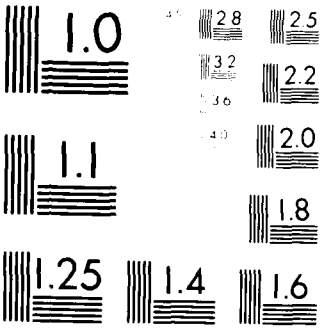
AD-A151 933 POSITIONING OF A ROBOTIC MANIPULATOR THROUGH THE USE OF VISUAL FEEDBACK(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH M T O'DONNELL MAY 85 1/2

UNCLASSIFIED AFIT/CI/NR-85-20T F/G 6/4 NL



MICROCOPY RESOLUTION TEST CHART

UNCLASS

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/CI/NR 85-20T	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Positioning Of A Robotic Manipulator Through The Use Of Visual Feedback		5. TYPE OF REPORT & PERIOD COVERED THESIS/DISSERTATION
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Maurice Thomas O'Donnell		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS AFIT STUDENT AT: University of New Hampshire		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS AFIT/NR WPAFB OH 45433		12. REPORT DATE May 1985
		13. NUMBER OF PAGES 113
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASS
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

APPROVED FOR PUBLIC RELEASE: IAW AFR 190-1

LYNN E. WOLAVER
Dean for Research and
Professional Development
AFIT, Wright-Patterson AFB OH

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

ATTACHED

DTIC
ELECTE
MAR 7 1985
S E D

AD-A151 933

DTIC FILE COPY

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASS

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

85 03 11 045

2LT MAURICE O'DONNELL
USAF
1985
113 pages
M.S. ELECTRICAL ENGINEERING
UNIVERSITY OF NEW HAMPSHIRE

Accession For
NTIS GRA&I X
DTIC TAB
Unannounced
Justification
By
Distribution/
Availability Codes
Avail and/or
Dist Special
A-1

POSITIONING OF A ROBOTIC MANIPULATOR THROUGH THE USE OF VISUAL FEEDBACK

By

Maurice Thomas O'Donnell

B.S., University of New Hampshire, 1983

A THESIS

Submitted to the University of New Hampshire

in Partial Fulfillment of

the Requirements for the Degree of

Master of Science

in

Electrical Engineering

May, 1985

This thesis has been examined and approved.

Filson H. Glanz

Thesis director, Filson H. Glanz,
Associate Professor of Electrical and
Computer Engineering

W. Thomas Miller, Associate Professor of
Electrical and Computer Engineering

Gordon Kraft, Associate Professor of
Electrical and Computer Engineering

12-20-84

Date

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to Professor Filson H. Glanz who generously gave much time and thought to helping me complete this thesis research. I would also like to express my gratitude to the United States Air Force for providing the means by which I could afford to receive both my undergraduate and graduate degrees and still provide for the needs of my family. Above all I would like to give special thanks to my family; Linda, Michael, Kerilyn, and Jennifer who put up with an absent father and husband on many important occasions during the last four and one-half years with few complaints. In addition I would like to thank them for some very happy memories and giving me a place I enjoyed calling 'home'.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS ---------- iii

LIST OF TABLES ---------- vi

LIST OF FIGURES ---------- vii

ABSTRACT ---------- ix

CHAPTER I INTRODUCTION ---------- 1

CHAPTER II IMAGE ACQUISITION AND PROCESSING ---------- 7

Computer Vision Systems ---------- 7

Static Scene Analysis ---------- 16

Dynamic Scene Analysis ---------- 20

Equipment Used in the Study ---------- 22

Processing Technique Employed ---------- 24

CHAPTER III ROBOTIC MANIPULATOR KINEMATICS ---------- 31

Forward Kinematics ---------- 32

Forward Kinematics for P-5 Robot ---------- 47

Inverse Kinematics ---------- 51

Inverse Kinematics for P-5 Robot ---------- 53

Calibration of Camera to Robot ---------- 57

Components of the System ---------- 59

The Estimation Used in the Demonstration Task ----- 62

CHAPTER IV EXPERIMENTAL RESULTS AND ANALYSIS ---------- 67

Experiments and Results ---------- 67

Model for Experimental Results ---------- 75

CHAPTER V CONCLUSIONS AND FUTURE RESEARCH ---------- 82

Suggestion for Future Research ---------- 83

REFERENCES ---------- 89

APPENDIX A: LEAST SQUARES CALIBRATION METHOD AND SOFTWARE LISTING- 92

APPENDIX B: COMPUTER PROGRAM LISTING ---------- 100

LIST OF TABLES

TABLE		PAGE
3.1	P-5 Robot Joint Parameters	48
4.1	Image Processing Error Data	70
4.2	Overall System Error Test Data	74
4.3	General Discrete Kalman Filter	77

LIST OF FIGURES

FIGURE		PAGE
1.1	System Block Diagram	4
2.1	Image Orthicon Camera	9
2.2	Image Dissector Camera	10
2.3	The Vidicon Camera	11
2.4	CCD Array (Line Transfer)	12
2.5	CID Array	12
2.6	Mechanical "Drum" Scanning Mechanism	13
2.7	Dual Moving Mirror Image Plane Scanner	14
2.8	Examples of Compass Gradient Masks	19
2.9	Image Technology, Inc. Hardware	22
2.10	512x512 Frame Buffer	25
2.11	512x512 Frame Buffer with Three Images	26
2.12	Turtle Movement	29
2.13	Cross Hairs Method	30
3.1	Two Types of Robotic Joints	31
3.2	Body Attached Coordinate Frame and Reference Frame	33
3.3	Rotation of the Body Attached Frame	34
3.4	Orientation Vectors Associated with Manipulator	37
3.5	Euler Angles	39
3.6	Roll, Pitch, and Yaw Coordinates for a Manipulator	40
3.7	Cylindrical Polar Coordinates	41
3.8	Spherical Polar Coordinates	41
3.9	Pure Rotation and Translation Matrices	43

FIGURE		PAGE
3.10	Joint Variables and Quantities	46
3.11	Five Link Robot Similar to the P-5 Process Robot	48
3.12	P-5 Robot, Vidicon Camera, Laser	57
3.13	Block Diagram of System	59
3.14	Estimation Technique	64
4.1	Variance of Error in Estimation as a Function of Prediction Time	76
4.2	Kalman Filter Estimates and Experimental Observations for Circular Motion	80
4.3	Circular Motion Prediction Using Linear Predictive Algorithm	81
5.1	Averaging Technique for Noise Problem	85
5.2	Image Field at Two Different Heights	87

ABSTRACT

POSITIONING OF A ROBOTIC MANIPULATOR
THROUGH THE USE OF VISUAL FEEDBACK

by

MAURICE THOMAS O'DONNELL
University of New Hampshire, May, 1985

A system for positioning a general purpose industrial robot by means of information extracted by a computer vision system is described. The specific example implemented was the interception of a moving object by a robotic manipulator based on three sequential images of the object. The techniques used and constraints imposed are discussed. The robot used is an industrial G.E. P-5 robot with five degrees of freedom. Several experiments used to evaluate the various components of system error were run. The results of these experiments were compared to the behavior of a linear predictive Kalman filter model of the system. A list of conclusions is presented along with a discussion of particular areas where improvements on this system can be made. A listing of the Fortran programs used and an outline of the camera-robot coordinate system calibration process are included in the appendices.

CHAPTER I

INTRODUCTION

The underlying objective of this project is to direct the movement of a robotic manipulator using information acquired through an image acquisition system. As an example of such a movement, a robotic manipulator is driven to intercept a moving object. The prediction of the object's position at a fixed future time is based on spacial information extracted from three sequential images of the moving object.

For the most part in today's industrial workplace robots are performing what seems to be rather trivial tasks. That is, tasks consisting of maneuvers that can be played back over and over again. Possibly this could be the primary reason why industry has been the ground breaking area for use of robotic manipulators. This is because a majority of industrial tasks are clearly repetitive and can be expressed as a sequence of fixed motions [Horn and Ikeuchi, 1984]. Tasks that have been considered tedious, boring, or even hazardous, such as spray painting, welding, and part manipulation have been the perfect opportunity to nurture robot technology. The particular maneuvers could be taught to the robot by moving the robot through the sequence of motions while the robot control facilities would store the sequence in some type of memory. Then merely by running the stored information through a loop the robot would be able to proceed with little human intervention.

The drawback with these maneuvers is that when the environment changes in some way the system breaks down rather quickly. For example,

manipulate the data in a specific way to yield some additional information or to better order or clarify the information that is already stored.

Finally, the output device presents this new data in some form to be used by the operator. It may consist of a monitor which simply displays the digitized image. Furthermore, if the output is information about an image the signal may be given to any one of a large group of devices, such as a controller, a robot, a burglar alarm, a communications channel and so on.

When discussing the image signal in monitors or television screens, there are some standard features. The image or scene on a television screen is due to the presence of an electron beam striking a phosphorous coating. That is, the gray level of the particular point in the scene depends on the intensity of the electron beam at that point. It is fairly standard for the electron beam to be moved from left to right and top to bottom. The term 'field' has been applied to the image produced by a sequence of 262½ horizontal scans lines at a rate of 60 times per second [Nanin, 1983]. By interlacing two such fields into a single image, a single frame is generated. The frame rate is therefore approximately 30 frames per second. This rate prevents the appearance of flicker in the scene because the human visual system is able to sustain the image between frames.

Obviously, the ability for the monitor or television receiver to reproduce the image acquired by the camera depend in large part on whether the scanning systems of the camera and receiver are in 'sync'. This synchronization is performed by the generation of synthetic video signals by the camera at particular timing spots. These timing spots

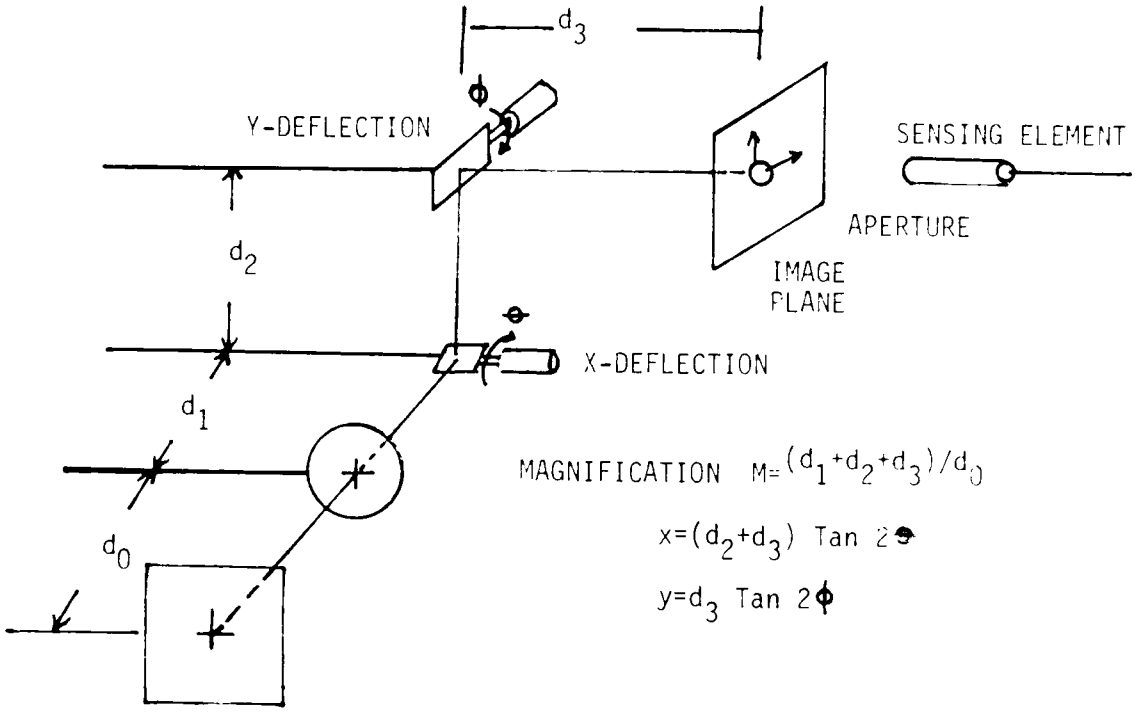


Figure 2.7 Dual Moving Mirror Image Plane Scanner [Castleman, 1979]

The current technology of digitizers, as was stated before, will take an analog signal and quantize it through an analog-to-digital converter and then store the information in some type of semiconductor memory. For the reader interested in the A/D and quantization information Hoeschele's text [Hoeschele, 1968] presents a complete presentation on the subject. For those concerned with use and types of semiconductor memory Muroga's text [Muroga, 1982] presents an adequate discussion in this area.

At this point in the process there is available to the user a two-dimensional representation of the image or scene located in a semiconductor memory which can now be processed by the computer. It is obvious that the main task of the computer in the vision system is to

Some popular mechanical scanning devices are described below.

Mechanical Drum: With this device the image is wrapped around a cylindrical drum. The image is then rotated past a stationary aperture. The aperture is moved by way of a lead-screw. After an entire line is digitized the lead screw is repositioned. This process is repeated until the entire image is scanned [Figure 2.6].

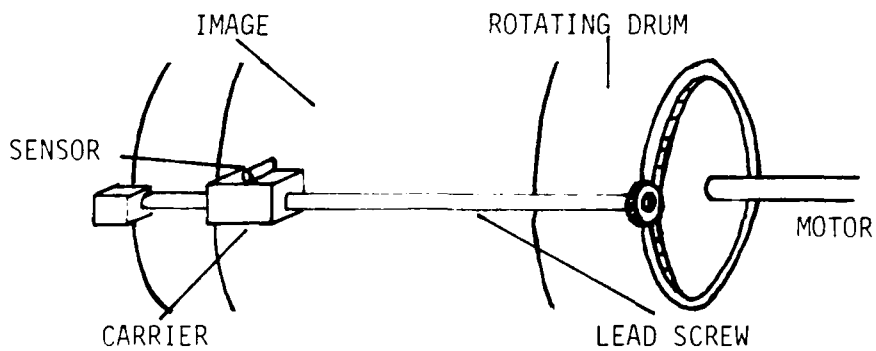


Figure 2.6 Mechanical "Drum" Scanning Mechanism [Castleman, 1979]

Flat bed scanner: This device is similar to the drum system, however the image is placed on a flat bed. In this structure either the bed or the image is repositioned in the digitizing process.

Laser scanner: A source of light (laser) and a mirror configuration are used to obtain a planar image. The mirrors are connected physically to galvanometers, which are driven by external sources, to provide deflection in the x or y direction [Figure 2.7].

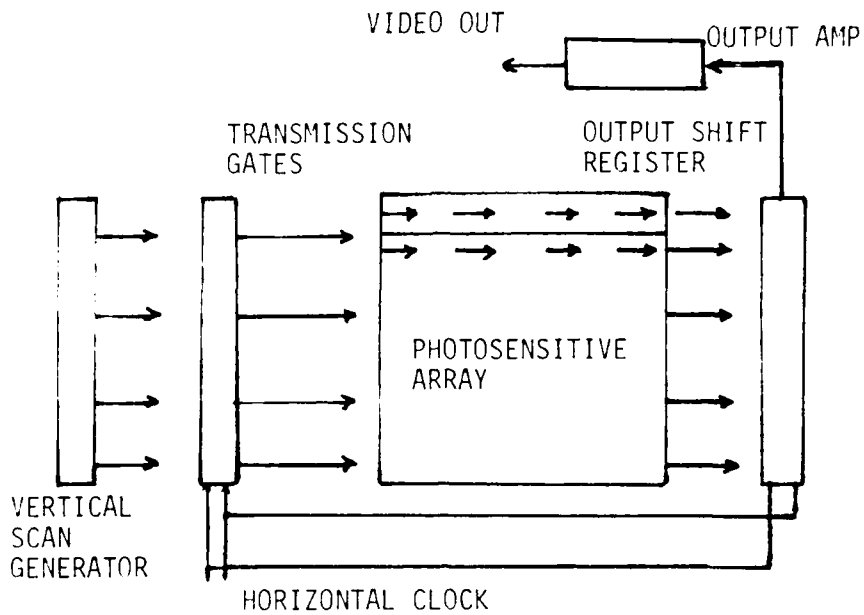


Figure 2.4 CCD Array (Line Transfer) [Ballard and Brown, 1982]

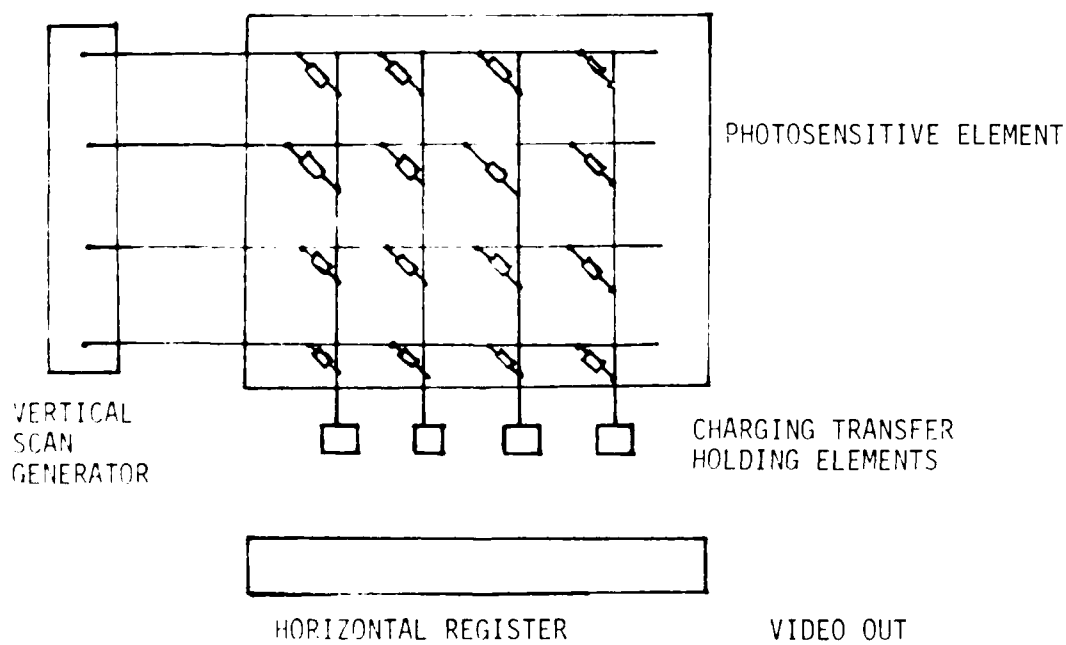


Figure 2.5 CID Array [Ballard and Brown, 1982]

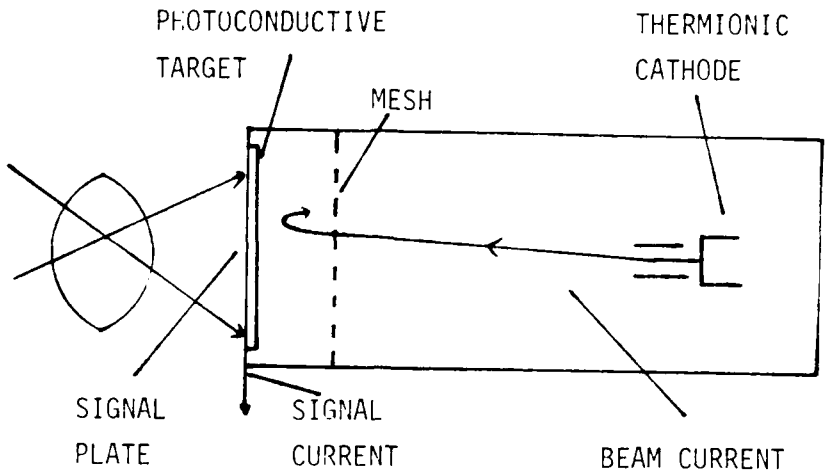


Figure 2.3 The Vidicon Camera [Kunt, 1980]

Solid state arrays: As a result of present technology there are a number of solid state devices used for image formation [Figures 2.4 and 2.5].

Charged coupled devices (CCD) are specially manufactured semi-conductor transistors which are photosensitive. When placed in an array they resemble what amounts to be a "bucket brigade" of charge in a shift register. Through a series of clocking pulses the charges of a specific depletion area are presented to the output.

Charge injection devices (CID) are similar to a CCD array, however a charge is not transferred to the output. The specific transistor must be addressed in order for the data to be read.

current at the rear of the tube is the difference between the scanning current and the locally absorbed current.

Image dissector tube: Here light from an object is focused on a photocathode which converts photon energy into electron flow. Only those electrons emitted from a specific area of the cathode are deflected into an aperture and reach a photo multiplier. The spectral range of the image dissector is from ultraviolet to infrared (Figure 2.2).

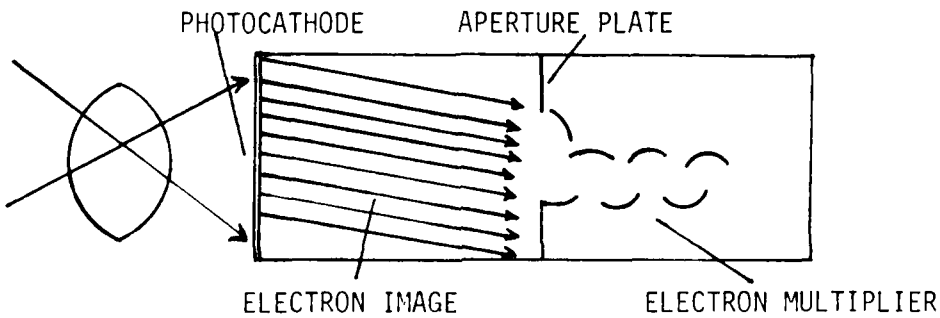


Figure 2.2 Image Dissector Camera [Kunt, 1980]

The vidicon: This device converts the energy of incoming photons to an electron flow through a photoconductive target. The target is coated with an emulsion whose resistance decreases when illuminated. An electron beam is scanned on the back side of this target. Due to the capacitance the charges are held by the target. The result is a capacitive current which is used as the video signal [Figure 2.3].

ation of these two. It allows for the object to be illuminated by a moving spot and sampled through a moving aperture. Such a system is obviously complex and so has had limited use [Castleman, 1979].

Some of the popular electronic scanning devices are described below.

Image orthicon camera: This system (Figure 2.1) has several sections. Light from the scene is focused onto a photocathode. The

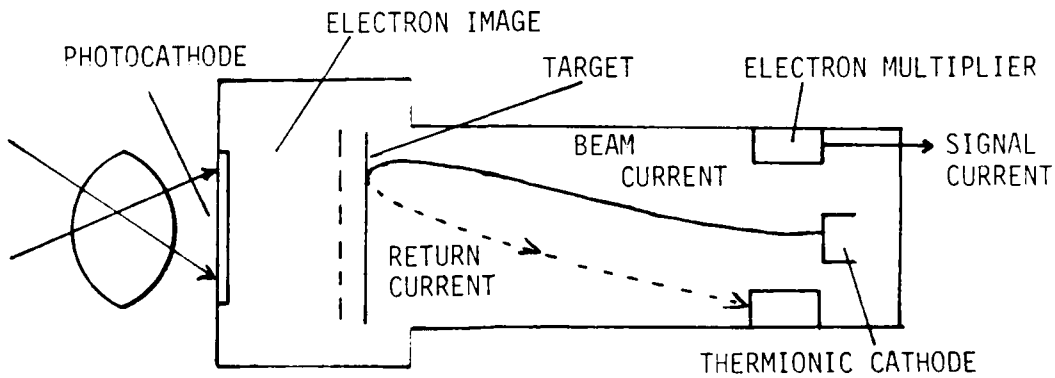


Figure 2.1 Image Orthicon Camera [Kunt,1980]

photocathode in turn emits a number of electrons proportional to the light intensity toward a positively charged target. This second target, which consists of a thin glass disk with a wire grid facing the photocathode, will produce a secondary emission when struck by the electrons. As a result, a positive charge begins to build up on the photocathode side of the second target. The back side of the disk is continuously scanned by a moving electron beam. The disk absorbs these electrons in a neutralizing process. The areas of high intensity will use a large number of electrons to neutralize this part of the target. The output

3. An output device that displays either an output image or the results of the computer processing.

From the description of the first device it is apparent that its purpose is to take an analog real world signal and hold it in a recognizable form to be processed by a computer. This initial device is known as an image digitizer. There are several key elements in the digitizer. First, it must possess some type of transducer elements which are sensitive to the energy given off by the object or scene. Secondly, it must be able to quantize the incoming information into a form capable of being processed. Thirdly, it must have the ability to isolate the energy at particular positions of the scene to facilitate investigations on individual picture elements known as pixels.

