EVALUATION OF VIDEO-CAMERA CONTROLS FOR REMOTE MANIPULATION

by

REAL FRENETTE

B.A.Sc., Laval University, 1982

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

(Department Of Electrical Engineering)

We accept this thesis as conforming

to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

February 1985

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Electrical Engineering

The University of British Columbia
1956 Main Mall
Vancouver, Canada
V6T 1Y3

Date 14 February 1985

Abstract

The control of the video-camera plays an important factor in the overall efficiency of a teleoperator system. A computer-based video-camera control has been designed to compare and evaluate four different modes of control. A situation where an operator does not have a free hand for the control of the video-camera has been selected: such a situation can be found in subsea applications where the operator is required to steer a submarine and to manipulate a robot arm.

The four modes are:

- manual control mode : The operator's right hand is used to control both the robot arm and the camera system. The orientation of the camera (with close-up lens) is performed by pressing push buttons.
- automatic tracking mode : The camera (with close-up lens) automatically tracks the end effector of the slave arm, without direction from the operator.
- voice-operated mode : The orientation of the camera (with close-up lens) is accomplished by spoken commands.
- fixed-camera-position mode : A wide angle lens is used in this mode. The camera constantly remains in a straight ahead position and no controls are required.

A tracking task and a pick-and-drop task were performed during the experiments. Measures of speed and accuracy were

taken and analyzed; subjective remarks were also gathered.

Results showed significant differences between the modes. Specifically, automatic tracking mode and voice-operated mode were found to offer the best ergonomic environment for the operator in terms of speed-accuracy tradeoff.

taken and analyzed; subjective remarks were also gathered.

Results showed significant differences between the modes. Specifically, automatic tracking mode and voice-operated mode were found to offer the best ergonomic environment for the operator in terms of speed-accuracy tradeoff.

Table of Contents

List Of Tables

List Of Figures

Acknowledgements

I would like to thank my supervisor Dr P.D. Lawrence for his support and encouragement during the course of this work. I am also very thankful to my co-supervisor Dr Terry Peace from Robotic Systems International (RSI) for his dedication toward this project.

I would also like to thank RSI for their cooperation and for giving me access to their facilities and manpower.

Finally, I would like to express my gratitude towards the following people for their technical assistance: Ken Madore and Tony Leugner from the Electrical Engineering Department, Ken Soles and Allan Rylandsholm from RSI.

I dedicate this thesis to my parents, to whom I am very grateful.

This research has been supported by the Natural Sciences and Engineering Research Council of Canada through a Postgraduate Scholarship to its author and through a Research Grant #4924, and by Robotic Systems International.

I. INTRODUCTION

Robotics has become a field of increasing interest in recent years. In many instances, autonomous robots have replaced human workers to perform repetitive and well structured tasks. Good examples of this phenomenon can be found in assembly lines (e.g. car manufacturing industry). In such situations, the robot performs the task (e.g. welding, assembling parts, etc.) on its own, without any cooperation from a human operator.

Assembly line utilization of robots covers only one area of robotics. There is yet another aspect to robotics, the benefit of which has also been felt in today's industry, and for which current research is being carried out to meet present and future needs. This branch of robotics is commonly referred to by the term "teleoperation".

A teleoperator system is one that combines both man and machine capabilities into an integrated engineering system [12]. As can be seen in Figure 1.1, it is essentially a man-machine system whereby a human operator remotely controls a mechanical arm through a communication channel. The minimum information required by the operator to perform a given task with the mechanical arm must be of a visual nature, since the operator's workstation and the mechanical device are located in two separate environments. The control of the remote mechanical arm

is usually achieved through a scale model of the remote arm: the commands sent to move the remote arm are such that any motions of the scale model are reproduced by the remote arm. Such a configuration is called a master/slave system since the control of the remote arm (called the slave arm) originates from the scale model (called the master arm). The slave arm always maintains a spatial correspondence with the master arm.

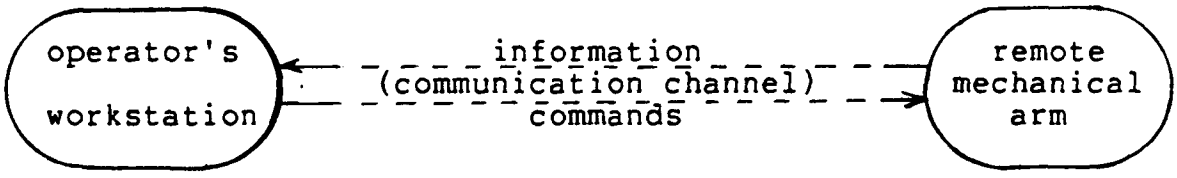


Figure 1.1: Teleoperator system basic constituents

Teleoperator systems have proven their necessity in tasks such as handling of biological, chemical, toxic and radioactive materials [10]. Maintenance and repair of nuclear plants [7] and fuel reprocessing plants [6] are other good examples where teleoperator systems are required to perform a task in environments that are dangerous or inacessible to humans. Also, it is predicted that teleoperators will find invaluable use in space applications for satellite retrieval, servicing or maintenance; deploying or assembling space platforms, large antennas and solar power stations [2,3]. Such work has already started on the USA space shuttle with its Canada Arm. Bejczy describes teleoperation as a means to extend "the manipulative capabilities of the human arm and hand to remote, physically difficult, or dangerous environments" [2].

