerland, Vol. 1, pp. 342-347, 2002.

[34] Stanciu R., Oh P.Y., “Feedforward Control for Human-in-the-loop Camera Systems” *IEEE International Conference on Robotics and Automation (ICRA)*, V1. pp-1-6 New Orleans, LA, April 2004.

[35] Stanciu, R., Oh, P.Y., “Human-in-the-loop Visually Servoed Tracking” *International Conference on Computer, Communication and Control Technologies (CCCT)*, Vol. 5, pp. 318-323, Orlando, FL, July 2003.

[36] Tenne, D., Singh, T., “Optimal Design of $\alpha - \beta - (\gamma)$ Filters”, *Proceedings of the American Control Conference*, Vol. 6 pp. 4348-4352, June 2000.

Vita

ION RARES STANCIU

EDUCATION

Doctor of Philosophy, Mechanical Engineering, Drexel University, December 2004
Master of Science, Electrical Engineering, Technical University of Timisoara, June 1995
Bachelor of Technology, Electrical Engineering, Technical University of Timisoara, June 1994
Bachelor of Technology, Electronics and Telecomunication Engineering, Technical University of Timisoara, June 1997

RESEARCH INTERESTS

Image Processing, Visual Servoing, Robotics, Control.

RESEARCH EXPERIENCE

Research Assistant, September 2001 - December 2004
Department of Mechanical Engineering and Mechanics, Drexel University.

TEACHING EXPERIENCE

Teaching Assistant, September 2001 - December 2004
Department of Mechanical Engineering and Mechanics, Drexel University
Teaching assistant for several courses, namely, Introduction to Controls, Mechanics of Vibrations, Theory of Machines, Dynamics, Mechanics of Materials

MAJOR PUBLICATIONS

Stanciu R., Oh P., “Designing Visually Servoed Tracking to Augment Camera Teleoperators,” in *International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, October 2002.

Stanciu R., Oh P., “Feedforward Control for Human-in-the-Loop Camera Systems,” in *International Conference on Robotics and Automation*, April 2004, New Orleans Louisiana.