It is through certain characteristics that one is able to catagorize a particular digitizer. In Ballard and Browns' text [Ballard and Brown, 1982] three such characteristics are presented:

1) Size of the sampling aperture - the basic spatial resolution of one pixel.

2) Gray Level resolution - normally expressed as the number of quantization levels between black and white.

3) Scanning technique - the particular technique for collecting the light energy. There are basically three catagories of scanning techniques. One is the "scan-out" method where the entire object is illuminated continuously and the sampling aperture is positioned so that the energy from only one pixel is received by the sensor. The "scan-in" method, where only one area of the object is illuminated and all the energy is collected by the sensor. Finally a third method is a combin-

CHAPTER II

IMAGE ACQUISITION AND PROCESSING

Digital image processing is an area of study that cannot be placed under any one scientific or engineering discipline. The definitions range from simply "...the manipulation of images by computer..." [Castleman, 1979] to more informative statements such as "...the construction of explicit, meaningful descriptions of physical objects from images...." [Ballard and Brown, 1982]. It is obvious that within this second quote lies the reason for most of the activity in this fairly new field. Computer vision, for the purposes of this thesis, is a term used synonymously with digital image processing. It has proved to be a very useful tool for learning additional information about the world in which we live. The early applications of this field which included space exploration and x-ray technology have opened doors to other fields including robotics, pattern recognition, image graphics and so on. This chapter presents certain facts describing techniques and equipment used today in the field of computer vision. After this presentation the techniques used in the study will be described.

Computer Vision Systems

There are basically three fundamental parts of any computer vision system:

1. A device which will take and store an image.
2. The digital computer which performs some type of processing.

In Chapter II of this thesis a brief introduction to specific aspects of computer vision is presented. The elements of a general image acquisition system are discussed. The specific hardware of the system used in this research is briefly described. Finally, the image processing techniques peculiar to this system are explained.

In Chapter III a discussion of robot configurations and kinematics is presented. A description is given of some specific details of the general system. The transformation from the attached camera coordinates to the manipulator coordinates is derived. This chapter ends with a discussion of the specific estimation technique used in the example.

In Chapter IV the system performance is evaluated by means of three experiments which were designed to quantify imaging error, positioning error and total system error. Also introduced is a simple model using a Kalman filter that will serve as a basis for examining the experimental data of the system.

In Chapter V is a presentation of the conclusions of this research. Finally, a discussion of specific improvements on the system is presented along with an indication of future research areas that are possible with this system.

graphical look at the major components of the system. The four components are:

1) The Supervisory Computer receives data from both the image acquisition system and the robotic front end processor and controller. Through the use of a high level language the supervisory computer can examine the information concerning the existing environment and then direct a movement of the robotic manipulator in response. The response is delivered to the robotic front end processor and controller by means of a serial communication line.

2) The Front End Processor and Controller performs two main functions. The first is a command function which interprets the information on the serial line as either data or instructions to be accomplished. The second function is that of control whereby the actual robotic manipulator's trajectory and velocity are specified.

3) The Image Acquisition System provides digital images of the area directly below the end-effector of the robot. These images basically can act as a window on this area which when examined in sequence provide dynamic information about this particular scene.

4) The Robotic Manipulator is the structure that is being directed by the system.

The example used to test the robotic system developed in this thesis is a simple linear tracker. The change in the environment that is being observed is a moving object. The system monitors this object and makes an estimate about the object's position at some point in the future. Once the future position is calculated the system derives the necessary information to direct the robotic manipulator to intercept this object at the future point in time.

With the increased demand for intelligent robots the applications will most certainly bring them out of the industrial area into new expanding fields such as space exploration and deep sea mining. Possibly any area that might present a hazard to man while necessitating a decision being made will offer new areas of use for the intelligent robot.

It turns out that "computer vision, the collection of techniques... for obtaining measurements and inferences from images, appears to offer the richest source of sensory information for intelligent robotic manipulation in the greatest number of environments" [Hall et al, 1982]. The need for vision-equipped robots is seen when estimations are presented that a quarter of all industrial robots will be equipped with some form of vision system [Miller, 1984].

In this research an attempt at establishing an intelligent robot system is carried out. The block diagram seen in Figure 1.1 reveals a

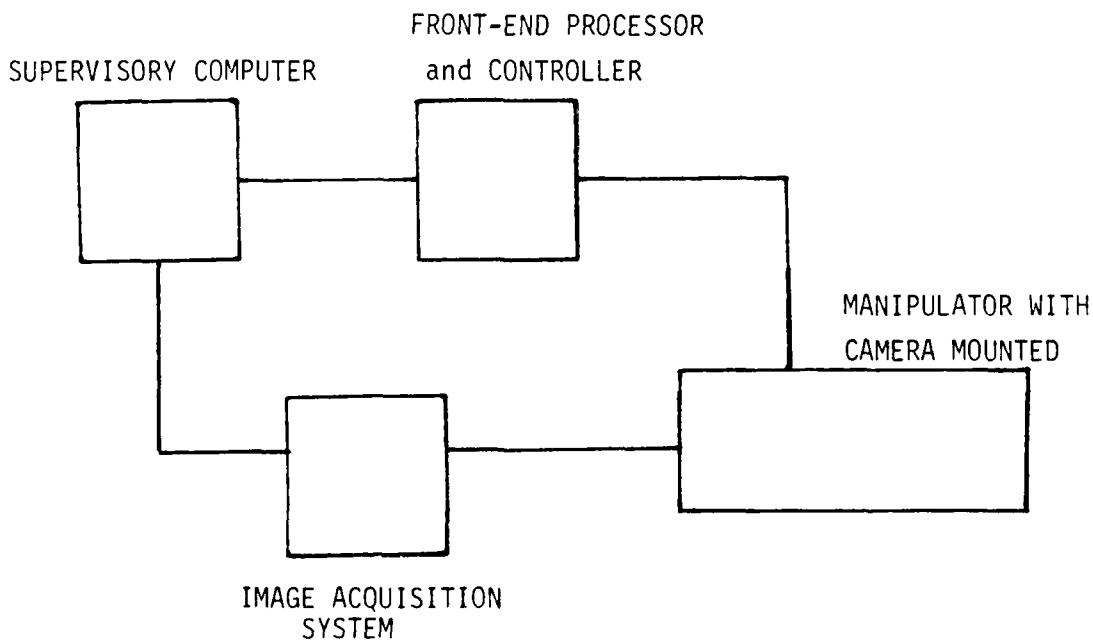


Figure 1.1 System Block Diagram

These reasons, and additionally the fact that numerous other tasks could benefit, have led to the continuing development of intelligent robots. Research in the areas of transportation, bin-picking, assembly, and quality control has been on-going for the last few years. The problems involved with transporting items from place to place in a "blind" workplace have been stated already. The ability of an intelligent robot to overcome these problems could help to avoid production loss due to down time, damage to equipment and most importantly, injuries to human operators.

Bin-Picking is a term that is applied in robotics to a situation where a number of objects are in close proximity and the system is able to delineate a particular object and then to maneuver the manipulator in such a way as to grasp and transport the object.

Assembly is a term applied to a technique whereby a robot can grasp several objects in a specific order and then orient them with reference to each other in some predetermined fashion. For example, putting a nut onto the end of a bolt or a peg into a hole. These common tasks should not be belittled because of the complexities involved in directing such manipulations. A simple assembly task may involve several very difficult subtasks, such as, pattern recognition, bin-picking, and properly aligning the individual objects.

Finally, Quality Control, which is a repetitive examination of manufactured parts seems to be an excellent application for an intelligent robot equipped with a vision capability, thus freeing the operator for a more responsible position with much more satisfaction. This task may run the gamut from inspection of integrated circuit boards to paint jobs on the body of a new automobile.

if an obstacle [illegible] in the predetermined path of the manipulator, there is no way to sense the problem or to avoid the collision. With these "non-intelligent" robots the orientation and initial position of the manipulated object, the path of movement and the final position must be accomplished to very close tolerances.

It is obvious that a large number of tasks exist that are particularly suited to application of so-called "intelligent" robots. These are robots equipped with transducers so that changes in the environment can be detected that may cause the system to falter. These robots should be equipped to compensate for these contingencies.

Some authors are proposing that future industries will strive to improve the organization of the workplace. This movement is working to relieve any possibility that changes in the environment detrimental to system performance occur. This trend is in response to the fact that for non-intelligent robots all positions and orientations must be repetitive. Simple tasks such as moving objects from place to place presuppose the ability of the robot to grasp the object properly each time. Of course this ability is dependent on the position and orientation of the object. The proponents of this trend are advocating that storage structures for objects be carriers and pallets rather than bins. At the present time much research is devoted to designing devices that will perform the positioning and orientation tasks on objects of all sizes. There are a number of drawbacks that are evident in such a trend. First there is a manufacturing and design cost for these new devices. Second, if an item is modified in any way the orientation device must also be modified. Third,to provide orientation and position of parts may not be the most economical way to store them. Finally, it seems an impossible task to provide for all possible contingencies in any environment.

occur at the end of each horizontal scan when the camera's scanning system must be moved back from right to left. At this time, a horizontal blanking signal is generated by the camera so that no information is acquired during this retrace. In addition a horizontal sync pulse is generated at these times. Basically, the presence of the synthetic signals provide the timing information for the monitor to lock-in on the active video signal. There is also a vertical blanking signal that marks the time for the scanning system to reset itself to the top-left position. Because of the time involved with this vertical retrace only about 480 out of 525 horizontal scan lines provide information to the monitor.

Static Scene Analysis

At this point we are concerned with the data as it exists in the acquisition system. It can be viewed as an array of discrete picture elements, whose numerical value is an indication of its shading or gray level. The array of pixels, therefore, taken as a whole is a discrete approximation of the initial scene or object. Through the use of hardware and software algorithms the computer vision system "...extracts pertinent information from the image data. This module in essence performs the task of isolating 'interesting' areas for further analysis" [Computer, Vol. 13, 1980].

It is this idea that leads to the next area of discussion. What types of information processing can be performed on these distinct arrays? It is this extraction of data from the frame of pixels that is given the title of scene analysis. The additional feature of temporal information extraction is placed under the title of dynamic scene

analysis. There are a number of papers that give a good idea of this lively field of investigation which are listed in the reference area at the end of this thesis.

The authors of several articles use different terminology to delineate the methods of data extraction. However, the methods themselves are common to a majority of the authors.

The first method of interest is concerned with improvement of image clarity. This area of study is known as image enhancement. Histogram transformation is a popular approach to image enhancement. As the name implies the technique involves manipulating the gray level histogram of an image [Hummel, 1975]. A gray level histogram is simply a graphical representation displaying the frequency of occurrence of the individual gray-levels within a particular image. There are several ways to use the histogram for enhancement. The most popular technique is through thresholding the grey-level histogram by making a subjective decision about the threshold value which will highlight particular objects. For example, a common approach is to form a binary image. This is done by changing all gray levels below a specific threshold to black and those above to white.

The second technique is known as histogram equalization. This method tries to stretch the initial histogram consisting of n gray levels into a new histogram of p gray levels ($p>n$). The result of this technique improves contrast and therefore facilitates individual object recognition [Hummel, 1975].

The third technique used to manipulate the histogram is in histogram hyperbolization. This method tries to transform the histogram of displayed brightness levels by producing a uniform distribution of perceived

brightness levels. "All pictures processed in this way have been consistently considered of superior intelligibility than their histogram equalized counterparts." [Frei, 1977].

Along with enhancement there are a number of types of image processing techniques available to extract spacial properties from a scene. The first method here is called template matching where a pixel-by-pixel comparison is performed on one (usually "live") image with another (stored) image to be used as a reference. At times an operator known as a "template" is used to extract or detect a particular sub-image. The template is placed at several offsets on the initial image and a correlation is performed. The point of maximum match is determined to be the sub-image under investigation [Computer, 1980].

Several spatial properties can be extracted by the second method of segmenting the image into meaningful entities. This is usually performed by the approach known as edge detection. An edge is defined as an area in an image where local gray levels are changing rapidly. Through the use of an edge operator the presence of this edge can be detected. "The unifying feature of ... edge operators is that they compute a direction ... of maximum gray-level change and a magnitude describing the severity of this change" [Ballard and Brown, 1983]. Figure 2.8 is retrieved from Robinson's paper on gradient masks [Robinson, 1971] and presents several common edge operators and their directional sensitivity. The major problem with these simple operators is that they also respond to noise areas within the initial image. As a result a technique called edge relaxation is employed to improve the edge operator measurement by basing some measurements on the existence of neighboring edges. For example, the existence of two strong edges within the vicinity of a

Direction of Edge	Prewitt Masks	Kirsch Masks	Three-level Simple Masks	Five-level Simple Masks
North	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$
Northwest	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix}$
West	$\begin{bmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{bmatrix}$	$\begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$
Southwest	$\begin{bmatrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix}$
South	$\begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$
Southeast	$\begin{bmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}$
East	$\begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$
Northeast	$\begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}$

Figure 2.8 Examples of Compass Gradient Masks

relatively weak edge may give more credence to the existence of this weak edge rather then to the possibility that it is just a noise pattern in that area. The next step in this technique would be to somehow group these edges into logical objects. This technique therefore assumes that there is enough background information to specify what object a group of edges represents.

Another technique for segmentation of an image tries to overcome this problem. The method is known as region growing. It initially divides the image into basic areas either by grouping identical pixel values or simply by dividing the image into small regions. These distinct groupings or regions are then merged together on the basis of "similarity", where this criterion would be different for each system. There are several problems encountered with this technique. "Problems can arise in the selection of initial regions, and in selecting the merging criteria" [Ohlander et al., 1978].

All the methods mentioned thus far have been concerned with manipulating static frames in order to derive specific data with which to identify the existence of objects, to find their dimensions, and possibly to classify them based on some criterion. By adding the dimension of time we enter the field of dynamic scene analysis which obviously yields additional information. This new information holds the possibility for a number of interesting applications.

Dynamic Scene Analysis

In looking at the dynamic case it is easy to concur with Martin and Aggarwal in recognizing that "...a 'dynamic image' is a sequence of static images ... with a given or assumed time function relating the order

and elapsed interval between elements of the sequence" [Aggarwal and Martin, 1978]. Obviously when the time factor is introduced we can view groups of images which have similarities as well as differences. The authors proceed to demonstrate the idea that "... a dynamic image analysis system must be able to separate the constancies from the changes, and be able to separate the interesting changes from the noisy ones" [Aggarwal and Martin, 1978].

Within the area of dynamic scene analysis the idea of motion detection is one of the most researched. As the name implies this study deals with the ability to recognize and specify spacial changes by studying objects in motion. In a number of techniques motion detection is achieved, but the ability to gain information about specific features is lost. The earliest research in this area dealt with the detection and measurement of cloud motion from satellite photographs [Leese et al, 1970]. One of the approaches in these studies is to divide an initial image into sections and then to correlate these sections with related areas in the following image. The maximum cross-correlation coefficient is interpreted as a match for that section. As a result, the centers of the two sections are connected with a motion vector. The second technique that seems to be popular is the binary thresholding technique. With this method an image is divided into two gray levels. The dividing point is chosen so that the boundary of cloud formation is evident. The next step is to match each cloud formation to a formation in the following image. Obviously the drawbacks to these two techniques are that they assign a motion vector to a section, not to any feature within the section.

Another technique that is popular for an indication of change between two images is a simple subtraction technique. If two images have been aligned and one image is subtracted, pixel by pixel, from the other, a resulting image will yield gray levels in areas where there are changes. Again this technique, though straightforward, has some inherent problems. The features within the area of change are still not specified, and the presence of uncorrelated noise between images may appear as meaningful changes. When using any of the methods described one must remain aware of the limitations of the individual techniques and try to compensate for them in the particular applications.

Equipment Used in the Study

In performing the image processing, Image Technology Inc.'s digital picture acquisition system was used in conjunction with a PDP 11/60

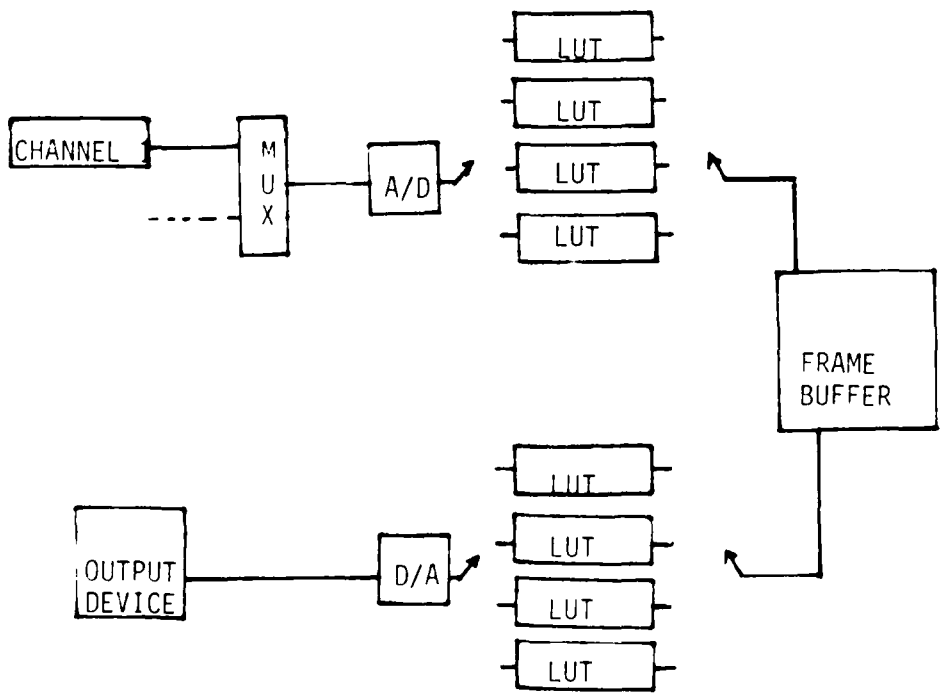


Figure 2.9 Image Technology, Inc. Hardware

digital computer. Figure 2.9 shows a graphical representation of how the system is constructed. The system as depicted has the ability of processing information from two cameras. The multiplexer merely selects the appropriate channel. The equipment digitizes the video signal in real time. This real time capability is due to the use of TRW TDC100TJ flash analog-to-digital converters.

The pixel's digitized value, which ranges from 0 to 255, acts as a pointer to a particular input look-up table (LUT). In the address of the LUT will be stored a value (from 0 to 255) which represents a specific gray level between black (0) and white (255). This gray level value is then stored as the picture element in the image frame buffer memory.

The image frame buffer memory consists of 256K bytes of dynamic ram constructed as a 512 x 512 array. The FB-512 board, as the manufacturer calls it, is capable of driving 3 analog signals (R (red), B (blue), G (green)). From the figure the output value from the frame buffer again acts as a pointer to an output look-up table. The particular address of this LUT contains a value from 0 to 255 which represents the gray level for the output representation. The final step then is to convert the digital value of gray level to an analog signal. The equipment here is the TRW TDC1016J-8 highspeed digital-to-analog converter. The signal is then processed by additional circuitry to form a composite video signal suitable for a standard TV monitor.

The control of this hardware is performed through a number of MACRO subroutines and functions. The ability to call these subroutines and functions from high-level languages makes the hardware functions almost invisible to the user. There is issued a short pamphlet entitled <u>IMAGING - Basic Driver Programmer's Manual</u> which explains the use of the different subroutines and functions.

Processing Techniques Employed

By the appropriate use of the system hardware and software it was possible to avoid many of the problems presented in the references. Since this research was not primarily concerned with image processing or motion detection, it was decided to simplify all aspects of object recognition. As a result, the number of objects in the field is limited to one. Also the object of interest is a symmetrical shape (circle). These two restrictions are used so that estimation of the object's center could be facilitated. It was intended to establish motion detection by specifying a change of position for the center of the object. Furthermore, by appropriately setting the input LUTs to produce a binary image, a high contrast between object and background is assured.

The purpose behind using the image acquisition system is to extract information about the movement of an object. The information should provide enough data to make an estimate of the object's position at some advanced point in time.

Image Technology's system allows for a rather intricate use of the frame buffer. The technique is called "zooming." Zooming enables a user to select an active video window within the frame buffer where image acquisition can be held to a local region. The user has the option to "zoom" in either the x or y direction or in both. The result is that the dimension of the applicable direction is reduced by one-half. The location of the active video window is chosen by panning the upper left corner of the window to a specific x location and then scrolling of this point to a specific y location [Figure 2.10]. The active video window may therefore take on any one of four two dimensional sizes.

a) 512 x 512 NO ZOOMING

b) 256 x 512 ZOOM ONLY IN X-DIRECTION

c) 512 x 256 ZOOM ONLY IN Y-DIRECTION

d) 256 x 256 ZOOM IN X- AND Y-DIRECTIONS

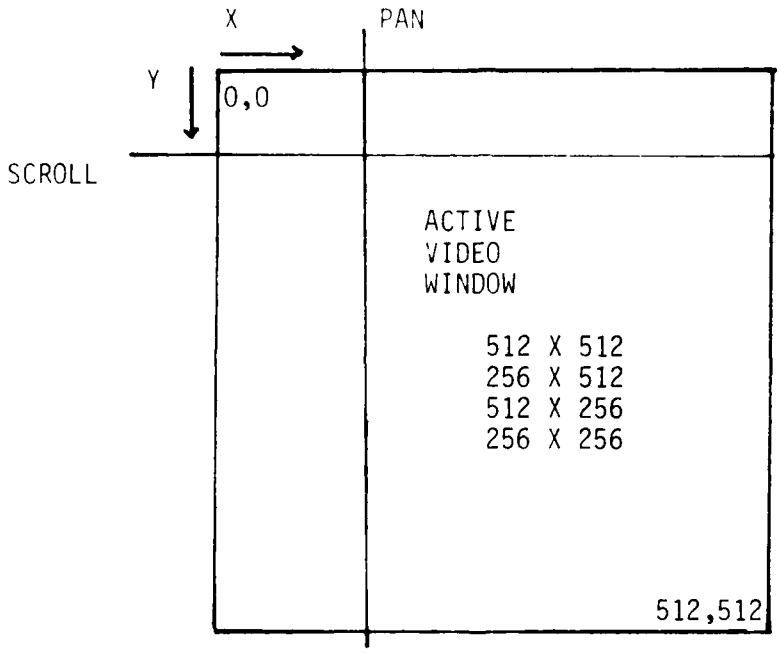


Figure 2.10 512 x 512 Frame Buffer

The method incorporated in this study is to zoom in both the x-and y-directions. This technique allows for the storage of several images in the frame buffer simultaneously. This technique allows for the processing of images to be carried out at the same time. Three images are taken at .25 seconds apart. Therefore, by examining the location of the center of the objects in each image an estimate of both velocity and direction can be extracted [Figure 2.11].