Subsea is yet another field where teleoperators find suitable applications. As was stated by Marchal, Rouyer and Vertut, "in general, below 300 meters deep, diver intervention, delicate and costly, will become progressively more and more exceptional. Even nearer to the surface, man will be progressively replaced by teleoperators or robots, for productivity, (.....) or safety reasons" [8]. Common types of work performed include cleaning, inspection and non-destructive testing of offshore oil platforms, installation and maintenance of power lines running at the bottom of the ocean, oceanographic work, etc.

The research work presented in this thesis applies to a subsea teleoperator system according to the set-up shown in Figure 1.2. A subsea robot arm along with a camera mounted on a pan-tilt unit are attached to an unmanned submarine. From a surface ship which is linked to the submarine, an operator must control the operation of the:

- robot arm : Through the use of a master arm, the operator moves the slave arm (which is underwater) to have it carry out a certain task.
- pan-tilt unit and camera : The operator must orient the camera according to what is needed (either to look at the robot arm or at the surroundings), as well as adjusting the camera itself (focus, iris, zoom and lights). The complexity of this task depends on the camera system used.
- submarine : The operator is also in charge of the steering

of the submarine. In particular, it is often required that the submarine remains in a fixed position while the arm moves around and this may require regular attention from the user.

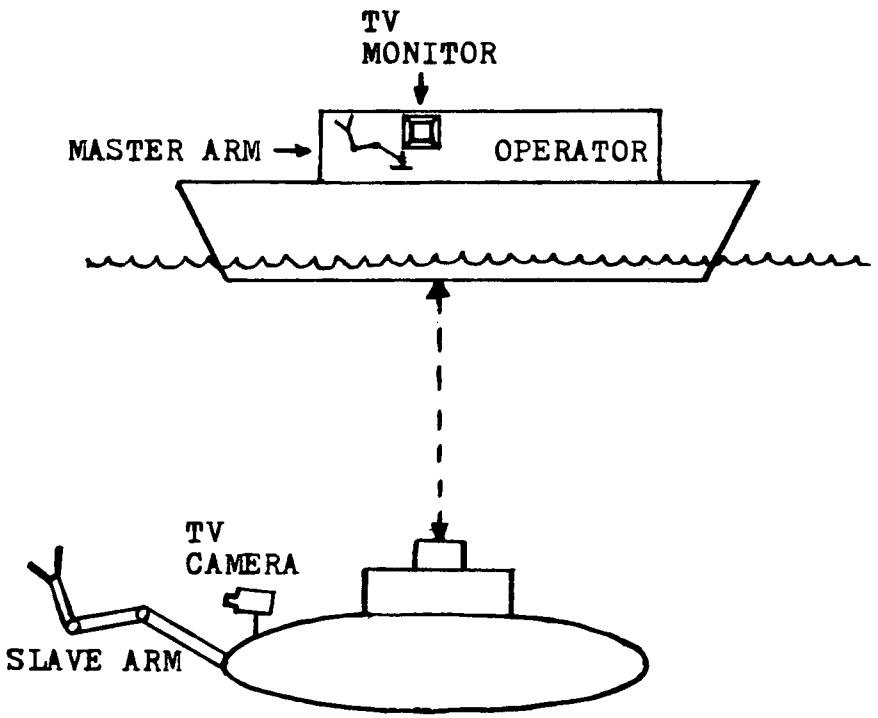


Figure 1.2: Subsea application of teleoperator system

This concise description of the control involved is sufficient to demonstrate that the workload on the operator is enormous. As Bejczy points out, "it is recognized that the human operator's input and output channel capacities are limited [compared to computerized machines]. In this sense, the human operator represents a limiting factor in the information and control environment of a remotely operated robot" [1]. It is

therefore important to assess whether or not the human operator is an essential component in the control of the arm. Replacing the teleoperator system (which necessitates the presence of a human operator in the control loop) by an autonomous robot would offer many obvious advantages.