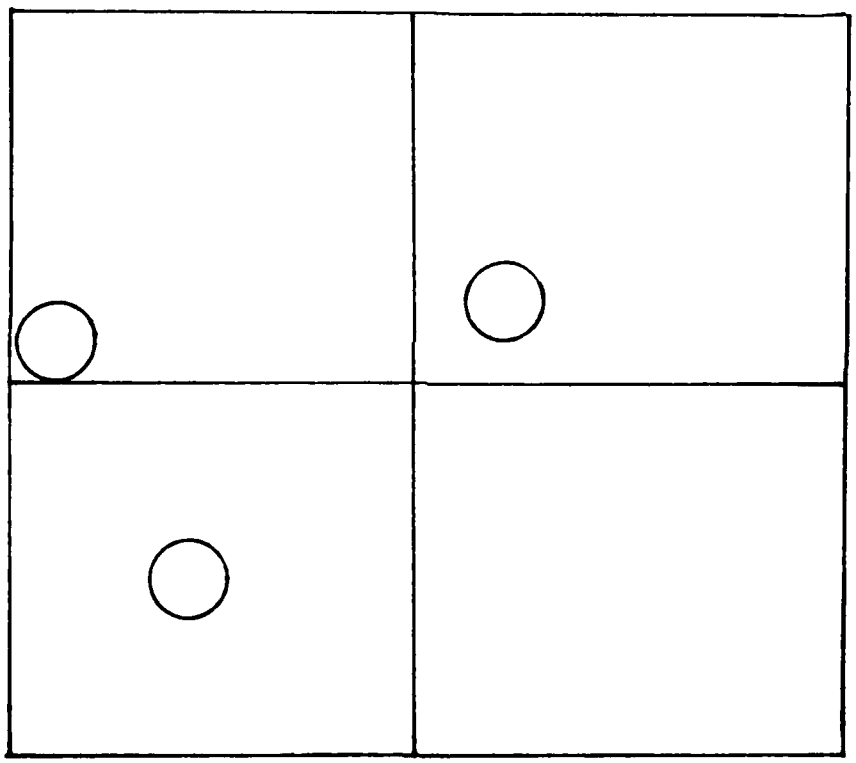


Figure 2.11 512 x 512 Frame Buffer with Three Images

During the initial stages of research two techniques were investigated to determine the objects' centers. The first of the techniques employed is known as the turtle boundary detector/follower [Duda, 1973].

The second method is a very simple approach that can be called the cross-hairs approach.

Both methods require finding an object's boundary point as a starting location. To avoid searching each entire section (256 x 256), a method was devised to conduct the search in a grid-like approach. That is the search was conducted by examining every fifth line of the y direction starting from the top of each section and working down.

The "turtle" method was derived to examine the boundary of an object located in a binary image. By keeping track of the x coordinates of the boundary pixels, y coordinates of the boundary pixels, and the total number of pixels in the boundary, the centroid of the object may be found [Dubois, 1984]. The turtle method consists of moving a "turtle" around the boundary of an image in the direction determined by whether the current pixel is an object pixel or background pixel. If the turtle is located on an object pixel, it will advance by making a left-turn as referenced to its last movement. If it is located on a background pixel, it will advance by making a right-turn as referenced to its last movement [Figure 2.12]. By referring to the figure, it is obvious that several pixels may be entered more then once. Software must insure that duplicate information is avoided. The object's centroid specified by an x and y coordinate is derived through two simple equations.

$$x_{center} = \frac{\sum_i x_i}{\sum_i i} \qquad y_{center} = \frac{\sum_i y_i}{\sum_i i} \qquad \text{where } \sum_i i = \text{number of boundary pixels.}$$

The most striking problem in using the "turtle" is its sensitivity to noise. In using this technique the investigation examined a black object on a white background. The noise was visible in both the background

and the object images. Several techniques were tried in order to reduce this sensitivity. The first attempt consisted of subjectively determining a threshold value that would reduce the background noise to a tolerable level. This attempt merely reaffirmed the fact that the turtle method, as presented, was unable to cope with any background noise. The second attempt was based on an assumption that the large number of white background pixels might be causing saturation in the automatic gain control of the vidicon camera. This attempt consisted of using the imaging system to produce a negative image of the scene. That is, pixels that were white would be black and those that were black would be white. This technique also did not reduce the noise as expected. The third attempt used a white object on a black background to test if the noise was contributed by the system hardware. Again the results revealed that the noise was present at an unacceptable level. At the same time these tests were being performed, the second method was used and performing well in the noisy environment. As a result this second method was chosen for continued tests.

The cross-hairs method was basically designed for use with symmetrical objects. The technique examines the horizontal chord of the object at the boundary point. Because of the symmetry of the object, the center of the chord is chosen as the initial x center for the object. The next step is to examine the diameter of the object in the y-direction passing through the x center location. Due to symmetry the middle of the diameter is chosen as the initial y center for the object.

Due to noise and inaccuracies in the system it was decided to extract additional information in determining the center coordinates. This was accomplished by examining points along the cross-hairs of the

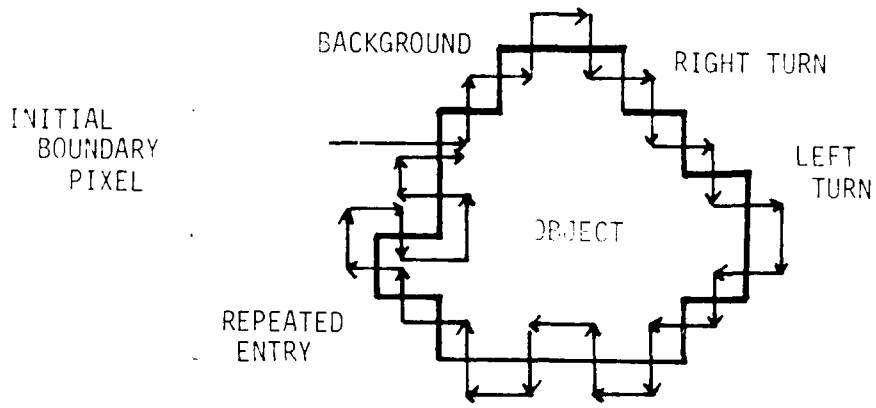
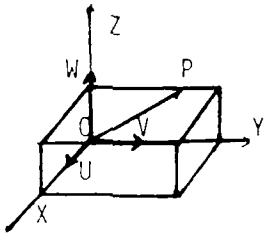


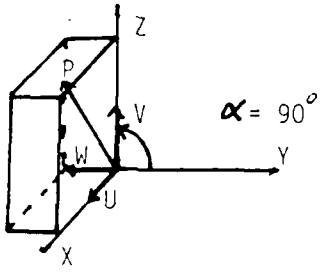
Figure 2.12 Turtle Movement

object. After the initial center coordinates were located the next step was to move $\pm$ 1/4 diameter in the x direction. At this point two chords were examined to find their midpoint in the y-direction. Finally an average of the three values was taken and used as the final y center coordinates. A similar procedure was used to determine the x center coordinate [Figure 2.13].

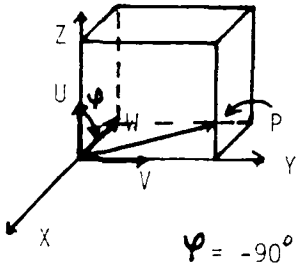
This technique works in a noisy environment by simply setting a limit on the minimum size object. That is, if the initial chord or diameter was less than four pixels long, the object was discarded as a noise pattern and the search continued.



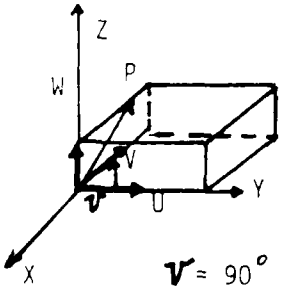
$$T = \begin{bmatrix} \bar{1} & 0 & 0 & \bar{0} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \underline{0} & 0 & 0 & \underline{1} \end{bmatrix}$$



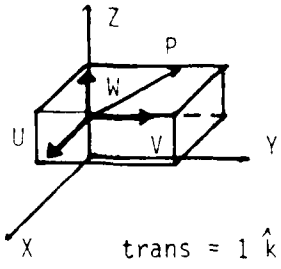
$$T = \begin{bmatrix} \bar{1} & 0 & 0 & \bar{0} \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ \underline{0} & 0 & 0 & \underline{1} \end{bmatrix}$$



$$T = \begin{bmatrix} \bar{0} & 0 & -1 & \bar{0} \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ \underline{0} & 0 & 0 & \underline{1} \end{bmatrix}$$



$$T = \begin{bmatrix} \bar{0} & -1 & 0 & \bar{0} \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \underline{0} & 0 & 0 & \underline{1} \end{bmatrix}$$



$$T = \begin{bmatrix} \bar{1} & 0 & 0 & \bar{0} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ \underline{0} & 0 & 0 & \underline{1} \end{bmatrix}$$

Figure 3.9 Pure Rotation and Translation Matrices

The conclusion is that the position and orientation of the manipulator can be expressed as the product of a translation matrix and an orientation matrix.

In the kinematics used for this thesis the positioning matrix was the Cartesian coordinate translation matrix and the orientation matrix specified by the $\hat{a}$, $\hat{o}$ and $\hat{n}$ unit vectors.

The general form of this transformation matrix is given by:

$$T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ \underline{0} & 0 & 0 & 1 \end{bmatrix}$$

Figure 3.9 shows five specific examples of the transformation matrix for various positions of a body-attached coordinate system in a reference frame. A particular column of the transformation matrix expresses the orientation of the corresponding body-attached coordinate axis in terms of the reference coordinate axes in the form $(\hat{i}_x, \hat{i}_y, \hat{i}_z, 0)^t$. That is, if we look at the first column of each matrix in the figure, the orientation of axis OU in the reference coordinate system OXYZ can easily be seen. Likewise, the axes OV's and OW's orientation can be determined by observing the 2nd and 3rd column vector of the T matrix. The 4th column matrix locates the origin of the revolving coordinate system in the reference coordinate system in the form $(\hat{i}_x, \hat{i}_y, \hat{i}_z, 1)^t$. Similar information about the orientation and location of the fixed reference system in terms of the rotating coordinate system can be obtained by merely taking the inverse of the T matrix:

about the z axis and finally a translation z along the z axis. Therefore the matrix representing such a position is given by:

$$\text{Cyl}\ (z,\alpha,r) = \text{Trans}\ (0,0,z)\ \text{Rot}(z,\alpha)\ \text{Trans}\ (r,0,0)$$

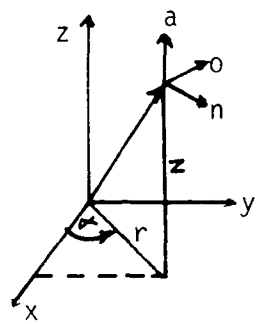


Figure 3.7 Cylindrical Polar Coordinates [Paul, 1981]

In Figure 3.8 the specification through spherical coordinates, r, β, and α corresponds to a translation r along the z axis, followed by a rotation β about the y axis and finally a rotation α about the z axis. In this case the position can be described as:

$$\text{Sph}\ (\alpha,\beta,r) = \text{Rot}\ (z,\alpha)\ \text{Rot}\ (y,\beta)\ \text{Trans}\ (0,0,r)$$

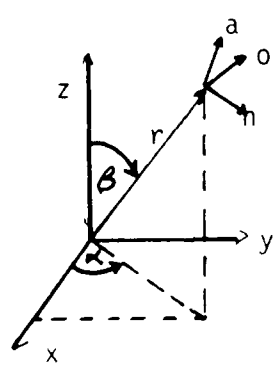


Figure 3.8 Spherical Polar Coordinates [Paul, 1981]

$$\text{Euler}(\phi,\theta,\psi) = \text{Rot}(z,\phi)\ \text{Rot}(y,\theta)\ \text{Rot}(z,\psi)$$

The third method for expressing the orientation of the end-effector is given in terms of the roll, pitch, and yaw angles, terms commonly used in speaking about ships. Figure 3.6 depicts the situation corres-

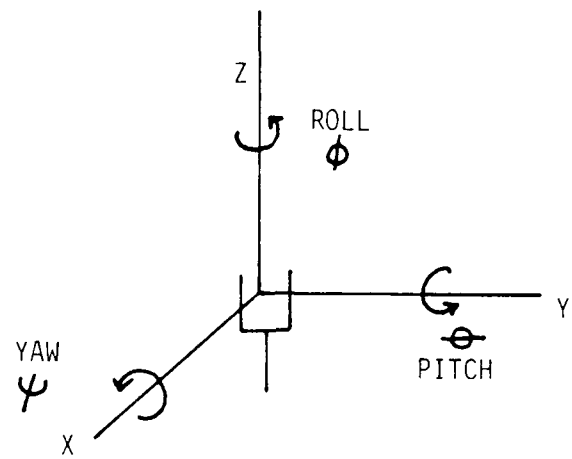


Figure 3.6 Roll, Pitch, and Yaw Coordinates for a Manipulator

ponding to a rotation ϕ about the z axis (roll), then a rotation θ about the y axis (pitch) and a rotation ψ about the x axis (yaw). The general rotation matrix is given as:

$$\text{RPY}(\phi,\theta,\psi) = \text{Rot}(z,\phi)\ \text{Rot}(y,\theta)\ \text{Rot}\ (x,\psi)$$

In addition to specifying the orientation of the coordinate frame of the end-effector, a position for the origin of this frame must be established. In the first method a vector p was introduced to specify this position in the base coordinates. This position can also be described in both cylindrical coordinates and spherical coordinates. In Figure 3.7 the specification through cylindrical coordinates r, α, and z corresponds to a translation r along the x axis, followed by a rotation α

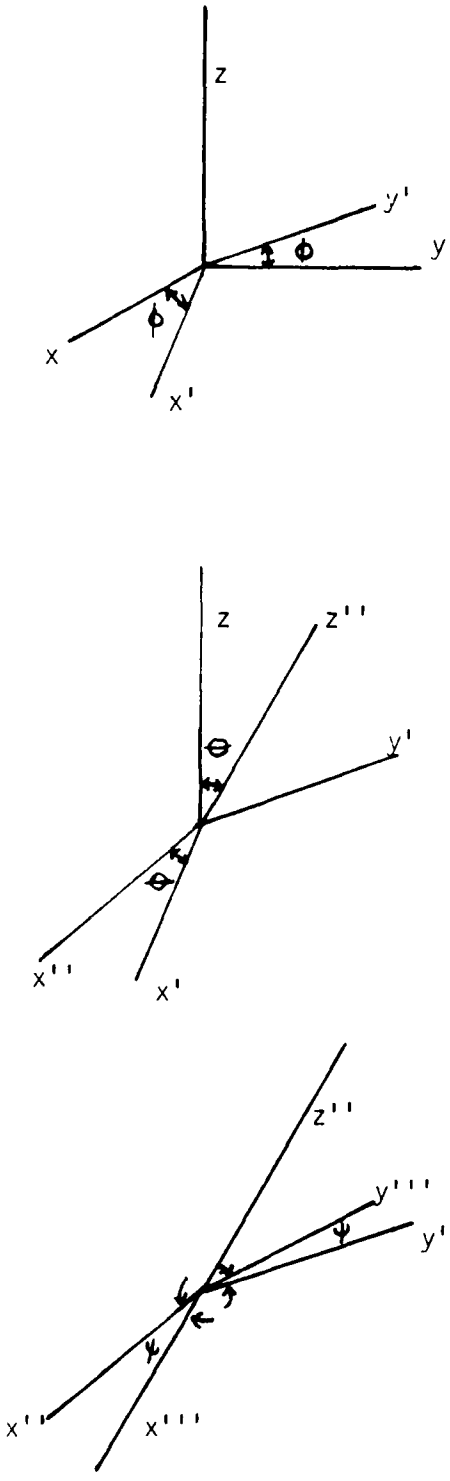


Figure 3.5 Euler Angles

The orientation and position of this hand can be described by an attached coordinate frame whose origin is located at the midpoint between the two fingers. This origin is also described by a vector p whose coordinates are expressed in the defining reference frame. There are three unit vectors which describe the orientation of the hand. The approach vector, $\hat{a}$, is defined as the direction from which the hand would approach an object. The orientation vector, $\hat{o}$, is in the direction specifying the orientation of the hand, from fingertip to fingertip. The normal vector, $\hat{n}$, will complete the right handed coordinate system and is specified by crossing the $\hat{o}$ vector into the $\hat{a}$ vector [Paul, 1981].

There are a number of ways of specifying the orientation of a manipulator or in this case the end-effector. It is obvious that within the general matrix there are only a few values that afford any information. For example the bottom row will be three zeros and a one depending on whether scaling and perspective become involved.

The first method presented in the text [Paul, 1981] is by specifying the three vectors $\hat{a}$, $\hat{o}$ and $\hat{n}$ discussed before. These three vectors specify the orientation of a coordinate frame whose position can be defined in a number of ways to be discussed later. The constraints on this method are simply that the $\hat{a}$, $\hat{o}$ and $\hat{n}$ vectors are of unit magnitude, and the p vector describes a location that can be reached by the manipulator.

The first method can be viewed as a Cartesian approach because orientation is expressed as distances along these three axes. The second and third methods are expressed as a set of rotations. The second method, using a set of Euler angles, can describe any orientation in terms of a rotation ψ about the z axis, then a rotation θ about the new y axis y', and finally a rotation ϕ about the new z axis z" [Figure 3.5]. In this case the general rotation matrix can be expressed as:

$$T_{tran} = \begin{vmatrix} \bar{1} & 0 & 0 & d\bar{x} \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ \underline{0} & 0 & 0 & 1 \end{vmatrix}$$

In deriving kinematics equations for any robot the main consideration is that of orienting and positioning the end-effector of the robot. For the purposes of this thesis the end-effector is the fifth link of the robotic manipulator. Therefore in specifying the individual elements of the general transformation matrix ${}^{0}T_{n}$ much more information is established than merely transforming the coordinates of a vector expressed in the final link's coordinate system into the base coordinates.

One method of interpreting the general matrix is by specifying three vectors $\hat{n}$, $\hat{o}$, and $\hat{a}$ as shown in Figure 3.4, which represents the end effector of a robot.

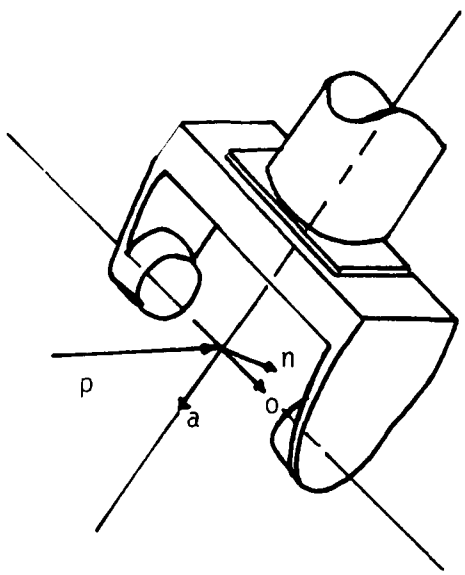


Figure 3.4 Orientation Vectors Associated with Manipulator [Paul, 1981]

The homogeneous transformation matrix takes the form:

$$T = \begin{bmatrix} R_{3x3} & P_{3x1} \\ f_{1x3} & 1x1 \end{bmatrix} = \begin{array}{cc} \text{Rotation matrix} & \text{Translation Position Vector} \\ \hline \text{Perspective Transf.} & \text{Scaling Factor} \end{array}$$

Therefore the homogeneous transformation matrices for pure rotation are:

$$T_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{y,\beta} = \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{z,\gamma} = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 & 0 \\ \sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The homogeneous matrix for pure translation transforms the origin of the body attached coordinate system which has been translated (no rotation) into the reference coordinate system. The basic homogeneous translation matrix takes the form:

an angle $\alpha = 90°$. This transformation can be expressed as:

$$p_{xyz} = R_{x,\alpha}\, p_{uvw}$$

Therefore this rotational matrix can be expressed as:

$$R_{x,\alpha} = \begin{bmatrix} \hat{i}_x \cdot \hat{i}_u & \hat{i}_x \cdot \hat{j}_v & \hat{i}_x \cdot \hat{k}_w \\ \hat{j}_y \cdot \hat{i}_u & \hat{j}_y \cdot \hat{j}_v & \hat{j}_y \cdot \hat{k}_w \\ \hat{k}_z \cdot \hat{i}_w & \hat{k}_z \cdot \hat{j}_v & \hat{k}_z \cdot \hat{k}_w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

Similarly for rotations about the axes OY and OZ we can find rotation matrices $R_{y,\theta}$ and $R_{z,\nu}$ respectively. These can be expressed as

$$R_{y,\theta} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \qquad R_{z,\nu} = \begin{bmatrix} \cos\nu & -\sin\nu & 0 \\ \sin\nu & \cos\nu & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

By knowing these 3x3 matrices we can decompose complex rotations into their basic rotation matrices and derive the complex rotational matrix R.

$$R = R_{x,\alpha} \cdot R_{y,\theta} \cdot R_{z,\nu}$$

By adding an additional coordinate to the position vector

$$p_{xyz} = (wp_x,\ wp_y,\ wp_z, w)^t$$

the position of the point is expressed in <u>homogeneous coordinates</u>. As a result, the 4x4 homogeneous transformation matrix now has the capability of expressing coordinate system rotation, translation, scaling, and perspective. The homogeneous matrix is composed of the original rotation matrix in addition to 3 new components:

- 3x1 position vector
- 1x3 perspective vector
- 1x1 scaling factor

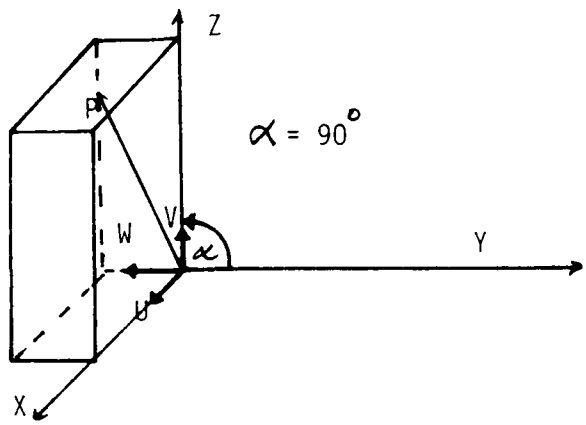


Figure 3.3 Rotation of the Body Attached Frame

$$p_x = \hat{i}_x \cdot p_{uvw} = \hat{i}_x \cdot \hat{i}_u p_u + \hat{i}_x \cdot \hat{j}_v p_v + \hat{i}_x \cdot \hat{k}_w p_w$$

$$p_y = \hat{i}_y \cdot p_{uvw} = \hat{j}_y \cdot \hat{i}_u p_u + \hat{j}_y \cdot \hat{j}_v p_v + \hat{j}_y \cdot \hat{k}_w p_w$$

$$p_z = \hat{i}_z \cdot p_{uvw} = \hat{k}_z \cdot \hat{i}_u p_u + \hat{k}_z \cdot \hat{j}_v p_v + \hat{k}_z \cdot \hat{k}_w p_w$$

These three equations can be expressed in matrix form:

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} \hat{i}_x \cdot \hat{i}_u & \hat{i}_x \cdot \hat{j}_v & \hat{i}_x \cdot \hat{k}_w \\ \hat{j}_y \cdot \hat{i}_u & \hat{j}_y \cdot \hat{j}_v & \hat{j}_y \cdot \hat{k}_v \\ \hat{k}_z \cdot \hat{i}_u & \hat{k}_z \cdot \hat{j}_v & \hat{k}_z \cdot \hat{k}_v \end{bmatrix} \begin{bmatrix} p_u \\ p_v \\ p_w \end{bmatrix}$$

and so the transformation matrix A that relates the body attached coordinate system to the reference coordinate system has been determined. Likewise there exist a 3x3 matrix B that transforms a vector in the OXYZ coordinate system into the OUVW coordinate system:

$$p_{uvw} = B\ p_{xyz}$$

Since any complex rotation can be divided into three component rotations, the next step in developing the forward kinematics is to derive the basic rotation matrices for rotation about the three axes of the reference frame. Figure 3.3 shows a rotation about the OX axis by

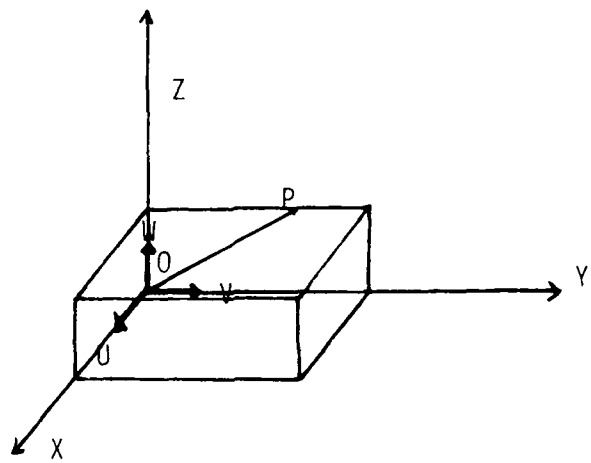


Figure 3.2 Body Attached Coordinate Frame and Reference Frame

Obviously point p can be specified in either of the coordinate systems, that is:

$$p_{uvw} = (p_u, p_v, p_w)^t$$
$$p_{xyz} = (p_x, p_y, p_z)^t$$

If the block is rotated by some arbitrary angle, the point p, which is fixed to the block, is also rotated [Figure 3.3]. That is, the vector $(p_x, p_y, p_z)^t$ has changed while the vector $(p_u, p_v, p_w)^t$ has remained constant.

A rotation matrix (3x3) can be constructed that maps the coordinates of a position vector in a rotated coordinate system (OUVW) into a reference coordinate system (OXYZ) [Lee, 1982]. Therefore the vector $p_{uvw} = p_u \hat{i}_u + p_v \hat{i}_v + p_w \hat{i}_w$, when projected onto the coordinate frame OXYZ axes, will yield:

In order to perform even the smallest movement of a robotic arm (manipulator), there must be specific changes in the angles or displacements that exist between the set of links and joints. The manipulator's position and orientation in space may be specified by examining the position, orientation, and dimensions of each link.

The kinematics problem is generally divided into two parts:

- forward kinematics is concerned with the position and orientation of the manipulator given a set of joint angles.
- inverse kinematics is concerned with deriving a legitimate set of joint angles given a position and orientation of the end-effector of a robot in some reference coordinate system.

Forward Kinematics

In viewing the forward kinematics problem it is necessary to investigate the relationship between a stationary or reference frame and a coordinate system that is able to revolve and/or translate. Lee [Lee, 1982] discusses this topic using a rigid body example, where there is a body-attached coordinate frame on a block located in a reference coordinate system [Figure 3.2].

In this figure we have two right-handed rectangular coordinate systems. OUVW is a body-attached coordinate system that will change position and orientation as the rigid-body does so. OXYZ, on the other hand, is a fixed reference frame. The purpose here is to develop a given transformation matrix from the OUVW coordinate system to the OXYZ coordinate system.

CHAPTER III

ROBOTIC MANIPULATOR KINEMATICS

In this chapter robot configurations and kinematics are reviewed and a description is given of the specific robot manipulator used in this study and of the computer network providing control for that manipulator. Then a description of the system equations used in the particular estimation problem which illustrates the integration of the vision acquisition system with the mechanical manipulator arm is presented.

Kinematics is basically a description of the geometries associated with a mechanical system. Many industrial robots of today consist of N+1 rigid bodies called links and N connections known as joints. There are two types of joints, translational and revolute, whose names indicate the types of motions the joints perform [Figure 3.1].

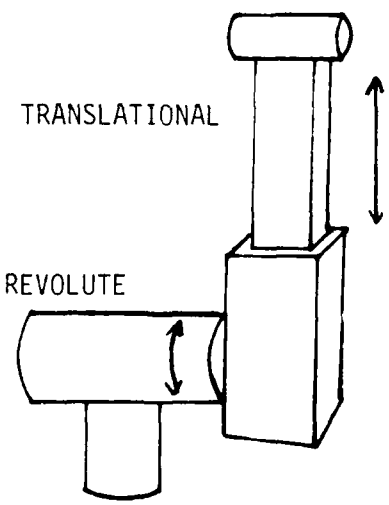


Figure 3.1 Two Types of Robotic Joints

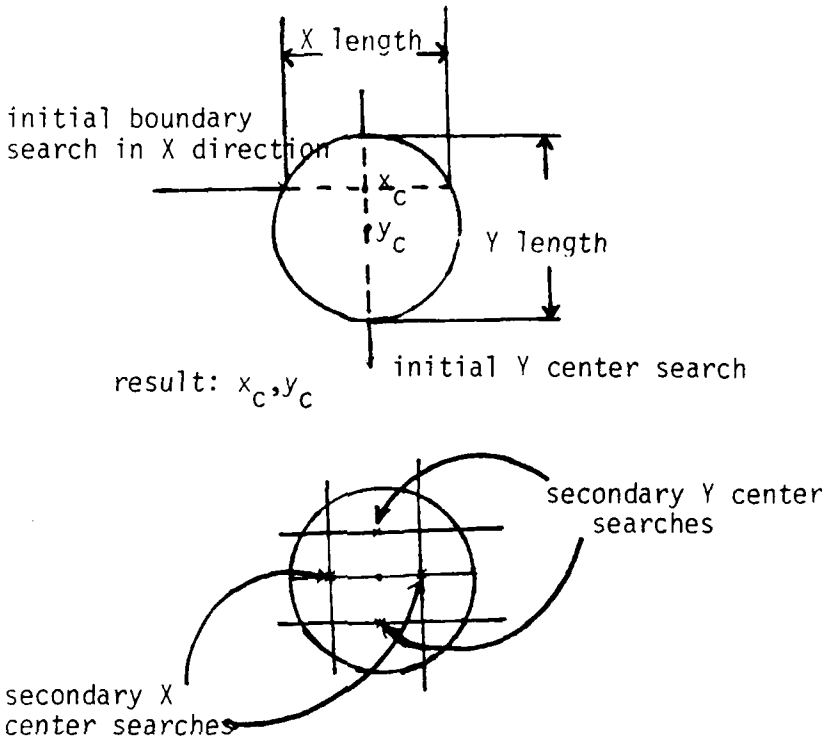


Figure 2.13 Cross Hairs Method

$$T^{-1} = \begin{bmatrix} \bar{n}_x & n_y & n_z & -n\cdot\bar{p} \\ o_x & o_y & o_z & -o\cdot p \\ a_x & a_y & a_z & -a\cdot p \\ \underline{0} & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \bar{\quad} & & & -n\cdot\bar{p} \\ R^t_{3x3} & & & -o\cdot p \\ & & & -a\cdot p \\ \underline{0} & 0 & 0 & 1 \end{bmatrix}$$

To further understand the derivation of the forward kinematics problem the idea of coordinate system transformation is applied to a robotic arm. Because the links of the robot can rotate and/or translate with respect to a reference coordinate frame, a body-attached coordinate system is established at the joint for each link [Lee, 1982].

The establishment of these body-attached coordinate frames were initially described by J. Denavit and R.S. Hartenberg [Denavit and Hartenberg, 1955]. Their method is outlined in [Paul, 1981] and [Lee, 1982]. The general principles consist of three rules (i refers to the ith link, x,y,z are coordinate directions):

1) the z_i axis lies along the axis of motion of the (i+1)th joint.

2) the x_i axis is along the axis resulting from the cross-product of the z_{i-1} and z_i axes,

$$x_i = \pm (z_{i-1} X z_i)/\|z_{i-1} X z_i\|$$

or along the common normal between the z_i and z_{i-1} axes. The y_i axis is chosen to complete the right-handed coordinate system.

3) The origin of the ith coordinate frame lies either at the intersection of the z_{i-1} and z_i axes or at the intersection of their common normal with the z_i axis.

Denavit and Hartenberg also present four additional quantities that completely describe any revolute or translational joint. It is important to recognize that the links maintain a fixed relationship between the joints [Paul, 1981]. The four parameters are:

θ_i - the joint angle from the x_{i-1} axis to the x_i axis about the z_{i-1} axis

d_i - the distance from the origin of the (i-1)th coordinate frame to the intersection of z_{i-1} axis with the x_i axis along the z_{i-1} axis.

a_i - the offset distance from the intersection of the z_{i-1} axis with the x_i axis to the origin of the ith system along the x_i axis (or the shortest distance between the z_{i-1} and z_i axes).

α_i - the offset angle from the z_{i-1} axis to the z_i axis about the x_i axis (using the right-hand rule) [Lee, 1982].

Depending on the type of joint some of the quantities are constants. That is, for a revolute joint the values of d_i, a_i and α_i remain constant. Since θ_i changes, it is given the term "joint variable." While with a translational joint the values of a_i, α_i and θ_i are constant and d_i becomes the joint variable [Figure 3.10].

Once the coordinate system for each link has been established, a homogeneous transformation matrix relating the coordinate system of link i to that of link (i-1) can be found. That is, the orientation of the ith coordinate system in the (i-1)th system can be reduced to four basic transformations:

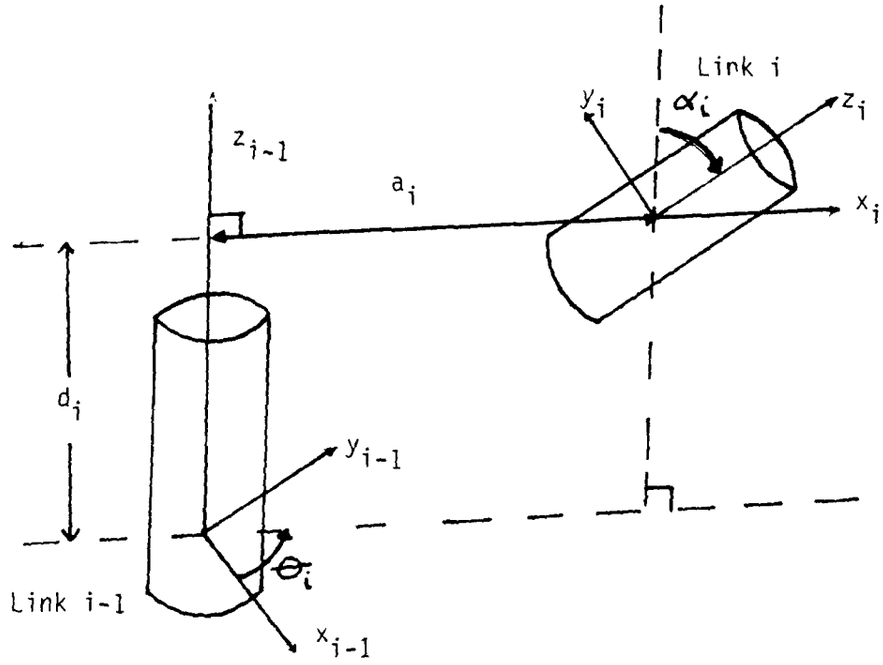


Figure 3.10 Joint Variables and Quantities

1) rotation about z_{i-1} axis an angle θ_i to align the x_{i-1} axis with x_i axis.
2) translation along the z_{i-1} axis a distance d_i so that x_{i-1} and x_i become coincident.
3) translation along the x_i axis a distance a_i to bring the origin of both coordinate systems into coincidence.
4) rotation about the x_i axis an angle α_i so that z_{i-1} axis and z_i axis are aligned.

As a result the homogeneous transformation matrix relating any ith coordinate system to the (i-1)th coordinate system would be the matrix product of these four basic transformations:

$$A_{i-1}^{i} = T_{z,\theta}\, T_{z,d}\, T_{x,a}\, T_{x,\alpha}$$

For revolute joints, the resulting matrix takes the form

$$\begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and for translational joints, it becomes

$$\begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & 0 \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Forward Kinematic for P-5 Robot

The robot used in this thesis research was the General Electric P-5 process robot. A graphical depiction of the robot can be found in Figure 3.1. Using the Denavit and Hartenburg method for establishing coordinate frames, and both Paul and Lee's descriptions on how the homogeneous matrix between joints and coordinate frames can be established, the following description of the forward kinematics of the P-5 was arrived at.

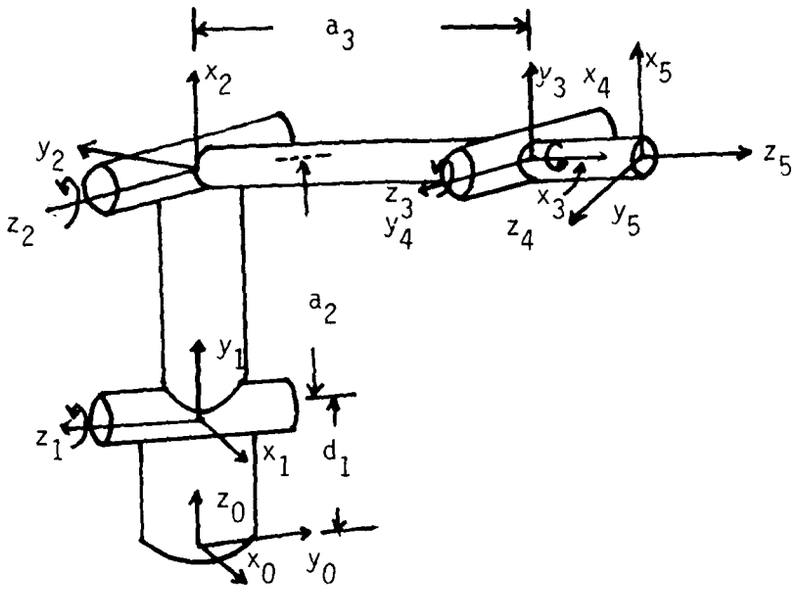


Figure 3.11 Five Link Robot Similar to the P-5 Process Robot

In Table 3.1 the necessary link parameters of the P-5 process robot are tabulated.

Table 3.1 P-5 Robot Joint Parameters

Link	Joint variable	Home position θ_i =	Constant parameters a	α	d
1	θ_1	0°	0	90°	65cm
2	θ_2	90°	60cm	0°	0
3	θ_3	-90°	80cm	0°	0
4	θ_4	90°	0	90°	0
5	θ_5	0°	0	0°	10cm

In order to establish the position of a point in coordinate system i referred to the base coordinate system we must develop the homogeneous transformation matrix ${}^{0}T_i$. This is given by the product of the individual transformation matrices $A_0^1 A_1^2 \cdots A_{i-1}^i$.

The matrix A_0^1 as was stated in the previous section is a specific application of the general matrix A_i^{i-1} whose form was given previously. For the P-5 robot with the notation $\cos\theta_i = C_i$ and $\sin\theta_i = S_i$, the individual A matrices take the form:

$$A_0^1 = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad . \quad \begin{matrix} d_1 = 65 \text{ cm} & a = 0.0 \text{ cm} \\ \alpha_1 = 90^\circ & \end{matrix}$$

$$A_1^2 = \begin{bmatrix} C_2 & -S_2 & 0 & a_2C_2 \\ S_2 & C_2 & 0 & a_2S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad . \quad \begin{matrix} d_2 = 0^\circ & a_2 = 60 \text{ cm} \\ \alpha_2 = 0^\circ & \end{matrix}$$

$$A_2^3 = \begin{bmatrix} C_3 & -S_3 & 0 & a_3C_3 \\ S_3 & C_3 & 0 & a_3S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad . \quad \begin{matrix} d_3 = 0 \text{ cm} & a_3 = 80 \text{ cm} \\ \alpha_3 = 0^\circ & \end{matrix}$$

$$A_3^4 = \begin{bmatrix} C_4 & 0 & S_4 & 0 \\ S_4 & 0 & -C_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad . \quad \begin{matrix} d = 0 \text{ cm} & a_4 = 0 \text{ cm} \\ \alpha_4 = 90^\circ & \end{matrix}$$

$$A_4^5 = \begin{bmatrix} C_5 & -S_5 & 0 & 0 \\ S_5 & C_5 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad . \quad \begin{matrix} d = 10 \text{ cm} & a = 0 \text{ cm} \\ \alpha = 0^\circ & \end{matrix}$$

Now the coordinates of any link relative to some reference coordinate frame can be expressed as an appropriate product of these matrices.

For example, if one wishes to express some point in the coordinate system attached to link 5 in terms of the reference coordinate system of link 4 we merely perform the matrix multiplication expressed by:

$$A_3^4\ A_4^5\ p_5$$

$$= \begin{bmatrix} \bar{C}_4 & 0 & S_4 & \bar{0} \\ S_4 & 0 & -C_4 & 0 \\ 0 & 1 & 0 & 0 \\ \underline{0} & 0 & 0 & \underline{1} \end{bmatrix} \begin{bmatrix} \bar{C}_5 & S_5 & 0 & 0 \\ S_5 & C_5 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ \underline{0} & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \bar{C}_4C_5 & -C_4S_5 & S_4 & d_5\bar{S_4} \\ S_4C_5 & -S_4S_5 & -C_4 & -d_5C_4 \\ S_5 & C_5 & 0 & 0 \\ \underline{0} & 0 & 0 & \underline{1} \end{bmatrix}$$

Now suppose one wished to express a point in the coordinate system attached to link 5 in the base coordinate system. This can be expressed as:

$$p_o = {}^oT_5\ p_5 = A_o^1\ A_1^2\ A_2^3\ A_3^4\ A_4^5\ p_5$$

where

$${}^oT_5 \doteq A_o^1\ A_1^2\ A_2^3\ A_3^4\ A_4^5$$

The general transformation matrix takes the form

$${}^oT_5 = \begin{bmatrix} C_1C_{234}C_5+S_1S_5 & -C_1C_{234}S_5+S_1C_5 & C_1S_{234} & C_1[d_5S_{234}+a_3C_{23}+a_2C_2] \\ S_1C_{234}C_5-C_1S_5 & -S_1C_{234}S_5-C_1C_5 & S_1S_{234} & S_1[d_5S_{234}+a_3C_{23}+a_2C_2] \\ S_{234}C_5 & -S_{234}S_5 & -C_{234} & -d_5C_{234}+a_3S_{23}+a_2S_3+d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where $\sin(\theta_i + \theta_j + \theta_k) = S_{ijk}$

$\cos(\theta_i + \theta_j + \theta_k) = C_{ijk}$

Therefore with the general matrix transformation ${}^{0}T_5$ any position of the end effector expressed in the fifth coordinate system can easily be transformed to base coordinates.

Inverse Kinematics

The inverse kinematic's problem may be the more important problem in this discussion. The reason for this is simply that most robotic arms or manipulators are positioned and oriented by a combination of joint angles. This gives importance to the ability to transform real world coordinates into this particular set of joint angles. "Obtaining a solution for the joint coordinates requires intuition and is the most difficult problem we will encounter ..." [Paul, 1981].

Paul presents a general approach to the solution along with some problems that may be avoided. He also suggests that since the end effector of an n-link manipulator has a general transformation matrix

$$ {}^{0}T_n = A_0^1 \, A_1^2 \cdots A_{n-2}^{n-1} \, A_{n-1}^{n} $$

by simply recursively premultiplying the matrix by the inverse of a link transformation the specifications for the necessary joint angles can be obtained. To facilitate understanding of the technique it will help to introduce Paul's discussion of Euler angles and their solution [Paul, 1981].

Euler angles can describe any possible orientation of the end effector in terms of a rotation ϕ about the z axis, then a rotation θ about the new y axis y' and finally a rotation ψ about the new z axis z". As previously discussed, the orientation of the end effector can be expressed as a product of 3 rotation matrices which becomes:

$$\text{Euler}(\phi,\theta,\psi)=\begin{bmatrix} \cos\phi\cos\theta\cos\psi - \sin\phi\sin\psi & -\cos\phi\cos\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\sin\theta & 0 \\ \sin\phi\sin\theta\sin\psi + \cos\phi\sin\psi & -\sin\phi\cos\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\sin\theta & 0 \\ -\sin\theta\cos\psi & \sin\theta\sin\psi & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Since this should yield the homogeneous transformation matrix that was presented in the previous discussion, it must be true that:

$$\text{Euler}(\phi,\theta,\psi) = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

As a result, the unknown Euler angles should be determinable from this matrix equation. The obvious solutions from these equalities are:

$$\theta = \cos^{-1}(a_z)$$

$$\phi = \cos^{-1}(ax/\sin\theta)$$

$$\psi = \cos^{-1}(-n_z/\sin\theta)$$

As Paul points out there are several problems with this approach.

1) In using the arc cosine function:

   a) the sign is undefined

   b) the accuracy in determining the angle itself is dependent on the angle

2) In solving for θ and ψ:

a) problems with the arc cosine as discussed above

b) use of sine leads to inaccuracy whenever $\sin\theta \approx 0$

c) values undefined when $\theta = 0$ or $\pm$ 180°

Paul suggests looking for arctangent functions in doing his iterative approach and goes on to justify his choice of arctangents with the following arguments.

a) angles are returned in range $-\pi < \theta < \pi$ by taking into account the signs of the ordinate and abscissa

b) ability exists to detect zero ordinate or abscissa and return proper result

c) accuracy for arctangent is uniform over full range of definition.