Human factors

As mentioned in the beginning, autonomous robots, i.e. those operating without a human operator, can only be used when the conditions are well enough defined and structured. However, one is faced here with undefined and unpredictable conditions as well as non-repeatable problems. Computerized controls (using artificial intelligence) can assist the operator, but "his unique capability to reason and adapt to changing conditions cannot be replaced with near-term available computer systems" [10]. Therefore, the human operator is an essential key control element in remote applications of robots for it is his presence in the control loop that provides versatility, analysis and decision capabilities which are necessary in a highly variable and unpredictable situation [1,10].

With a human operator in the control loop, a man-machine interface must be designed in order to enhance and extend his capability through the machine. The design of such an interface calls for a careful study of the human factors involved. On the one hand, the operator must be supplied with enough sensory information coming from the remote site to enable him to project

himself into that remote environment: such information can include visual feedback, force feedback, proximity sensors, touch and slip sensors, etc. All this information must be easily available to the operator. On the other hand, the interface should also facilitate any activities performed in the remote site: manipulation of the mechanical arm, modification of the pan-tilt unit and camera, lights, etc. In other words, the interface must enhance the control of the teleoperator system, while increasing task efficiency [9,11].

Video system

The video system constitutes a very important part of a teleoperator system. Improving the visual information as well as the control of the video system can result in a significant enhancement of the overall system efficiency. Afterall, it is through that source of information that the operator gets most of his knowledge of the remote site.

Three aspects of the video system need to be looked at in the design of the man-machine interface: the location of the camera, the choice of the camera lens, and the kind of remote control of the pan-tilt unit (and of the lens). The camera can be either attached to the end effector of the mechanical arm (and therefore is mobile), or it can be mounted on a pan-tilt unit which is, in turn, attached in a fixed position relative to the base of the arm. The latter case was considered for the matter of this research work (see Figure 1.2).

The choice of the camera lenses consists of three types: wide angle lenses, close-up lenses and zoom lenses. The difference lies in their respective focal length or, in other words, in the field of vision that they exhibit: the field of vision of a wide angle lens is much larger than that of a close-up lens, since its focal length (i.e. wide angle lens) is smaller. The range of focal length covered by a zoom lens can include both the wide angle lens and the close-up lens. Changing the focal length of a zoom lens, however, results in an additional control for the operator to look after. A direct implication of the field of vision of the camera lens is seen in the need for changing the camera orientation and in the frequency of the changes as the slave arm is moved around: the larger the field of vision, the less frequent the changes. In particular, a wide angle lens with a small enough focal length could free completely the operator from having to move the pan-tilt unit at any time. Thus, there exists an important tradeoff between the focal length of the camera lens and the frequency of pan-tilt unit re-orientations. It is important to note that a smaller field of vision provides more details of a specific region than does a larger field of vision.

The video system requires control of its pan-tilt unit (i.e. modifying the orientation if necessary), the camera (i.e. focus, iris or zoom depending on the complexity of the lens being used), and the lighting (if necessary). There are many ways of achieving the overall control of the video system. It

is important at this stage to point out that the operator in the situation considered in this study has both hands in full use: one hand operates the master arm while the other looks after the steering of the submarine. In addition, his visual attention is tied to the video monitor while performing any tasks.

A common type of control for the video system consists of a joystick and/or push buttons. With such a control, the operator, desiring to bring a modification on the video system, must first visually locate the joystick or pushbuttons; the operator then freezes the control of one hand (either for the submarine or the slave arm), brings his hand over to the joystick or pushbuttons and proceeds with the modification. Such a process causes an interruption in the task in progress, diverts the operator's visual attention from the video monitor and manual work, and distracts his mental concentration. "All these can contribute to lengthening the whole operation and to increasing operator workload" [4]; furthermore, Bejczy adds that it "often renders the whole operation inflexible and inefficient" [1].

Voice control

Speech recognition seems like one reasonable solution to this inefficient man-machine interaction. It has the advantage of using a most natural mode of communication for humans, and it offers an open and direct communication channel that does not require any manual or specific visual contact between man and

machine [1]. Its advantages in teleoperator stations are obvious in situations where the operator has his hands and eyes busy, as is the case here.

However, today's speech recognition technology still falls short of the capability of man. Much research is being done to make continuous speech recognizers (i.e. those that do not require pauses between words) more affordable and more accurate: they are, as yet, impractical in teleoperator environments. Computer-based word recognizers have been on the market for a long time and have become suitable for control applications. These systems require pauses between words. For a large vocabulary capacity (i.e. 20 words and more), only speaker-dependent systems are available: by this, one means that the recognizer needs to be trained to each user's voice beforehand.