Inverse Kinematics for the P5 Robot

For the application of this technique to the P-5 robot the ${}^{0}T_5$ matrix can be expressed as:

$$ {}^{0}T_5 = A_0^1\, A_1^2\, A_2^3\, A_3^4\, A_4^5 $$

The first step in the application of this technique to the P-5 robot is then to premultiply the ${}^{0}T_5$ matrix by $(A_0^1)^{-1}$.

$$ (A_0^1)^{-1}\, {}^{0}T_5 = {}^{1}T_5 = \begin{bmatrix} C_1 n_x + S_1 n_y & C_1 o_x + S_1 o_y & C_1 a_x + S_1 a_y & C_1 p_x + S_1 p_y \\ n_z & o_z & a_z & p_z - d_1 \\ S_1 n_x - C_1 n_y & S_1 o_x - C_1 o_y & S_1 a_x - C_1 a_y & S_1 p_x - C_1 p_y \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

which should also be equivalent to:

$$
{}^{1}T_5 = \begin{vmatrix} C_{234}C_5 & -C_{234}S_5 & S_{234} & d_5S_{234}+a_3C_{23}+a_2C_2 \\ S_{234}C_5 & -S_{234}S_5 & -C_{234} & -d_5C_{234}+a_3S_{23}+a_2S_3 \\ S_5 & C_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} .
$$

To be equivalent each term of the matrix must be equal to the corresponding term. By equating the third elements of columm four, θ_1 can be found:

$$S_1p_x - C_1p_y = 0$$

$$\tan\theta_1 = \frac{p_y}{p_x}$$

$$\theta_1 = \tan^{-1}\frac{p_y}{p_x}$$

The next angle of interest is θ_5. By equating the third elements of the 1st and 2nd columns θ_5 can be found:

$$S_5 = S_1n_x - C_1n_y$$

$$C_5 = S_1o_x - C_1o_y$$

$\theta_5 = \tan^{-1}\left(\frac{S_1n_x - C_1n_y}{S_1o_x - C_1o_y}\right)$. Next θ_{234} can be determined by equating the 1st and 2nd elements of the third columns.

$$S_{234} = C_1a_x + S_1a_y$$

$$-C_{234} = a_z$$

$$\theta_{234} = \tan^{-1}\left(\frac{(-C_1a_x - S_1a_y)}{a_z}\right)$$

The solutions for $\theta_2, \theta_3, \theta_4$ can be found using a trigonometric technique. After equating appropriate terms the identities of interest are:

$$C_1p_x + S_1p_y = S_{234}d_5 + C_{23}a_3 + C_2a_2$$

$$p_z - d_1 = -C_{234}d_5 + C_{23}a_3 + S_2a_2$$

By manipulating the equations and defining two intermediate values it can be shown that:

$$q = C_1 p_x + S_1 p_y - S_{234} d_5 = C_{23} a_3 + C_2 a_2$$

$$r = p_z - d_1 + C_{234} d_5 = S_{23} a_3 + S_2 a_2$$

$$q^2 = C_{23}^2 a_3^2 + 2C_2 C_{23} a_2 a_3 + C_2^2 a_2^2$$

$$r^2 = S_{23}^2 a_3^2 + 2S_2 S_{23} a_2 a_3 + S_2^2 a_2^2$$

$$q^2 + r^2 = (C_{23}^2 + S_{23}^2) a_3^2 + 2a_2 a_3 (C_2 C_{23} + S_2 S_{23}) + a_2^2 (C_2^2 + S_2^2)$$

$$= a_3^2 + 2a_2 a_3 (C_2 C_{23} + S_2 S_{23}) + a_2^2$$

Now by examining the middle term, it can be shown that:

$$C_2 C_{23} + S_2 S_{23} = \cos(\theta_{23} - \theta_2) = \cos\theta_3 = C_3$$

$$q^2 + r^2 = a_3^2 + a_2^2 + 2a_2 a_3 C_3$$

$$C_3 = \frac{q^2 + r^2 - a_3^2 - a_2^2}{2a_2 a_3}$$

$$S_3 = \pm\sqrt{1 - C_3^2} \qquad \leftarrow \text{yields 2 values}$$

$$\theta_3 = \tan^{-1}(S_3/C_3)$$

$$q = C_{23} a_3 + C_2 a_2 = C_2 C_3 a_3 - S_2 S_3 a_3 + C_2 a_2$$

$$r = S_{23} a_3 + S_2 A_2 = S_2 C_3 a_3 + C_2 S_3 a_3 + S_2 a_2$$

$$q = S_2 (-S_3 a_3) + C_2 (C_3 a_3 + a_2)$$

$$r = S_2 (C_3 a_3 + a_2) + C_2 (S_3 a_3)$$

Cramer's rule is then exployed in finding solutions for S_2 and C_2:

$$= \begin{vmatrix} -S_3a_3 & C_3a_3 + a_2 \\ C_3a_3+a_2 & S_3a_3 \end{vmatrix} = S_3^2a_3^2 - (C_3a_3 + a_2)^2$$

$$= -S_3^2a_3^2 - [C_3^2a_3^2 + 2a_3C_3a_2 + a_2^2]$$

$$= -a_3^2 - 2a_2a_3C_3 - a_2^2$$

$$= -(a_3^2 + 2C_3a_2a_3 + a_2^2)$$

$$S_2 = \begin{vmatrix} q & C_3a_3 + a_2 \\ r & S_3a_3 \end{vmatrix} = qS_3a_3 - r(C_3a_3 + a_2)$$

$$S_2 = \frac{S_2}{} = \frac{qS_3a_3 - r(C_3a_3 + a_2)}{-(a_3^2 + 2C_3a_2a_3+a_2^2)}$$

$$C_2 = \begin{vmatrix} -S_3a_3 & q \\ C_3a_3+a_2 & r \end{vmatrix} = -rS_3a_3 - q(C_3a_3 + a_2)$$

$$C_2 = \frac{C_2}{} = -\frac{(rS_3a_3 + q(C_3a_3 + a_2))}{-(a_3^2 + 2C_3a_2a_3+a_2^2)}$$

$$_2 = \tan^{-1}\left(\frac{S_2}{C_2}\right)$$

$$_{234} = {}_2 + {}_3 + {}_4$$

$$_4 = {}_{234} - {}_2 - {}_3$$

the vector of joint angles is thus determined.

Within the working system there are two added constraints that have an effect on the specific form of the ${}^0T_5$ matrix. The first is the task

of maintaining the orientation of the fifth link parallel to the floor. It can be accomplished by properly selecting the vectors $\hat{a}$, $\hat{o}$, and $\hat{n}$, which were discussed previously. The set of equations that specify this task is:

$$\frac{p_x}{|p|} = n_x = o_y$$

$$\frac{p_y}{|p|} = n_y = -o_x$$

$$0.0 = n_z = o_z$$

$$\vec{a} = \vec{n} \times \vec{o}$$

The second constraint is that the manipulator is maintained at a fixed height when not in motion. This constraint is realized by passing a constant value for the z parameter to the controller.

Calibration of Camera to Robot

In this project an additional transformation is due to the introduction of the camera into the system. The camera's image field acts as another coordinate frame. As a result of the restrictions in link 5 and the fact that the camera is mounted to a fixed frame [Figure 3.12], the camera lens is maintained parallel to the floor. Therefore, this transformation is only necessary in two dimensions.

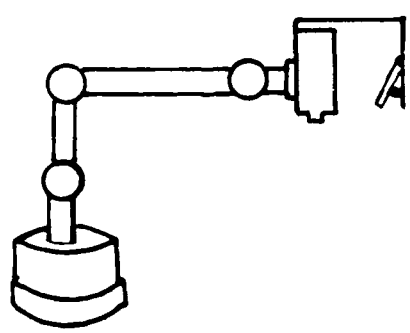


Figure 3.12 P-5 Robot, Vidicon Camera, Laser

deviation are relatively of the same size in the range .5 - 1.5 for the x and y coordinate and also the radial distance.

The next significant source of error was that due to the repeatability of robot motion. That is, what is the standard deviation of the error between a desired position and the actual position, when the robot is repeatedly asked to go to the desired position. In this test the robot was started at the home position. The image of a stationary object was used to direct a motion of the robot. When the move was completed, a new image of the object was taken, and the coordinates of its center were determined. The results of 100 such trials were:

$$\overline{\sigma_x} = 1.98$$

$$\overline{\sigma_y} = 6.12$$

$$\overline{\sigma_r} = 3.59$$

In each case an average value was determined. Then, using this value as the mean, a variance and standard deviation were calculated. In this experiment it is obvious that the repeatability of a move is different in the x and y direction. Because these standard deviations are expressed in pixels the true deviation of the repeated position is rather small. However these pixel values provide a basis for evaluating the overall system error. Without any estimation taking place these values give an indication of how accurately the robot can be placed over a stationary object. It is obvious that in the overall system the fact that a moving object is used and an estimation is attempted should yield higher expected standard deviations.

The time constraint on the minimum estimation time is influenced by two specific times. One was the time used to calculate the object's estimated position, and the second was the time period between the be-

TABLE 4.1 Image Processing Error Data

1A)	$\bar{x} = 99.725001$	$\sigma_x = 1.161368$	1B)	$\bar{x} = 99.690002$	$\sigma_x = 1.111713$
	$\bar{y} = 84.803001$	$\sigma_y = .471371$		$\bar{y} = 84.830002$	$\sigma_y = .501097$
	$\bar{R} = 130.918182$	$\sigma_R = .818020$		$\bar{R} = 130.901230$	$\sigma_R = .769482$
2A)	$\bar{x} = 29.895$	$\sigma_x = .876342$	2B)	$\bar{x} = 29.976999$	$\sigma_x = .877766$
	$\bar{y} = 33.395$	$\sigma_y = .575304$		$\bar{y} = 33.471001$	$\sigma_y = .571981$
	$\bar{R} = 44.830269$	$\sigma_R = .531333$		$\bar{R} = 44.941757$	$\sigma_R = .517673$
3A)	$\bar{x} = 223.826004$	$\sigma_x = 1.010803$	3B)	$\bar{x} = 223.617996$	$\sigma_x = .966475$
	$\bar{y} = 186.552994$	$\sigma_y = .670218$		$\bar{y} = 186.472000$	$\sigma_y = .575512$
	$\bar{R} = 291.377655$	$\sigma_R = .805518$		$\bar{R} = 291.166077$	$\sigma_R = .754707$
4A)	$\bar{x} = 57.59$	$\sigma_x = .813569$	4B)	$\bar{x} = 57.681999$	$\sigma_x = .908227$
	$\bar{y} = 187.220001$	$\sigma_y = .677937$		$\bar{y} = 187.162994$	$\sigma_y = .546288$
	$\bar{R} = 195.878860$	$\sigma_R = .697025$		$\bar{R} = 195.851807$	$\sigma_R = .581503$
5A)	$\bar{x} = 218.020996$	$\sigma_x = .728394$	5B)	$\bar{x} = 218.056000$	$\sigma_x = .904914$
	$\bar{y} = 25.496000$	$\sigma_y = .624487$		$\bar{y} = 25.608999$	$\sigma_y = .588320$
	$\bar{R} = 219.506973$	$\sigma_R = .706355$		$\bar{R} = 219.554825$	$\sigma_R = .873478$

$$\overline{\sigma_x} = \frac{9.359571}{10} = .94$$

$$\overline{\sigma_y} = \frac{5.802515}{10} = .58$$

$$\overline{\sigma_R} = \frac{7.0550913}{10} = .71$$

deviation from this average value was derived. The data derived in the experiment is presented in Table 4.1. The data in this table represents experiments performed at five locations within a zoomed image field. These five locations, labelled 1 through 5 in the table, include the four corners and the center of the zoomed image field in order to detect any variations over the field. Each position is represented by two experiments (A and B). Each block of experimental results consist of the average object center x coordinate and y coordinate. The average radial distance from the image field origin, and the corresponding standard deviations. In addition, the table includes a list of the average standard deviations over all data.

The data in the table reveals some interesting features about the image acquisition system for the static case. The most obvious result is that the standard deviation in the y direction is consistently lower then that of the x direction. At this time there is no obvious reason for this type of bias. Another interesting result that is evident is that the standard deviations of the x and y coordinates and radial distance are on the average less than 1.0 pixel. The data in this experiment typifies the actual data received in the working system. That is the zooming feature of the imaging system has been incorporated so that results of data acquisition within the tracking experiment would be consistent with the data provided here. This simply means that even with the reduced resolution of the object in a zoomed mode the variance from the mean would be less then a pixel in the static case. The third interesting feature revealed by this experiment is that no area of the camera field reveals any true deviation from value of other areas. That is the standard

The dynamics of the P-5 robot seem to be the major source of error. When the robot is directed to proceed to a position, although there is a form of velocity control, it is obvious that overshoot occurs and vibration is present at the conclusion of the move. This vibration may cause slight shifting of the camera or mispositioning of the robot. Both of these problems would result in errors for the image processing application. Another problem introduced by the P-5 robot that may cause inaccuracies in the calculations or even in the actual positioning is the repeatability of the robot. That is, given a position command how accurately can the robot repeat this position?

There are additional considerations which must be taken into account. For example, throughout the research the presence of noise in the image field has been observed. This source of error influenced the choice of the center finding technique as discussed previously. Furthermore, computational and mechanical movement times put limits on the minimum prediction time.

The first source of error of interest is that introduced by the image aquisition system. As was stated previously, the main task of this system is to determine the value for the center of the object that is being tracked. This does not have to be the true centroid, however the value that is determined should be repeatable to some degree of accuracy even in a "noisy" environment. The test to discover the error in this measurement was a static test. Although the motor power to the robot was on, there was no directed movement during the test. The process consisted of placing an object at several locations in the image field and making 1000 measurements at each position. For each position an average x,y, and $(x^2 + y^2)^{1/2}$ value was determined and a standard

CHAPTER IV

EXPERIMENTAL RESULTS AND ANALYSIS

In this chapter a number of experiments and their results used to evaluate the performance of the "intelligent robot" system are presented. At the conclusion of the chapter a Kalman filter model of a linear estimator is given in order to have a basis for evaluating the experimental results obtained.

In order to facilitate the analysis of the data recorded in this research, it will be important to understand in general what the specific sources of errors are. To this end an attempt was made to examine all the contributions to the overall error in terms of camera pixels.

Experiments and Results

The system as described in the previous chapter contains a number of devices that may cause the introduction of errors into the system. Although the software has tried to compensate for image acquisition errors there are a number of possibilities that may arise that could raise the probability of inaccuracies in the image gathering technique. For example, if the robot is vibrating or moving in any way while the images are being taken each image will not be taken at the exact same location. Even in the static case the software may not derive the same center of the object each time due to noise or slight rotation of the object. The improper illumination of the object field may also cause the center of the object to shift.

In addition to this estimation routine for the moving object there is a process in the software that attempts to avoid possible indications of false velocity. If the object is stationary an error in the center finding routine may indicate a change of position and therefore a velocity that is not truly present. As a result, depending on the estimation period, the robot may be directed to an erroneous position. To avoid this problem the software observes very small changes in positions as such, and directs the robot to remain at the present position for a specific sampling period. By storing the robot's previous position in this process a "catch up" maneuver can be made when the object has definitely traversed a specific distance. As a result there may be several sampling periods when the object is moving slowly enough to warrant the robotic manipulator to remain in place. However, even with very slow movement, over a period of time a large enough distance will have been traveled so that the manipulation must be moved just to catch up to the object.

one having x_1 as the initial position, the other having x_2 as the initial position. In a similar manner the estimated position $\hat{y}_3$ is found. The equations are:

$$\hat{x}_3 = \frac{1}{2}(x_1 + 2\dot{x}_1\Delta T + x_2 + \dot{x}_1\Delta T)$$
$$\hat{y}_3 = \frac{1}{2}(y_1 + 2\dot{y}_1\Delta T + y_2 + \dot{y}_1\Delta T)$$

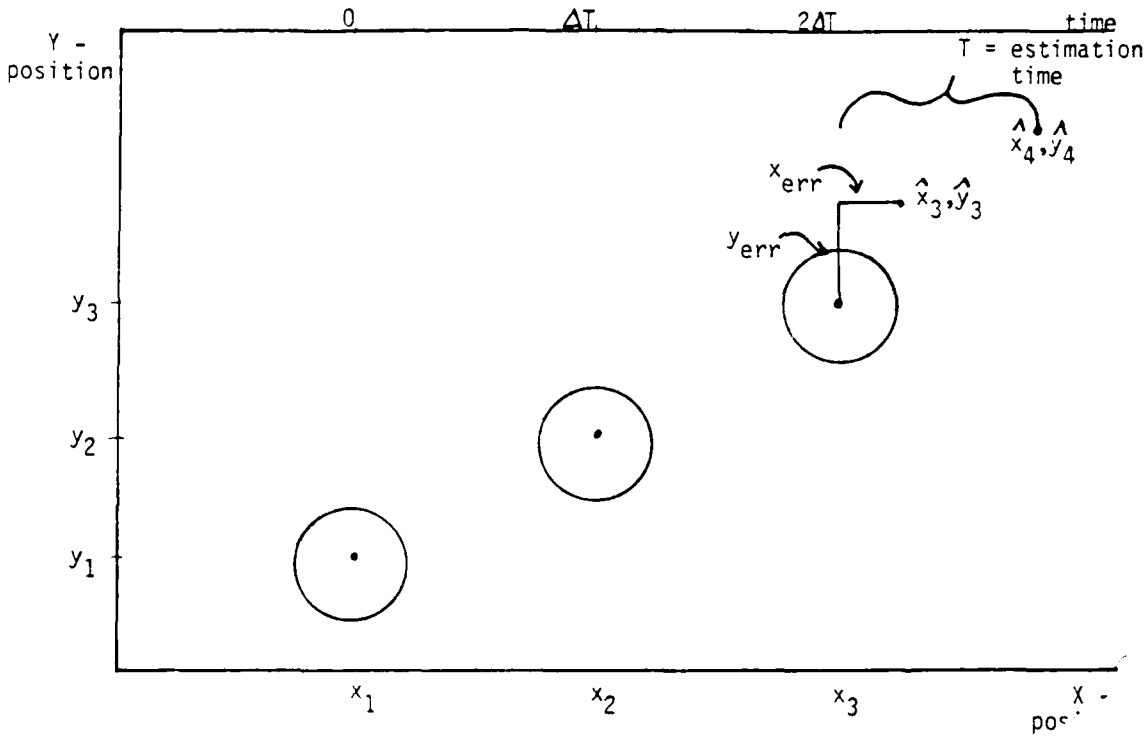
Then error terms in x and y are formed using the actual x_3 and y_3 values obtained from the third image:

$$x_{err} = x_3 - \hat{x}_3$$
$$y_{err} = y_3 - \hat{y}_3$$

Finally, the estimates $\hat{x}_4$ and $\hat{y}_4$ are found as a combination of the average estimates resulting from using the coordinate positions of the three images as separate initial conditions and appropriate velocities and times plus a contribution proportional to the error term. The necessary equations are

$$\hat{x}_4 = \frac{1}{3}(x_1 + (\dot{x}_1+\dot{x}_2)\Delta T + \dot{x}_{avg}T + x_2 + \dot{x}_2\Delta T + \dot{x}_{avg}T + x_3 + \dot{x}_{avg}T)$$
$$+ x_{err}T/\Delta T$$
$$\hat{y}_4 = \frac{1}{3}(y_1 + (\dot{y}_1+\dot{y}_2)\Delta T + \dot{y}_{avg}T + y_2 + \dot{y}_2\Delta T + \dot{y}_{avg}T + y_3 + \dot{y}_{avg}T)$$
$$+ y_{err}T/\Delta T.$$

The predicted position $\hat{x}_4$ at T seconds after acquisition of image three is the sum of four terms. The first term is the $\hat{x}_4$ coordinate estimate using x_1 as the initial position and using velocities $\dot{x}_1$ and $\dot{x}_2$ and their respective ΔT's and velocity $\dot{x}_1 + \dot{x}_2/2 = \dot{x}_{avg}$ and time T. The second and third terms are similar to the first but with initial positions x_2 and x_3 and using velocities $\dot{x}_2$ and $\dot{x}_{avg}$. These three terms are averaged and the fourth term represents an error that is linear with time. The y equation follows in a similar manner.



$\dot{x}_1 = (x_2 - x_1)/\Delta T$ $\qquad$ $\dot{y}_1 = (y_2 - y_1)/\Delta T$

$\dot{x}_2 = (x_3 - x_2)/\Delta T$ $\qquad$ $\dot{y}_2 = (y_{\sim}$

$\dot{x}_{avg} = (\dot{x}_1 + \dot{x}_2)/2$ $\qquad$ $\dot{y}_{avg} = (\dot{y}_1 +$

$x_{err} = x_3 - \hat{x}_3$ $\qquad$ $y_{err} = y_3 - \hat{y}_3$

Figure 3.14 Estimation Technique

of this data, velocities $\dot{x}_1$ and $\dot{y}_1$ are then calculated. Using this information, an estimate of the objects position in image three is made. The estimated position x_3 is found from the average of two predictions,

several object images whose temporal relationships are know. The information gained through this sequence of images allows estimation of object position at a future time. The limits on the specific estimation time period are governed by two considerations. First, the time required for image acquisition, image processing, estimate calculation, coordinate transformations, and response time of the actual manipulator movement determine the minimum estimation time. Secondly, the accuracy of the estimated position, which decreases as the estimation period increases, and the size of the camera image field, which limits the maximum position error allowed, determine the maximum estimation time that will insure that the object is in the field of view.

The demonstration example which is used to illustrate the ideas of this thesis consists of estimating the position of a moving object based on information from a sequence of three images. Figure 3.14 shows the space and time relationships in the estimation technique used. This is also performed as a software function. That is, as the program proceeds, the encoder counters are compared to the desired values. This set of differences is used to determine the velocities of the corresponding links. The method sets a joint's velocity at a fraction of the maximum velocity equal to the ratio of its encoder count difference to the maximum count difference over all five joints. This approach is applied repetitively as the move progresses, thereby providing closed-loop control.

The equations used attempt to take into account all information at each step for the purpose of minimizing large errors in any one measurement. The technique initially takes three images ΔT apart in time and finds their centers (x_1,y_1),(x_2,y_2), and (x_3,y_3). On the basis

two control loops. The first is a tachometer-controlled loop which produces a so-called "desired" motor current. This signal is compared then to the actual motor current. The error signal in this loop will determine the pulse width of a 20 KHz pulse train used to drive the individual motors of the robot. The purpose of using two control loops is to account for the nonlinearities of the robot's motors.

The P-5 process robot is electrically driven and is capable of performing a wide variety of industrial tasks. It is a five joint, five link robot which was initially designed for teaching playback type of operation.

The Estimation Used in Demonstration Task

Recall that in this thesis the main objective is to move the robotic manipulator in response to some visual information obtained from an image acquisition system. This visual information may be related to an object which is either fixed or moving. The ability to extract information such as object position, shape, size, speed, direction of movement, or even acceleration, makes possible the handling of a number of industrially related tasks. Typical tasks include object removal from or placement on a moving conveyor belt; painting, welding or other operations on a stationary or moving object; separation of objects based on the results of a visual inspection performed by the image acquisition and processing system. Since a number of industrial tasks require some interaction with moving objects, the ability to track or estimate positions of these objects is an important element in the operating features of an intelligent robot. These considerations have led to the selection of moving object interception as the demonstration example used to illustrate the ideas investigated in this thesis. This interception problem requires the use of

robot angles, it converts these to desired counter values and drives the appropriate robot motors in order to reduce the difference between the desired and actual counter values. As the difference becomes close to zero proper robot link orientation will result so that the robot end effector is in the desired position.

The assembler language program running on the CAMAC performs two basic functions. The first is a command function that interprets the information from the serial line as data and commands to be accomplished. For example, maximum velocity can be changed, the tolerance that specifies the completion of a move can be changed, zeroing the values of specific encoder counters can be accomplished and so on. The second function is that of control. In this loop the angles specified in the new "move" command are converted to encoder counts.

The velocity of the individual motors on the robot can be updated within each loop. This allows for motion of the manipulator that seems to imitate human motion. This is also performed as a software function. That is, as the program proceeds, the encoder counters are compared to the desired values. This set of differences is used to determine the velocities of the corresponding links. The method sets a joint's velocity at a fraction of the maximum velocity equal to the ratio of its encoder count difference to the maximum count difference over all five joints. This approach is applied repetitively as the move progresses thereby providing closed-loop control. This assures that each link arrives at its designated position at approximately the same time.

The output of the CAMAC is an analog signal that has been generated by the CAMAC's Digital-to-Analog converters. This voltage is applied to the control box of the P-5 robot. The voltage is amplified and fed into

The system is driven by means of a high-level Fortran program run on a PDP 11/60 digital computer. The program directs the flow of control from image acquisition, through image processing, to data delivery to a CAMAC computer.

Following the CAMAC computer an interface has been built to work with the control box of a P-5 process robot. The interface and control consist of several pieces of hardware that provide the signals to operate the motors of each of the links on the robot.

Image acquisition is accomplished through an imaging system developed by Imaging Technology, Inc. The system, which was disucssed in greater detail in Chapter II, allows for the acquisition or construction of 2-D images. The system provides the ability to drive the imaging hardware by use of high-level languages, such as Fortran and Basic. There are a number of useful subroutines that allow for a wide variety of processing applications.

The PDP 11/60 is a digital computer that acts as the main controller in the overall system. The PDP 11/60 processes the acquired data and presents the necessary information over a serial line to the CAMAC computer.

The CAMAC computer is basically a digital computer that acts as an interface between the digital world of the PDP 11/60 and the analog world of the P-5 robot's control box.

The CAMAC operation monitors the values held by a group of up-down counters which record the number of encoder pulses from the corresponding joint angle encoder of the robot. The values of these counters are a measure of the appropriate joint angles of the robot and are zero at the robots "home" position. When the CAMAC computer receives a set of desired

the position of the laser. Step four consisted of moving the robot to several known positions and determining the location of the object in camera coordinates. After each step the vector that expresses the position of the end effector in the coordinates of the fifth link is derived using the equation:

$$p_5 = (T_5^o)^{-1} p_o$$

As a result a relationship between the camera coordinates and the coordinates of the fifth link can be found. Finally, using a least squares approach, the A_{cam}^5 matrix transformation is determined [see Appendix A].

Components of the System

A description of the specific hardware and software system under consideration in this thesis is now given. The system block diagram is presented in Figure 3.13.

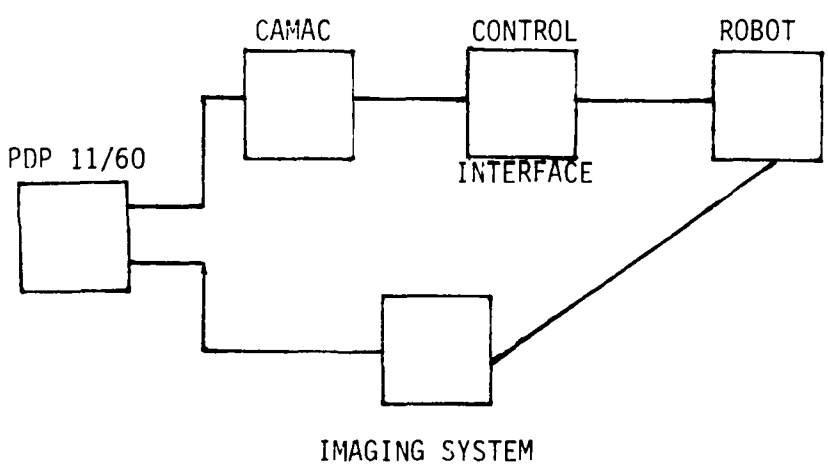


Figure 3.13 Block Diagram of System

The system observes the location of an object in the image field or camera coordinates p_{cam}. A transformation is necessary to express the object's position in link 5 coordinates (A^5_{cam}). Obviously, then the position expressed in base coordinates can be given as:

$$p_o = T^o_5 \, A^5_{cam} \, p_{cam}$$

The transformation matrix A^5_{cam} simply represents a set of equations that take the form of:

$$x_5 = ax_{cam} + by_{cam} + x_o$$
$$y_5 = cx_{cam} + dy_{cam} + y_o$$

These equations can be expressed in a homogeneous matrix form as:

$$\begin{vmatrix} x_5 \\ y_5 \\ z_5 \\ 1 \end{vmatrix} = \begin{vmatrix} a & b & 0 & x_o \\ c & d & 0 & y_o \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \\ 1 \end{vmatrix}$$

The system used in this research did not have a permanently attached camera. As a result, a calibration of the camera referred to link 5 coordinates had to be accomplished whenever the camera was remounted. The basic purpose of the calibration was to derive the six unknown constants of the A^5_{cam} matrix.

The logical design of the calibration routine was relatively straight forward. Step one consisted of positioning the robot in a 'home' orientation. Next, a laser which is mounted along with the camera was used to produce a light image in the camera field. Because the laser's position is fixed in reference to the camera's position, the location of the light image in the camera field is also fixed. Step three was to move the robot to a known position and place an object under

ginning of a move and the completion of the move. These two values were examined by using a system routine that was able to specify the time increments to within 1/60 second. The necessary parts of the program were repeated and timed as was the actual movement of the robot. The experiment was conducted using a wide range of robot motion. After running through the procedures 1000 times the values obtained were:

AVERAGE TIME FOR CALCULATION = .291 seconds

AVERAGE TIME FOR ROBOT MOVE = .854 seconds

The data in this section is provided to indicate to the user the average amount of time needed to process the data in software, derive an estimate of the object's position, and to actually move the robotic manipulator. The total of these two values will yield the minimum estimation time that the system can work with. Using a smaller estimation time would result in the robot's inability to beat the object to the position even if all error was removed from the system.

In the following section an evaluation of the overall system accuracy expressed in pixels is given. The experiment for obtaining data to examine the general system consisted of running the system with a minor software modification. This modification was the addition of a timing loop that timed out the estimation time using the system clock. At this point an image was taken and the object's position was examined

The experiment consisted of running this modified program through a loop 100 times. In the loop the robot started from the home position each time. At this location the acquisition of three images of a moving object was accomplished. Using these images, an estimate of the object's position at a specific time later was determined. The manipulator was directed to move to this estimated position where it waited for the system timer to indicate that the estimation period had elapsed. An

image was taken and evaluated to determine the object's new position compared to the location of the laser in the image field during calibration. It is assumed that the laser position can be considered the system pointer because the initial calibration of the system was performed using the laser image location as a reference point. An error, expressed in pixels, in both the x and y directions was calculated and stored. Finally, the robot was directed back to its home position and the process began again.

After the sequence of loops the error information was used to determine an average error and a standard deviation. The experiment examined both linear and circular motion using estimation times of three, five, seven and nine seconds.

The different types of motion were achieved by using a small DC motor. Linear motion was established by connecting the object to the driveshaft of the motor by means of a piece of thread. The driveshaft was driven at a constant rpm and as a result the thread was reeled in at a constant rate. The system was oriented so the object would move in a purely x direction or a purely y direction. The velocity of the object was measured at 36 pixels/sec. Circular motion was provided by placing a piece of cardboard with the object attached onto the driveshaft of the motor. The object was located a distance of 4 inches from the shaft and had an angular velocity of 1.5 rpm. In Table 4.2 the data for this experiment is listed.

The information in Table 4.2 consists of two sets of data associated with linear and circular object motion. The data consists of four values, the average error in both the x and y direction along with the appropriate standard deviations. The error value were derived by ex-

Table 4.2 Overall System Error Test Data

a) Linear Motion

	Estimation Time (sec.)	$\bar{x}_e$	$\bar{y}_e$	σ_x	σ_y
pure y direction	3	1.84	13.89	1.5474	4.5098
"	5	1.83	23.41	1.2966	4.5455
"	7	2.49	31.41	2.1748	6.6485
"	9	3.74	55.92	2.4067	10.6035
pure x direction	3	15.54	4,27	2.3934	3.2276
"	5	24.73	4.99	2.8770	3.4569
"	7	28.96	6.17	4.8988	4.1954
"	9	40.88	10.60	6.1307	7.6766

b) Circular Motion

	3	53.8	62.34	28.4109	31.1414
	5	79.82	91.13	38.7372	45.4492
	7	103.27	117.55	47.6411	55.3914
	9	153.78	160.68	70.8511	80.1495

amining the position of the object at the estimation point in reference to the laser's position in the image field. It is interesting to note the shift in the magnitude of error depending on whether the linear motion is in the x or y direction. The inactive direction has a smaller value of error compared to the active direction. This small error can come about from inaccuracies in determining the object's centers and may grow with the time of estimation if consecutive errors imply a velocity which is in fact not present. However, as the data shows, for reasonable prediction times these errors can be considered negligible. As a result, important information can only be discerned with the active direction data.

Model for Experimental Results

A framework for interpreting the experimental data presented in the previous section is developed in the following paragraphs. Figure 4.1 graphically shows how the error variance of a linear predictive motion model changes with prediction time. The simulation is performed using a discrete linear Kalman filter expressed by the equations given in Table 4.3. This model was used because the Kalman filter is considered the best (minimum error variance) linear filter for either state observation or estimation [Sage and Melsa, 1971]. As a result, by comparing how the error variance would change with time in the experiment compared to the values given in the Kalman filter, an indication of relative accuracy for the overall system is obtained.

Figure 4.1 shows how the error variance increases with increasing estimation time. This graph was determined by examining an equation that yields the change of error variance through a linear system [Sage and Melsa, 1971]. The equation is:

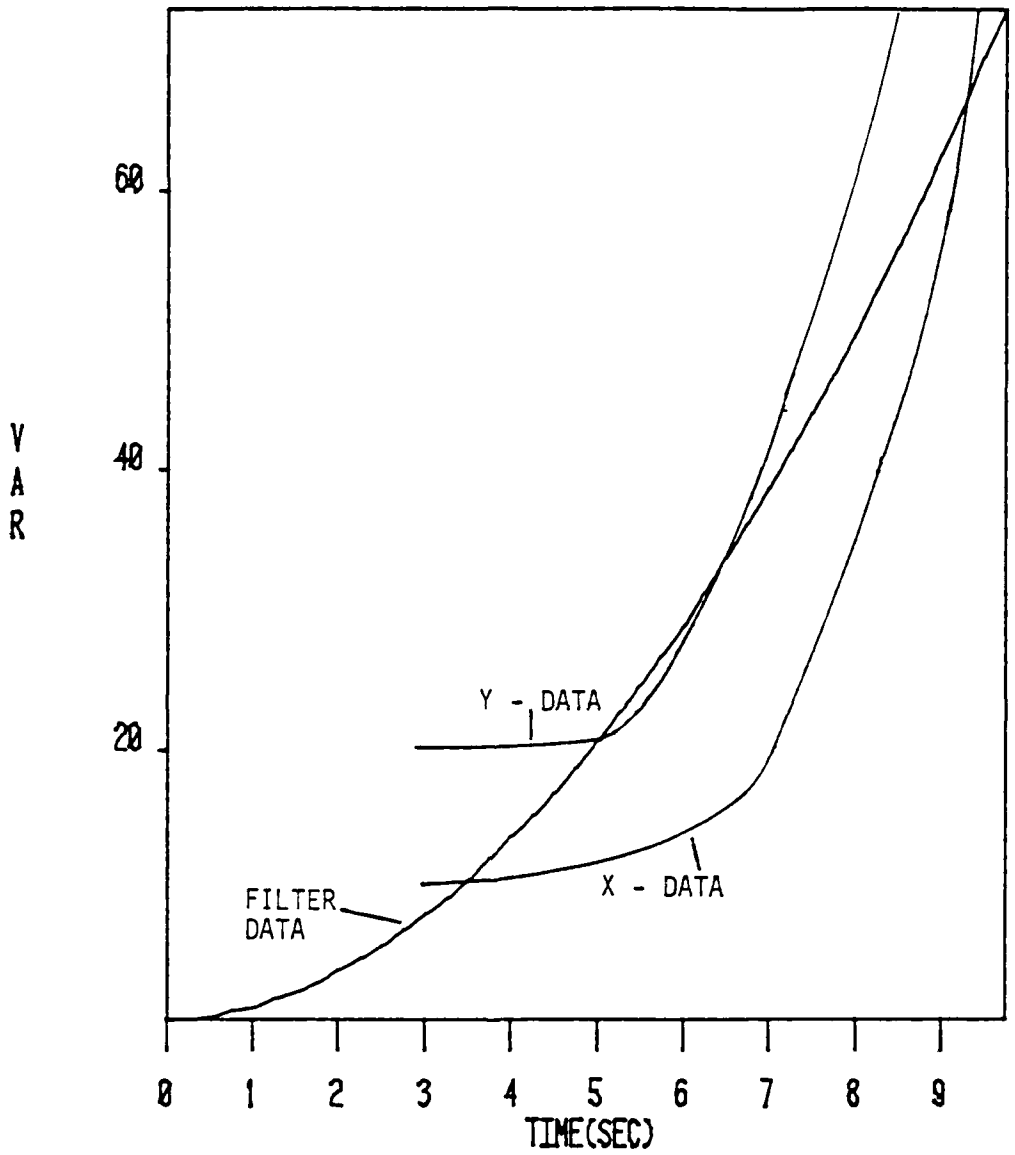


Figure 4.1 Variance of Error in Estimation as a Function of Prediction Time

Table 4.3 General Discrete Kalman Filter

Message model	$x(j+1) = \Phi(j+1,j)x(j) + \Gamma(j)w(j) + B(j)u(j)$
Observation model	$z(j) = H(j)x(j) + v(j) + y(j)$
Prior statistics	$E\{w(j)\} = \mu_w(j) \qquad E\{v(j)\} = \mu_v(j) \qquad E\{x(0)\} = \mu_x(0)$ $\mathrm{cov}\{w(j),w(k)\} = V_w(j)\delta_K(j-k)$ $\mathrm{cov}\{v(j),v(k)\} = V_v(j)\delta_K(j-k)$ $\mathrm{cov}\{w(f),v(k)\} = V_{wv}(j)\delta_K(j-k)$ $\mathrm{var}\{x(0)\} = V_x(0)$
One-state-prediction algorithm	$\hat{x}(j+1\|j) = \Phi(j+1,j)x(j) + \Gamma(j)\mu_w(j) + B(j)u(j)$ $+ K_p(j)[z(j) - \mu_v(j) - y(j) - H(j)\hat{x}(j)]$
Filter algorithm	$\hat{x}(j+1) = \hat{x}(j+1\|j) + K(j+1)[z(j+1) - \mu_v(j+1) - y(j+1)$ $- H(j+1)\hat{x}(j+1\|j)]$
One-state-prediction gain algorithm	$K_p(j) = \Gamma(j)V_{wv}(j)V_v^{-1}(j)$
Filter gain algorithm	$K(j+1) = V_x^{\sim}(j+1\|j)H^T(j+1)[H(j+1)V_x^{\sim}(j+1\|j)H^T(j+1)$ $+ V_v(j+1)]^{-1}$
A priori variance algorithm	$V_x^{\sim}(j+1\|j) = [\Phi(j+1,j) - K_p(j)H(j)]V_x^{\sim}(j)$ $x[\Phi(j+1,j) - K_p(j)H(j)]^T + \Gamma(j)V_w(j)\Gamma^T(j)$ $- K_p(j)V_v(j)K_p^T(j)$
A posteriori variance algorithm	$V_x^{\sim}(j+1) = [I - K(j+1)H(j+1)]V_x^{\sim}(j+1\|j)$
Initial conditions	$\hat{x}(0) = \hat{x}(0\|0) = \mu_x(0) = E\{x(0)\}$ $V_x^{\sim}(0) = V_x^{\sim}(0\|0) = \mathrm{var}\{x(0)\} = V_x(0)$

$$V_x(k+1) = \Phi(k+1)\ V_x(k)\Phi^T(k+1) + \Gamma(k)\ V_w(k)\Gamma^T(k)$$

where $V_x(k)$ is state error variance, $\Phi(k)$ is state transition matrix at time k, $\Gamma(k)$ is a weighting matrix and $V_w(k)$ is the variance of the noise. The second term of this sum is concerned with how the noise on the state measurement changes with time. Since in the estimation technique the prediction of an object's position involves no measurements this term is zero. Therefore the equation reduces to:

$$V_x(k+1) = \Phi(k+1)\ V_x(k)\Phi^T(k+1)$$

This equation reveals that the change of variance with time in the estimation application is dependent on the state transition matrix and the previous variance.

In evaluating the system data in reference to the Kalman filter, Figure 4.1 shows that system error variance in the prediction range from five seconds to eight seconds is very close to that of the Kalman filter. For both cases the values of error variance for estimation time above eight seconds are too large for practical prediction. It is important to recognize that for the range of estimate times from three to five seconds the error variance of the system is relatively constant and then begins to increase in the same way as the filter error variance.

The results here seem to suggest that with regard to linear motion estimation the variance changes of the demonstration system are very similar to those of the Kalman filter.

The circular motion experiment was conducted to examine how the linear models of the system and Kalman filter could handle such a trajectory. There would definitely be a difference between the Kalman filters and the system in merely tracking the object without estimation. Recall that there is a constraint of minimum prediction time due to

calculation and mechanical movement of the robotic system. Therefore this particular "intelligent" robotic system is limited to using a prediction algorithm. That is, a need for predicting ahead at least the necessary mechanical time is required. On the other hand, the software simulation does not have this limitation. Figure 4.2 shows that the Kalman filter was able to lock on to the actual trajectory after only a few iterations. This trajectory was provided by actual data of circular object motion obtained by visually sampling the object at .25 second intervals.

Both the system and filter faltered when attempt at prediction was taken. The data for the system reveals an average error of more then the diameter of an object even with a 3 second prediction time. Figure 4.3 reveals the inherent problem of using a linear model to predict circular motion. The model uses the velocity of the last data point and predicts ahead based on this trajectory. It is obvious that immediately the observations diverge from the estimations. Notice that at a specific point the errors start to decrease because the circular motion is in the same direction as the linear prediction.

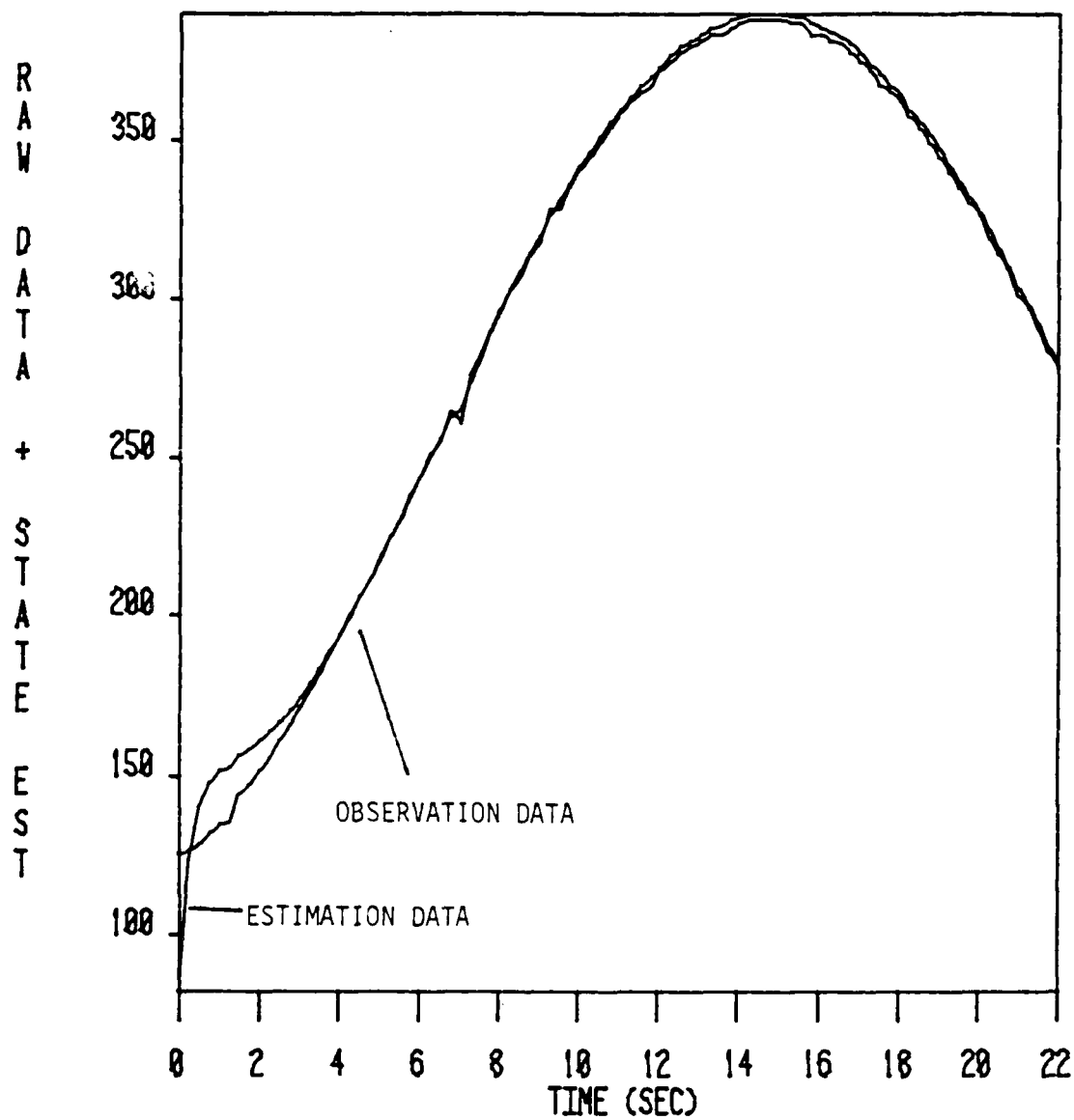


Figure 4.2 Kalman Filter Estimates and Experimental Observations for Circular Motion

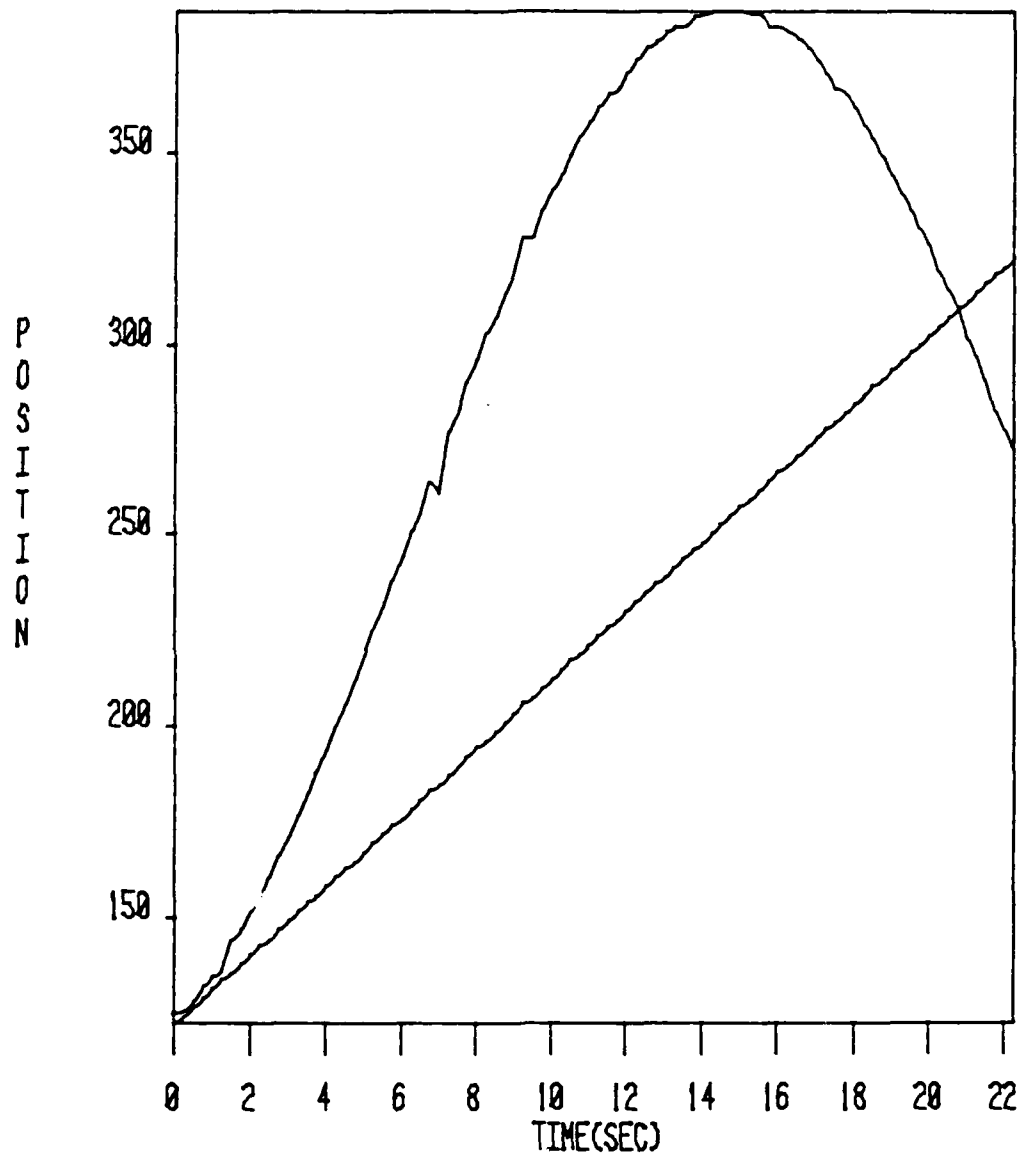


Figure 4.3 Circular Motion Prediction Using

Linear Predictive Algorithm

CHAPTER V

CONCLUSIONS AND FUTURE RESEARCH

In this thesis a system that can direct the motion of an industrial robot in response to visual information acquired through an image acquisition system was investigated. In this chapter conclusions of both a general and specific nature are given and suggestions are made for future research related to this system.

The principal conclusion from this research is that an industrial robot can be directed by information acquired from a video source. The system was used to implement the software discussed in this thesis. As evidenced by the data presented in Chapter IV, the robot was able to respond very well to the movement of an object in the camera's image field. Along these lines there are a number of constraints on the system that effect how it can be used. As mentioned before, the time it takes to calculate the position of an object in the image field and transform this data into robot coordinates takes approximately 0.2 seconds. In addition the average time the robot takes to complete a movement is approximately 1.0 second. Therefore, even if the system was designed to have an accurate object following routine, it would be necessary to make an estimate of the object's position at least 1.2 seconds in advance. If the intent was to have the system follow a moving object with no attempt made to compensate for this time lag, system inaccuracy would increase with object velocity. Another interesting consideration is the basic limitation on object velocity. The velocity of the object must be limited to that of the robot in any tracking operation. Also since the data for

linear prediction in Chapter 4 represented objects moving rather slowly (36 pixel/sec) the error values indicate that prediction times of more than 7 seconds cannot be justified.

The robotic system as it is set up now is unable to respond accurately to movement of the object while the robot arm is in motion. This is a very complicated problem for both the image processing technique and the robotic controller. The acquisition of the image while the robotic arm is in motion may cause blurring which can introduce an inaccuracy into the system that the stationary system does not have. In addition the inability to know exactly where the robot was at the time when the image was taken yields the fundamental weakness of this dynamic system for very accurate tracking and prediction attempts.

The robot system controller as it now operates allows for a large amount of overshoot and settling time at the end of each movement. This could be a problem if image acquisition were to take place during this settling time. The effect of vibration could enter a larger amount of inaccuracy into any images taken during the period. In addition the system assumes that the camera is parallel to the working plane and remains at a constant height. Any deviation from these constraints will also introduce inaccuracies into the system.

Suggestion for Future Research

The system as it now stands, and as it was initially drawn out, is an introduction to this lively field of robotic vision. At this point there are a number of areas where improvements are both necessary and possible.

The first area is that of the image processing. The fact that noise is introduced by the robot's motors eliminated the use of the turtle routine as mentioned in Chapter two. With the incorporation of a turtle routine the ability to find the true centroid of an object even a non-symmetrical object would be possible. There are a number of methods by which the noise problem could be approached. Through simple software techniques the system could incorporate a low pass filter or average over a specified window within the image field. The effect can be seen in Figure 5.1. The window would pass over the 2-D image contained in the frame buffer and examine a number of pixels under the window [3x3, 5x5, 7x7, etc]. The algorithm would average all pixels and replace the center pixel of the window with this average. Of course, if a binary image was being used some threshold average would be decided on and if the average was below the threshold the center pixel would be replaced with a black pixel, otherwise a pure white pixel would take its place. The obvious effects would be random noise pixels should be eliminated, the time needed to process the image would be increased and the image of the object would suffer a slight distortion. One must not overlook the possibility of using a hardware filter to eliminate this high frequency noise. Maybe by investigating the source of this sensitivity to noise an appropriate shielding technique can be used to avoid the two previous techniques. In either case the tradeoffs would need to be examined and weighed.

The striking effect of eliminating the noise pixels and then the incorporation of the turtle boundary/follower routine would be the entrance into the area of pattern recognition and the applications this arena offer.

NOISE PIXEL

0	0	0	0	0	0	0
0	250	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	250	250	250
0	0	0	0	250	250	250
0	0	0	0	250	250	250

ORIGINAL IMAGE

OBJECT PIXELS

0	0	0	0	0	0	0	
0	27.5	27.5	0	0	0	0	
0	27.5	27.5	27.5	55.0	82.5		
0	0	55.0	110.0	165.0	165.0		
0	0	0	82.5	165.0	250	250	
0	0	0	82.5	165.0	250	250	

FILTERED IMAGE

0	0	0	0	0	0	0	
0	0	0	0	0	0	0	
0	0	0	0	0	* 250	* 250	
0	0	0	* 250	250	250	250	
0	0	0	* 250	250	250	250	
0	0	0	* 250	250	250	250	

FILTERED & THRESHOLDED IMAGE

threshold = 80

* blurring

Figure 5.1 Averaging Technique for Noise Problem

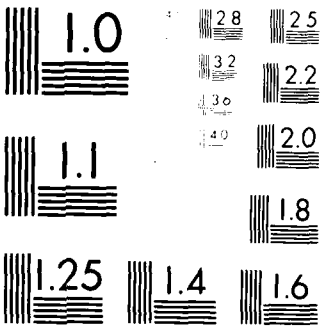
AD-A151 933 POSITIONING OF A ROBOTIC MANIPULATOR THROUGH THE USE OF VISUAL FEEDBACK(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH M T O'DONNELL MAY 85 2/2

UNCLASSIFIED AFIT/CI/NR-85-20T F/G 6/4 NL

END

FILMED

DTIC

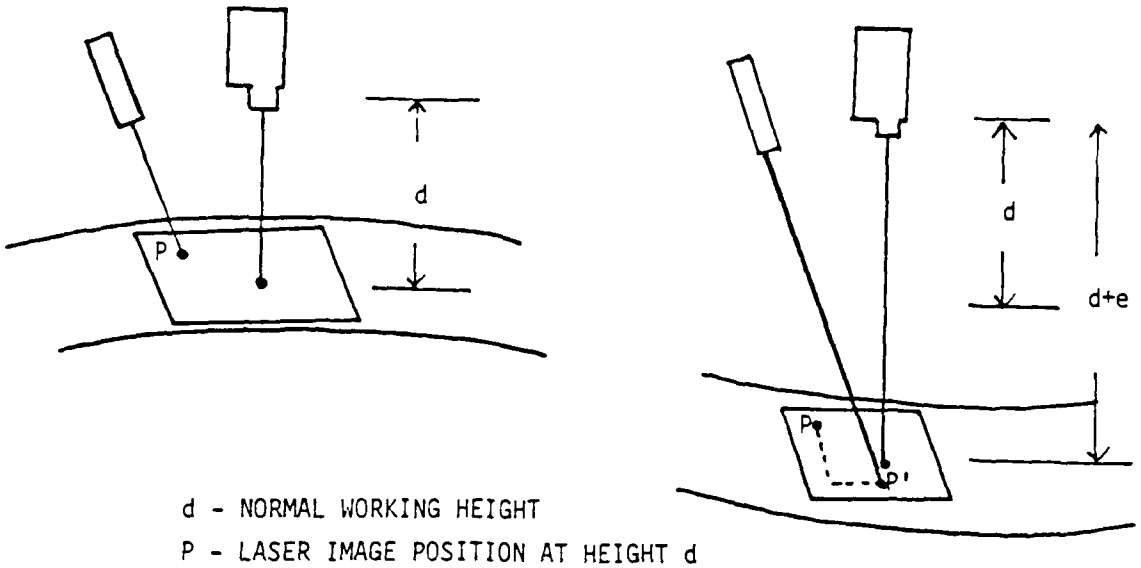


MICROCOPY RESOLUTION TEST CHART

By delineating between shapes [Dubois, 1984] the ability to approach the bin-picking problem is facilitated. At this point the system moves from the periphery of the robotic vision research into areas of current research.

Also in the area of image processing the ability to move beyond the area of two dimensional information into the 3-D arena is within the grasp of the system as it now stands. The structure of the laser source mounted with the camera had a two-fold purpose. First, the laser provided a reference point in the calibration of camera to robot coordinates. Secondly it offers the possibility of gauging depth between the camera height and working level. At the present time the system incorporates a flat top table as the working surface. However, by incorporating knowledge about the position of the laser in the image field, the ability of maintaining a fixed height above a variable depth surface is not far out of reach. In Figure 5.2 the effect of driving the robot so that the laser image remains fixed in the image field results in this fixed depth application. The result here may be seen as the ability to apply equal layers of paint on a non-flat surface or to inspect equal size areas of such a surface. The true results of such an ability may only be reaped when the algorithm is incorporated.

In the area of kinematics the present movement is obviously awkard. That is, the robot is directed to a position and the system then waits for the movement to be complete. At which point the necessary information is available to give the next direction. The system does have the capability of receiving a second movement directive during the actual initial move and responding to it. However, by incorporating the mounted camera's visual information a reference is needed to know when and where



d - NORMAL WORKING HEIGHT

P - LASER IMAGE POSITION AT HEIGHT d

P'- LASER'S POSITION AT NEW HEIGHT

Figure 5.2 Image Field at Two Different Heights

the visual information was obtained. That is the ${}^{0}T_5$ matrix must constantly be available in order to have the proper information for an updated movement. The most obvious result of this improvement would be a more efficient use of the time during which the robot is moving. There is a possibility that with the acquisition of images during the movement of the manipulator a blurring effect may be introduced and as a result additional picture processing may be called for. The field of Image Enhancement and techniques for deblurring of images are some of the current topics in image processing.

The ability to improve upon the technique for estimation in the system's tracker application is definitely within reach. The inability of this system to give effective estimates for circular motion could obviously be improved by basing the estimate on a curvilinear model or increasing the order of the estimation equation. Furthermore by incorporating a type of adaptive model, there is a possibility of enhancing the system's performance with both linear and circular motion.

This adaptive solution may examine the error values for both types of model in an iterative fashion. Depending on which model yielded the smallest error an estimation would be made. The ability to improve upon either estimation technique would result in a tracking system that could be used in a range of expectant trajectories.

REFERENCES

Aggarwal, J.K. and Martin, W.N., "Dynamic Scene Analysis - A Survey," Computer Graphics and Image Processing, Vol. 7, 356-374, 1978.

Agin, Gerald J., "Computer Vision Systems for Industrial Inspection and Assembly," Computer, Vol. 13, No. 5, 11-20, 1980.

Ballard, Dana H., and Brown, Christopher M., Computer Vision, Prentice Hall, Englewood Cliffs, NJ, 1982.

Bolc, Leonard and Zenon, Kulpa, Digital Imaging Processing Systems, Springer-Verlag, Berlin, 1981.

Castleman, Kenneth R., Digital Image Processing, Prentice Hall, Englewood Cliffs, NJ, 1979.

Davis, Larry S., "A Survey of Edge Detection Techniques," Computer Graphics and Image Processing, Vol. 4, 248-269, 1975.

Denavit, J. and Hartenberg, R.S., "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices", Journal of Applied Mechanics, 215-221, 1955.

Dubois, Susan Rae, The Classification of Objects by the Use of Autoregressive Models of Their Digital Images, University of New Hampshire, Master's Thesis, 1984.

Duda, R.O. and Hart, P.E., Pattern Classification and Scene Analysis, Wiley, New York, 1973.

Frei, Werner, "Image Enhancement by Histogram Hyperbolization," Computer Graphics and Image Processing, Vol. 6, 286-294, 1977.

Hall, E.L., Tio, J.B.K., McPherson, C.A. and Sadjadi, F.A., "Measuring Curves Surfaces for Robotic Vision", Computer, Vol. 15, No. 12, 42-53, 1982.

Hanson, Allen R., and Riseman, Edward M., Computer Vision Systems, Academic Press, New York, 1978.

Hoeschele, David F. Jr., Analog-to-Digital/Digital-to-Analog Conversions Techniques, Wiley, New York, 1968.

Horn, B.K.P. and Ikevchi, K., "The Mechanical Manipulation of Randomly Oriented Parts", Scientific American, Vol. 251, No. 2, 100-111, 1984.

Hummel, Robert A., "Histogram Modification", Computer Graphics and Image Processing, Vol. 4, 209-225, 1975.

Image Technology Inc. Imaging-Basic Driver Programmer's Manual, 1984.

Kunt, M., "Acquisition and Visualization", Fundamentals in Computer Vision, edited by O.D. Faugeras, Cambridge University Press, Cambridge, 1983.

Lanczos, Cornelius, Applied Analysis, Prentice, Englewood Cliffs, NJ, 1956.

Lee, C.S.G., "Robot Arm Kinematics, Dynamics, and Control", Computer, Vol. 15, No. 12, 62-79, 1982.

Leese, J.A., Novak, C.S., and Taylor, J.R., "The Determination of Cloud Pattern Motion From Geosynchronous Satellite Image Data", Pattern Recognition, Vol. 2, 279-292, 1970.

Miller, R.K., Machine Vision - Fundamentals, Vol. 1, Technical Insights, Fort Lee, N.J., 1984.

Muroga, Saburo, VLSI System Design, Wiley, New York, 1982.

Nahin, P.J., Image Processing by Computer, University of New Hampshire, Durham, 1983.

Ohlander, Ron, Price, Keith, and Reddy, Ray D., "Picture Segmentation Using a Recursive Region Splitting Method," Computer Graphics and Image Processing, Vol. 8, 313-333, 1978.

Paul, R.P., Robot Manipulators - Mathematics, Programming, and Control, The MIT Press, Cambridge, 1981.

Robinson, Guner S., "Edge Detection by Compass Gradient Masks", Computer Graphics and Image Processing, Vol. 6, 492-501, 1971.

Sage, Andrew P., and Melsa, James L., Estimation Theory With Applications to Communications and Control, McGraw-Hill, Inc., 1971.

APPENDICES

APPENDIX A: LEAST SQUARES CALIBRATION METHOD AND SOFTWARE LISTING

The least square calibration of the camera to robotic link 5 (end effector) coordinate transformation is obtained as follows:

$$X = (A^T A)^{-1} A^T B$$

where

X = the vector of least squares coordinate transformation constants. $\begin{bmatrix} a & c \\ b & d \\ x_o & y_o \end{bmatrix}$

A = the matrix of observed camera coordinates with 12 rows of values $(x_{cam}, y_{cam}, 1)$.

B = the matrix of the x and y position values of the object expressed in fifth link coordinates with 12 rows of values (x_5, y_5).

Thus

$$\begin{bmatrix} x_5 \\ y_5 \end{bmatrix} = [x_{cam} \; y_{cam} \; 1] \begin{bmatrix} a & c \\ b & d \\ x_o & y_o \end{bmatrix}$$

```
C
C       FILE LSTSQS.FTN          REVISED 12 JUNE 1984
C                                AUTHOR MARK FURBER
C
C       THIS FILE CONTAINS FORTRAN SUBROUTINES:
C
C                        LSTSQS
C                        SNVERT
C
C
C
C
C       SOME SUBROUTINES USED BY LSTSQS AND INVERT ARE CONTAINED IN MATH.F
C
C*********************************************************************
C*********************************************************************
C       NON-RECURSIVE LEAST-SQUARES SUBROUTINE
C       SOLVES A(MXN) * X(NX1) = B(MX1)

0001            SUBROUTINE      LSTSQS(A,X,B,AT,ATA,ATA1,ATA1AT,ASYM,M,N,P,IER)


0002            INTEGER M,N,P

0003            REAL    A(M,N), X(N,P), B(M,P)

                REAL    AT(N,M), ATA(N,N), ATA1(N,N), ATA1AT(N,M), ASYM(N*(N+1)/2

0005            INTEGER IER
                ERROR CODE RETURNED BY INVERSION ROUTINE

C-----------------------------------------------------------------


0006            CALL    MATTRN(AT, A, N, M)
0007            CALL    MATMUL(ATA, AT, A, N, M, N)
0008            CALL    SNVERT(ATA1, ATA, ASYM, N, IER)
0009            IF(IER .NE. 0) GOTO 999

0010      100   CALL    MATMUL(ATA1AT, ATA1, AT, N, N, M)
0011            CALL    MATMUL(X, ATA1AT, B, N, M, P)


0012      999   RETURN
0013            END
```

```
C*********************************************************************
C*********************************************************************
C       SYMMETRIC MATRIX INVERSION SUBROUTINE

        SUBROUTINE      SNVERT(A1, A, ASYM, N, IER)


        INTEGER N

        REAL    A(N,N), A1(N,N)

        REAL    ASYM(N*(N+1)/2)

        INTEGER IER

        REAL    D1, D2

        INTEGER I,J

C-----------------------------------------------------------------

        IER = 0

C       CREATE IDENTITY MATRIX IN A1
        DO 100 I=1,N
        DO 110 J=1,N
        A1(I,J)=0.0
 110    CONTINUE
 100    CONTINUE
        DO 120 I=1,N
        A1(I,I)=1.0
 120    CONTINUE

        CALL    MATSYM(ASYM,A,N)
        CALL    LEQT1P(ASYM, N, N, A1, N, B, D1, D2, IER)

        RETURN
        END
```

```
C
C
C       THIS FILE CONTAINS THE ROBOT/CAMERA CALIBRATION PROGRAM
C
C
C*********************************************************************
C*********************************************************************

0001          REAL     PHOME(4),PCENTER(4)
C             STANDARD POSITIONS FOR ROBOT

0002          REAL     PBUB(4,12),PUB(4),PBUB5(4),PBUB0(4)
C             MATRIX CONTAINING CENTER OF BUBBLE COORDINATES

0003          REAL PLASER(4)
C             POSITION OF LASER IN CAMERA COORDINATES (FIXED)

0004          REAL     PICPTO(4,12)
C             CALIBRATION PATTERN

0005          INTEGER NBUB
C             NUMBER OF BUBBLES FOUND BY SBUB

0006          REAL     PPICO(4),THETA(5)
0007          REAL     T05(4,4), PICT05(4,4), PICT50(4,4), OBJT05(4,4)

0008          REAL     A(12,3),X(3,2),B(12,2)
0009          REAL     AT(3,12),ATA(3,3),ATAI(3,3),ATAIAT(3,12),ATASYM(6)
C             FOR LEAST-SQUARES

0010          REAL     AXHAT(12,2),MSERRX,MSERRY,ITST(3,3)
C             FOR CALCULATING MEAN-SQAURE ERRORS AND MATRIX INVERSION

0011          INTEGER STAT
C             STATUS WORD RETURNED BY CAMAC

0012          INTEGER IER
C             ERROR CODE RETURNED BY LSTSQ5


0013          INTEGER I,II,J,10,110
C             LOOP COUNTERS

0014          REAL     NWAIT

C-----------------------------------------------------------------------

0015          DATA     PHOME0/90.0, 0.0, 60.0, 1.0/


0016          DATA     PICPTO/ 0.0,     -12.0,   0.0,     1.0,
              1                12.0,    0.0,     0.0,     1.0,
              1                0.0,     12.0,    0.0,     1.0,
              1                -11.0,   0.0,     0.0,     1.0,
              1                7.79,    -4.5,    0.0,     1.0,
```

```
     1                7.79,    4.5,     0.0,     1.0,
     1                -7.79,   4.5,     0.0,     1.0,
     1                -7.79,   -4.5,    0.0,     1.0,
     1                3.0,     -5.20,   0.0,     1.0,
     1                3.0,     5.20,    0.0,     1.0,
     1                -3.0,    5.20,    0.0,     1.0,
     1                -3.0,    -5.20,   0.0,     1.0/

C-----------------------------------------------------------------------

      WRITE(5,4)
    4 FORMAT(/' CALIBRATION ROUTINE '/)

      WRITE(5,5)
    5 FORMAT(/' ENTER X-Y COORDINATES OF OBJECT (IN CM):'/
     1        '    X = '$)
      READ(5,*)FOBJO(1)
      WRITE(5,6)
    6 FORMAT('    Y = '$)
      READ(5,*)FOBJO(2)
      FOBJO(3) = 60.0
      FOBJO(4) = 1.0

      WRITE(5,7)
    7 FORMAT('  /' NUMBER OF SECONDS TO WAIT = '$)
      READ(5,*)NWAIT

C     MOVE TO HOME POSITION
      CALL    MOVE(PHOME0,TOS,THETA,STAT)


C     FIND LOCATION OF LASER
      WRITE(5,8)
    8 FORMAT(' FINDING LASER'/)
      CALL    SRUB(PRUBC, NRUB, 3)
      IF( NRUB .NE. 1 ) GOTO 999
      CALL    MATMOV(PLASRC, PRUBC(1,1), 4, 1)


C     GET PARTIAL DATA IF ANY
      J = 0
      WRITE(5,9)
    9 FORMAT(' ANY PARTIAL DATA IN CALDAT.DAT FILE (1=YES,-1=NO)? '$)
      READ(5,*)II
      IF(II .NE. 1) GOTO 20
      WRITE(5,29)
   29 FORMAT(' PREVIOUS DATA '/)
      CALL    ASSIGN(3,'SY:CALDAT.DAT',13)
C     READ IN DATA UNTIL EOF
   30 READ(3,99,END=35)A(J+1,1),A(J+1,2),A(J+1,3),B(J+1,1),B(J+1,2)
      WRITE(5,91)A(J+1,1),A(J+1,2),A(J+1,3),B(J+1,1),B(J+1,2)
      J = J + 1
      GOTO 30
   35 CLOSE( UNIT=3 )
```

```
0049      20      J0  = J + 1

0050              CALL ASSIGN(3, 'SY:CALDAT.DAT',13)

          C       PLACE OBJECT
0051              CALL     MOVE(POBJ0,OBJT05,THETA,STAT)

0052              WRITE(5,50) J0
0053      50      FORMAT(' ',  J0 =  ,I4/' PLACE WHITE DOT THEN ENTER '1'[CR]: $)
0054              READ(5,*)II
0055              IF(II .NE. 1) GOTO 999

0056              IF(J0 .GE. 12) GOTO 400
0057              DO 400 J=J0,12

                      MOVE TO POSITION AT WHICH TO TAKE PICTURE
0058                  WRITE(5,37)J
0059      37          FORMAT(    ,  J =  ,I4/)
0060                  CALL          VCADD4( PPICO, POBJ0, PICPTO(1,J))
0061                  CALL          MOVE(PPICO,PICT05,THETA,STAT)

0062                  CALL          WAIT(NWAIT)
0063      49          CALL          SRUB(PRUBC, NRUB, 3)
0064                  IF(NRUB .EQ. 2) GOTO 55
0065                  WRITE(5,57)
0066      57          FORMAT(' TRYING AGAIN '/)
0067                  GOTO 49

                      DISTINGUISH BETWEEN LASER AND OBJECT
0068      55          IF(VECEQU(PLASRC,PRUBC(1,1),4,4.0))
                1        CALL MATMOV(POBJC,PRUBC(1,2),4,1)
0069                  IF(VECEQU(PLASRC,PRUBC(1,2),4,4.0))
                1        CALL MATMOV(POBJC,PRUBC(1,1),4,1)

0070                  WRITE(5,56)POBJC(1),POBJC(2),POBJC(3),POBJC(4)
0071      56          FORMAT(/' POSITION OF OBJECT IN CAMERA COORDINATES = 4F5.0)

          C           TRANSFORM TO COORDINATE SYSTEM 3
0072                  CALL          INVT05(PICT05, PICT50)
0073                  CALL          MATMUL(POBJ5, PICT50, POBJ0, 4, 4, 1)

                      ADD THIS POSITION TO DATA ARRAYS
0074                  A(J,1)=POBJC(1)
0075                  A(J,2)=POBJC(2)
0076                  A(J,3)=POBJC(4)
0077                  B(J,1)=POBJ5(1)
0078                  B(J,2)=POBJ5(2)
```

```
0017          IF(ABS(DELTA(1).LT.2).AND.(ABS(DELTA(2)).LT.2))GOTO 500

0018          C=0.0

        C     FIND THE VELOCITY OF OBJECT IN IMAGE TWO

0019          VELX2=DELX2/TIME
0020          VELY2=DELY2/TIME

      C NOW ESTIMATE WHERE THE THIRD OBJECT SHOULD BE

0021          ESTX13=PBUB(1,1)+VELX2*(2*TIME)
0022          ESTX23=PBUB(1,2)+VELX2*TIME
0023          ESTY13=PBUB(2,1)+VELY2*(2*TIME)
0024          ESTY23=PBUB(2,2)+VELY2*TIME
0025          EAV(1)=(ESTX13+ESTX23)/2
0026          EAV(2)=(ESTY13+ESTY23)/2
0027          ERRX3=PBUB(1,3)-EAV(1)
0028          ERRY3=PBUB(2,3)-EAV(2)

      C DEVELOPE ESTIMATION INFO FOR FOURTH OBJECT

0029          IMAG=(ERRX3*(TIME2/TIME))**2+(ERRY3*(TIME2/TIME))**2
0030          VEC3=SQRT(FLOAT(IMAG))
0031          IF(ERRX3.EQ.0)GOTO 100
0032          ANG2=ATAN(FLOAT(ERRY3/ERRX3))
0033          GOTO 110
0034     100  ANG2=3.14/2

      C       FIND THE VELOCITY OF OBJECT IN IMAGE THREE

0035     110  VELX3=DELX3/TIME
0036          VELY3=DELY3/TIME

      C       NOW ESTIMATE WHERE THE OBJECT WILL BE

0037          ESTX14=PBUB(1,1)+(VELX2+VELX3)*TIME+(VELX2+VELX3)/2*TIME2
0038          ESTX24=PBUB(1,2)+VELX3*TIME+(VELX2+VELX3)/2*TIME2
0039          ESTX34=PBUB(1,3)+(VELX2+VELX3)/2*TIME2
0040          ESTY14=PBUB(2,1)+(VELY2+VELY3)*TIME+(VELY2+VELY3)/2*TIME2
0041          ESTY24=PBUB(2,2)+VELY3*TIME+(VELY2+VELY3)/2*TIME2
0042          ESTY34=PBUB(2,3)+(VELY2+VELY3)/2*TIME2
0043          EAV(3)=(ESTX14+ESTX24+ESTX34)/3+VEC3*COS(ANG2)
0044          EAV(4)=(ESTY14+ESTY24+ESTY34)/3+VEC3*SIN(ANG2)

0045          GOTO 600

0046     500  DO 630 I=1,4
0047            PCAM(I)=PBUB(I,3)
0048     630  CONTINUE
0049          PCAM(1)=(PCAM(1)*2)-256
0050          PCAM(2)=(PCAM(2)*2)-240
0051          CALL PCRP(PCAM,FOB,5,PCUR,105)
```

```
C       FILE RETDAT.FTN   REVISED 18 SEPTEMBER 1984
C
C       THIS FILE CONTAINS THE SUBROUTINE
C                  RETDAT.FTN
C
C
C*********************************************************************
C*********************************************************************
      SUBROUTINE RETDAT(TIME,TIME2,PBUB,P,PREV,PCUR,T05,PCAM,POBJ5,C)
C THIS SUBROUTINE WILL PROVIDE AN ESTIMATE OF THE POSITION
C OF THE OBJECT AT SOME SPECIFIC POINT IN TIME
C       VELX2,VELY2   VELOCITIES BETWEEN IMAGES 1 AND 2
C       ESTX13,ESTY13 - ESTIMATE OF OBJECT IN IMAGE 3 USING
C                       POSITION IN IMAGE 1 AS INITIAL POSITION
C       ESTX23,ESTY23 - ESTIMATE OF OBJECT IN IMAGE 3 USING
C                       POSITION IN IMAGE 2 AS INITIAL POSITION
      INTEGER VELX2,VELY2,ESTX13,ESTX23,ESTY13,ESTY23
C       EAV(4) - ESTIMATE FOR OBJECT AT POSITION 3
C       ERRX3,ERRY3 - ERROR BETWEEN ESTIMATE AND OBSERVATION FOR
C                       POSITIION 3

      INTEGER EAV(4),ERRX3,ERRY3
C       THESE ARE THE ESTIMATE OF POSITION 4 USING DIFFERENT
C                       INITIAL POSITIONS
      INTEGER ESTX14,ESTX24,ESTX34,ESTY14,ESTY24,ESTY34
C       VELX3,VELY3 - VELOCITY VALUES BETWEEN IMAGES 2 AND 3

      INTEGER VELX3,VELY3

C       THESE ARE THE POSITION CHANGES BETWEEN THE FRAMES

      INTEGER DELX2,DELX3,DELY2,DELY3,DELTAX,DELTAY
      INTEGER C
      REAL T05(4,4),PREV(4),PCUR(4),PCAM(4),POBJ5(4)
      REAL PBUB(4,4),P(4),TIME,TIME2

C       INITIALIZE CONSTANTS

      C=1.0

C       FIND THE POSITION CHANGES BETWEEN IMAGES 1 AND 2,
C                       AND ALSO BETWEEN IMAGES 2 AND 3

      DELX2=PBUB(1,2)-PBUB(1,1)
      DELY2=PBUB(2,2)-PBUB(2,1
      DELX3=PBUB(1,3)-PBUB(1,2)
      DELY3=PBUB(2,3)-PBUB(2,2)
      DELTAX=PBUB(1,3)-PBUB(1,1)
      DELTAY=PBUB(2,3)-PBUB(2,1)


C       LITTLE CHANGE TREAT AS NO CHANGE
```

```
0038      253   DO 40 I=YCENT-COUNTY,YCENT+COUNTY
0039              IZ2=RPIXEL(XCENT+XLTH/4,I)
0040              IF(IZ2.EQ.0)GOTO 40
0041              COUNT2=COUNT2+1
0042              LAST2=I
0043       40   CONTINUE
0044            IF(COUNT2.LT.3)GOTO 254
0045            YCENT2=LAST2-(COUNT2/2)

0046      254   YCENT=(YCENT1+YCENT1+YCENT2)/3

0047            RETURN
0048            END
```

```
C ************************************************************
C ************************************************************

C        THIS ROUTINE IS USED TO PROVIDE A MORE
C        ACCURATE ESTIMATION OF THE OBJECT'S CENTER

         SUBROUTINE ACENT(XCENT,YCENT,COUNTY,XLTH)

         INTEGER XCENT,YCENT,XLTH,COUNTY
         INTEGER RPIXEL,XCENT1,XCENT2,YCENT1,YCENT2
         INTEGER COUNT1,COUNT2,LAST1,LAST2

C INITIALIZATION

         COUNT1=0
         COUNT2=0

         XCENT1=XCENT
         XCENT2=XCENT
         YCENT1=YCENT
         YCENT2=YCENT

         DO 10 I=XCENT-XLTH,XCENT+XLTH
           IZ1=RPIXEL(I,YCENT-COUNTY/4)
           IF(IZ1.EQ.0)GOTO 10
           COUNT1=COUNT1+1
           LAST1=I
   10    CONTINUE
         IF(COUNT1.LT.3)GOTO 251
         XCENT1=LAST1-(COUNT1/2)

  251    DO 20 I=XCENT-XLTH,XCENT+XLTH
           IZ2=RPIXEL(I,YCENT+COUNTY/4)
           IF(IZ2.EQ.0)GOTO 20
           COUNT2=COUNT2+1
           LAST2=I
   20    CONTINUE
         IF(COUNT2.LT.3)GOTO 252
         XCENT2=LAST2-(COUNT2/2)

  252    XCEN1=(XCENT+XCENT1+XCENT2)/3

         COUNT1=0
         COUNT2=0

         DO 30 I=YCENT-COUNTY,YCENT+COUNTY
           IZ1=RPIXEL(XCENT-XLTH/4,I)
           IF(IZ1.EQ.0)GOTO 30
           COUNT1=COUNT1+1
           LAST1=I
   30    CONTINUE
         IF(COUNT1.LT.3)GOTO 253
         YCENT1=LAST1-(COUNT1/2)
```

```
0055          FBUB(1,2)=(FBUB(1,2)-256)
0056          FBUB(2,3)=(FBUB(2,3)-240)
0057          RETURN
0058          END
```

```
C          EACH SECTION(256X240 PIXELS)

      300        DO 210 Y=D2*240,239+D2*240,10
                   DO 220 X=D1*256,255+D1*256,5
                     Z=RPIXEL(X,Y)
                     IF(Z.NE.0)GO TO 215
                     GO TO 220

      215            TEMPX=X
                     TEMPY=Y
                     DO 230 X1=TEMPX,TEMPX+40
                       Z1=RPIXEL(X1,TEMPY)
                       IF(Z1.NE.0)GO TO 230
                       IF(X1.LT.TEMPX+2)GO TO 220
                       XLTH=X1-TEMPX
                       XCENT=XLTH/2+TEMPX
                       GO TO 240
      230            CONTINUE
      240            DO 250 Y1=TEMPY-(XLTH/2)-15,TEMPY+(XLTH/2)+15
                       Z2=RPIXEL(XCENT,Y1)
                       IF(Z2.EQ.0)GO TO 250
                       COUNTY=COUNTY+1
                       LASTY=Y1
      250            CONTINUE

                     IF(COUNTY.LT.2)GO TO 251
                     YCENT=LASTY-(COUNTY/2)

                     CALL ACENT(XCENT,YCENT,COUNTY,XLTH)
                     DR(IND)=XCENT
                     DR(IND+1)=YCENT
                     IND=IND+2
                     NBUB=NBUB+1
                     COUNTY=0
                     IF(NBUB.EQ.1)GOTO 102
                     IF(NBUB.EQ.2)GOTO 100
                     IF(NBUB.EQ.3)GOTO 450
      251            COUNTY=0

      220          CONTINUE
      210        CONTINUE
      102      CONTINUE
      100    CONTINUE


C BUILD PBUB MATRIX

      450    DO 830 I=1,NBUB

               PBUB(1,I)=FLOAT(DR(2*I-2))
               PBUB(2,I)=FLOAT(DR(2*I-1))
               PBUB(3,I)=0.0
               PBUB(4,I)=1.0

      830    CONTINUE
```

```
C
C       FILE F3B.FTN   REVISED 20 AUGUST 1984
C
C       THIS FILE CONTAINS THE SUBROUTINE
C                   F3B.FTN
C
C
C*********************************************************************
C*********************************************************************
C       FILE F3B.FTN   REVISED 20 AUGUST 1984
C
C       THIS FILE CONTAINS THE SUBROUTINE
C                   F3B.FTN
C
C
C*********************************************************************
C*********************************************************************
        SUBROUTINE F3B(FBUB,NBUB)
C THIS SUBROUTINE LOCATES THE POSITION OF THE OBJECTS
C WITHIN THE FRAME BUFFER AND REFERS ALL POSITIONS TO
C THE TOP LEFT QUADRANT . THIS
C ROUTINE WAS UPDATED TO BE MORE ACCURATE WITH THE
C CENTER FINDING TECHNIQUE BY USING THE SUBROUTINE
C 'ACENT' WHICH FINDS SEVERAL VALUES FOR CENTER COOR-
C DINATES AND AVERAGES THEM TO DERIVE THE FINAL VALUE.

        INTEGER X,Y,RPIXEL,Z,I

C       TEMPX,TEMPY - INITIAL BOUNDARY COORDINATES
C       XLTH - LENGTH OF INITIAL CHORD MEASURED ON OBJECT

        INTEGER TEMPX,TEMPY,Z1,Z2,XLTH

C       COUNTY - LENGTHH OF INITIAL CHORD IN Y DIRECTION
C       LASTY - LOCATION OF LAST OBJECT PIXEL ON INITIAL CHORD
C       XCENT,YCENT - ESTIMATES OF CENTER COORDINATES

        INTEGER XCENT,YCENT,COUNTY,LASTY

C       DR - ARRAY CONTAINING CENTER COORDINATES
C       IND - INDEX TO SPECIFIC VALUES IN DR ARRAY
C       NBUB - NUMBER OF OBJECTS FOUND

        INTEGER DR(0:10),IND,NBUB
        INTEGER Y1,X1,D1,D2

C       FBUB - REAL ARRAY CONTAINING OBJECT VECTORS IN CAMERA COOR.

        REAL FBUB(4,4)
C       INITIALIZE CONSTANTS
        NBUB=0
        IND=0
C       START PROCESSING THE FRAME BUFFER

        DO 100 D2=0,1
          DO 102 D1=0,1
```

```
0001          SUBROUTINE DELAY(N)
        C SUBROUTINE DELAY WILL PROVIDE A DELAY OF N SECS.
0002          INTEGER L,M
0003          REAL N
0004          M=60*N
0005          DO 5 L=1,M
0006            CALL DVR
0007        5 CONTINUE
0008          RETURN
0009          END
```

```
C THIS SUBROUTINE WILL PRODUCE THREE PICTURES IN A
C SINGLE FRAME BUFFER.

      SUBROUTINE TSNAPS(TIME,TIME2,FBUB,NBUB)
      INTEGER NBUB
      REAL TIME,TIME2,FBUB

C INITIIALIZE CONSTANTS

C TAKE THREE PICTURES 'TIME' SECS. APART AND STORE
C IN FRAME BUFFER.

      CALL PAN(0)
      CALL SCROLL(0)
      CALL ZOOM(1,1)
      CALL SNAP

      CALL PAN(256)
      CALL VIEW
      CALL DELAY(TIME)
      CALL SNAP


      CALL PAN(0)
      CALL SCROLL(240)
      CALL VIEW
      CALL DELAY(TIME)
      CALL SNAP


      CALL PAN(0)
      CALL SCROLL(0)
   50 CALL F3B(FBUB,NBUB)
      RETURN
      END
```

```
0038              CALL TSNAPS(TIME,TIME2,PBUB,NBUB)
0039              CALL RETIAT(TIME,TIME2,PBUB,P,PREV,PCUR,T05,PCAM,PDBJ5,C)
0040              IF(C.EQ.0.0)GOTO 180
0041              DO 175 J=1,4
0042                PO(J)=PREV(J)
0043      175     CONTINUE
0044              GOTO 190
0045      180     CALL FORP(P,P5,PO,T05)

                UPDATE PREV VALUES

0046              PREV(1)=PO(1)
0047              PREV(2)=PO(2)


0048      190     CALL MOVE(PO,T05,THETA,STAT)
0049              CALL WAIT(TIME)
0050      200   CONTINUE


0051            STOP
0052            END
```

```
C       PACKAGED SOFTWARE(IMAGE TECHNOLOGY,INC.)

0006          CALL SELGRF(1)
0007          CALL SYNC(0)
0008          CALL FBINIT
0009          CALL APINIT
0010          CALL LUINIT
0011          CALL VIDCHN(0)

C       INITIALIZE A BINARY LUT(S'THRES'=0,BLACK)

0012          WRITE(5,999)
0013     999  FORMAT(1X,' PLEASE INPUT THRES FOR ZERO INTENSITY')
0014          READ(5,998)THRES
0015     998  FORMAT(I3)
0016          CALL SELLUT(0,0)
0017          DO 1 I=0,THRES
0018            CALL SETLUT(I,0)
0019       1  CONTINUE
0020          DO 2 I=THRES+1,255
0021            CALL SETLUT(I,255)
0022       2  CONTINUE

C       SELECT OUTPUT CHANNEL AND VIEW

0023          CALL SELLUT(1,0)
0024          CALL VIEW

C       BUILD INITIAL T05 MATRIX

0025          CALL MOVE(PHOME,T05,THETA,STAT)

C       INPUT TIME BETWEEN INITIAL SNAPSHOTS

0026          WRITE(5,99)
0027      99  FORMAT(1X,'INPUT TIME(IN SECS.) BETWEEN IMAGES.')
0028          READ(5,10)TIME
0029      10  FORMAT(F10.4)

C       INPUT TIME ESTIMATION POINT
0030          WRITE(5,98)
0031      98  FORMAT(1X,'INPUT TIME(IN SECS.) FOR FINAL POSITION.')
0032          READ(5,10)TIME2

C       INITIALIZE ROBOT POSITION FOR CATCH UP MODE

0033          PREV(1)=90.0
0034          PREV(2)=0.0
0035          PREV(3)=60.0
0036          PREV(4)=1.0


C       START TO TRACK OBJECT

0037          DO 200 I=1,100
```

APPENDIX B: COMPUTER PROGRAM LISTING

```
        C       THIS PROGRAM CONTROLS THE IMAGE PROCESSING AND
        C       ROBOTIC MOVEMENT FOR A SIMPLE TRACKING APPLI-
        C       CATION OF THE P5 INDUSTRIAL ROBOT. THE SYSTEM
        C       IMPLEMENTS A TYPE OF VISUAL CONTROL BY EXTRACTING
        C       POSITIONAL DATA THROUGH THE USE OF AN IMAGE
        C       ACQUISITION SYSTEM.
        C       LAST REVISED ON DECEMBER 05,1984

        C       MAURICE F. O'DONNELL

        C       NECESSARY VARIABLES

        C       STAT   FLAG TO SIGNAL PROPER COMPLETION OF MOVE
        C       NBUB   VALUE OF NUMBER OF OBJECTS FOUND
        C       THRES - THRESHOLD VALUE FOR BINARY IMAGE
        C       C - FLAG DENOTING CATCH UP ROUTINE IN EFFECT

0001            INTEGER STAT,NBUB,THRES,C

        C       TIME - TIME PERIOD BETWEEN SUCCESSIVE IMAGES
        C       TIME2 - ESTIMATION TIME PERIOD
        C       PHOME - ARRAY CONTAINING THE 'HOME' VECTOR FOR P5
        C       T05(4,4) - GENERAL TRANFORMATION MATRIX FROM LINK 5
        C                  COORDINATES TO BASE (0) COORDINATES
        C       TC5(4,4) - GENERAL TRANSFORMATION MATRIX FROM CAMERA
        C                  COORDINATES TO LINK 5 COORDINATES

0002            REAL TIME,TIME2,PHOME(4),T05(4,4),TC5(4,4)
        C       THETA(5) - ARRAY CONTAINING THE FIVE ANGLES WHICH DISCRIBE
        C                       THE ORIENTATION AND POSITION OF THE P5
        C       PREV - ARRAY OF VALUES THAT HOLD THE PREVIOUS POSITION OF
        C                       THE ROBOT IN BASE COORDINATES. THESE VALUES
        C                       ARE USED WHEN THE OBJECT IS MOVING VERY SLOW
        C       PCUR - ARRAY USED BY SUBROUTINE TO STORE CURRENT POSITION
        C                       OF OBJECT IN BASE COORDINATES
        C       PCAM - ARRAY USED BY SUBROUTINE TO HOLD LAST IMAGE COORDINATES
        C       POBJ5 - ARRAY PCAM TRANSFORMED INTO LINK 5 COORDINATES
        C       P0 - THIS ARRAY CONTAINS THE ESTIMATED POSITION OF THE
        C                       OBJECT AT THE ESTIMATION TIME IN BASE COOR-
        C                       DINATES

0003            REAL THETA(5),PCUR(4),POBJ5(4),PCAM(4),PREV(4),P0(4)
        C       PBUB(4,4) - ARRAY CONTAINING THE CAMERA COORDINATES OF
        C                       THE MOVING OBJECT
        C       P5(4) - ARRAY CONTAINING THE ESTIMATED POSITION OF THE
        C                       MOVING OBJECT IN LINK 5 COORDINATES
        C       P(4) - ARRAY CONTAINING THE ESTIMATED POSITION OF THE
        C                       MOVING OBJECT IN CAMERA COORDINATES

0004            REAL PBUB(4,4),P5(4),P(4)

        C       INITIALIZE CONSTANTS

0005            DATA PHOME/90.0,0.0,60.0,1.0/

        C       INITIALIZE IMAGING EQUIPMENT USING
```

```
      WRITE(5,960)
960   FORMAT(' ' /' ATA * ATAI'/)
      CALL     PRIMAT(ITST, 3, 3, 5)


999   CLOSE( UNIT=3 )
      STOP
      END
```

```
C           PRINT OUT UPDATED DATA
0079        DO 90 JJ=1,12
0080        WRITE(5,91)A(JJ,1),A(JJ,2),A(JJ,3),B(JJ,1),B(JJ,2)
0081   91   FORMAT(' ',3F10.5,7X,2F10.5)
0082   90   CONTINUE

C           SAVE IN PARTIAL RESULTS FILE
0083        WRITE(3,99)A(J,1),A(J,2),A(J,3),B(J,1),B(J,2)
0084   99   FORMAT(' ',5F15.8)

0085   400  CONTINUE


0086        WRITE(5,17)
0087   17   FORMAT(//)


0088        CALL    MOVE(PHOME0,T05,THETA,STAT)


C           SOLVE FOR COEFFICIENTS
0089        CALL    LSTSQS(A,X,B,AT,ATA,ATAI,ATAIAT,ATASYM,12,3,2,IER)


C           OUTPUT COEFFICIENTS TO TTY AND CAMCON.DAT FILE
0090        WRITE(5,200)X(1,1),X(2,1),X(3,1),X(1,2),X(2,2),X(3,2)
0091   200  FORMAT(' T5C(1,1)  = ',F15.8/
            1       ' T5C(1,2)  = ',F15.8/
            1         T5C(1,4)  = ',F15.8/
            1       ' T5C(2,1)  = ',F15.8/
            1         T5C(2,2)  = ',F15.8/
            1       ' T5C(2,4)  = ',F15.8/)


0092        CALL    ASSIGN(2,'SY:CAMCON.DAT',13)
0093        WRITE(2,210)X(1,1),X(2,1),X(3,1),X(1,2),X(2,2),X(3,2)
0094   210  FORMAT(6(F15.8/))
0095        CLOSE( UNIT=2 )

C           CALCULATE MEAN-SQUARE ERROR
0096        CALL    MATMUL(AXHAT,A,X,12,3,2)
0097        MSERRX=0.0
0098        MSERRY=0.0
0099        DO 500 II=1,12
0100        MSERRX=MSERRX+(B(II,1)-AXHAT(II,1))**2
0101        MSERRY=MSERRY+(B(II,2)-AXHAT(II,2))**2
0102   500  CONTINUE
0103        MSERRX=MSERRX/12.0
0104        MSERRY=MSERRY/12.0
0105        WRITE(5,550)MSERRX,MSERRY
0106   550  FORMAT(' MEAN-SQUARE ERROR IN X = ',F15.5/
            1         MEAN-SQUARE ERROR IN Y = ',F15.5)

C           CHECK OUT INVERSION OF ATA MATRIX
0107        CALL    MATMUL(ITST, ATA, ATAI, 3, 3, 3)
```

```
C       COMPARE PREVIOUS POSITION WITH CURRENT POSITION

        IF((ABS(PCUR(1)-PREV(1)).LT.2.0).AND.
     &  (ABS(PCUR(2)-PREV(2)).LT.2.0))GOTO 583

        PREV(1)=PCUR(1)
        PREV(2)=PCUR(2)

  583   P(1)=PREV(1)
        P(2)=PREV(2)
        P(3)=60.0
        P(4)=1.0

        WRITE(5,26)
   26   FORMAT(1X,'CATCH UP MODE')
        GOTO 610
C FORM THE P VECTOR

  600   P(1)=(FLOAT(EAV(3))*2)-256
        P(2)=(FLOAT(EAV(4))*2)-240
        P(3)=0.0
        P(4)=1.0
  610   RETURN
        END
```

```
C       THIS ROUTINE FORMS THE NECESSARY 'P' VECTOR TO MOVE ROBOT
0001    SUBROUTINE FORP(P,P5,P0,T05)
0002    REAL P(4),T05(4,4),P5(4)
C       ASSIGN LUN 2 TO CAMCON.DAT
0003    CALL ASSIGN(2, SY:CAMCON.DAT',13)
C       GET CAM-TO FOR CONSTANTS
0004    READ(2,*)A,B,X0,C,D,Y0
0005    CLOSE(UNIT=2)
0006      P5(1)=A*P(1)+B*P(2)+X0
0007      P5(2)=C*P(1)+D*P(2)+Y0
0008      P5(3)=0.0
0009      P5(4)=1.0
0010    CALL MATMUL(P0,T05,P5,4,4,1)
0011    RETURN
0012    END
```

END

FILMED

4-85

DTIC