The accuracy obtained by speaker-dependent word recognizers can be very good (i.e. > 99%) on the condition that one adapts one's speech to the restrictions of the recognizer. Besides having to introduce pauses between two consecutive words, one must be very consistent in one's speech: too much variation in the emotional tone, loudness, speech rate, etc. can result in misrecognitions. Also, changes in the backgroud noise can lead to problems. As was stated by P.E. Van Hemel, S.B. Van Hemel and W.J. King, "the requirement for adaptations and new behaviors by the ASR user [Automatic Speech Recognizer]

introduces human factors considerations" [15]. How much mental distraction and stress are involved in using such a system? Can a user really get used to those constraints and be able to concentrate his efforts on the tasks ahead of him? Is any kind of feedback from the system necessary and, if so, what type? These questions, and many others, are worth pondering as well as spending time analyzing experimentally.

Much study has been done to give guidelines to using automatic speech recognizers [4,15,16]. Choosing a proper vocabulary can bring a significant contribution to its effectiveness: use of words that are distinct enough from each other; longer words usually lead to better accuracy; concatenated words also offer advantages. The degree of complexity of the syntax used also plays an important role: a more complex syntax increases the system accuracy but at the same time adds restrictions on the user. Discussed also in the referenced studies are the need for feedback to the user (audible or visible) and confirmation from the user before actions are carried out. The training session, when the user gives his word templates to the system, should be performed under similar physical conditions (e.g. background noise) that will be encountered during the actual use of the voice system. Much emphasis must be placed on providing the user with an adequate model of the recognizer so that his speech and expectations may be tailored accordingly.

In itself, using voice control in a teleoperator environment is not new [5]. For example, a study has been done to demonstrate the advantages of voice control on a remote controlled unmanned submarine and to show its feasibility [13]. Also, the feasibility and utility of controlling the Space Shuttle TV cameras and monitors by voice has been investigated [4]. The studies report favorable conclusions regarding the use of speech recognition and bring forth some useful recommendations. However, most of the results are of a qualitative nature and we found no studies that present adequate quantitative measures in comparing speech recognition with other modes of control, while at the same time showing the tradeoffs involved.

It is this lack of data, which are necessary to really help one design a proper man-machine interface, that motivated the study presented herein. This study is not intended to be exhaustive on the subject of voice-controlled video system vs other modes. Certain modes (e.g. foot controls, head-coupled TV system) have been left aside to limit the scope of the study. However, it provides an excellent groundwork for further studies, while furnishing useful and meaningful data.

Purpose of this work

The specific purpose of the research presented in this thesis is to compare four different video system controls in the subsea teleoperator environment described earlier: the same

results can be applied to any similar environment having the same restrictions on the operator (i.e. both hands being busy) and requiring the same kind of manipulation. The four modes are:

- manual control mode : The operator's right hand is used to control both the camera system and the robot arm. The orientation of the camera (with a close-up lens) is performed manually with push buttons.
- automatic tracking mode : The camera (with a close-up lens) automatically tracks the end effector of the slave arm. To this end, the position of the end effector is continually computed (through the reading of the joint angles of the slave arm) with respect to the pan-tilt unit reference coordinate system; then, the computer positions the camera accordingly so that it point towards the end effector [14].
- voice-operated mode : Changes on the camera orientation are accomplished by spoken commands. The camera has a close-up lens.
- fixed-camera-position mode : In this mode, a wide angle lens is mounted on the camera. The operator is then freed from any control of the pan-tilt unit.

Two typical tasks were performed during the experiments. The operator was first asked to have the robot arm follow a certain path of a certain width: this was to simulate tasks like following a seam. Then, a pick-and-drop task was performed to simulate the handling of objects, as is often required.

Measures of elapsed time and errors were obtained for the first experiment showing the tradeoffs between speed and accuracy for the different modes of operation; for the second experiment, only elapsed time was considered. Analyses of the results of the experiments were done to measure differences between the modes, leading to some interesting and significant conclusions. Finally, subjective remarks were gathered.

A video system control was designed to undertake the experiments, and its design is presented in Chapter II. It consists of a computer-based system interfacing a discrete word recognizer to a camera mounted on a pan-tilt unit. Two lenses were available: a wide angle lens and a motorized zoom lens. The latter one was utilized as a close-up lens during the experiments. Note that the system allows one to vary the focal length of the zoom lens using spoken commands even though it was not used during the experiments. Also, the video system could be disconnected from the computer and connected to a manual control box having pushbuttons.

A speech recognizer cannot accomplish anything on its own: it requires the use of a computer to interpret the spoken commands that were recognized and to direct actions accordingly. Adding computer capabilities opens up new horizons for control. In that respect, computerized functions are included in the system such as: automatic tracking of the end effector of the mechanical arm by the camera (which is one of the modes of

operation for the experiments), "memorization" of orientations of the pan-tilt unit and the ability to set the pan-tilt unit back into any of the memorized positions through spoken commands, continuous and discrete motions, variable speed, etc.

II. SYSTEM DESCRIPTION

2.1 General Description Of The Teleoperator System

Figure 2.1 gives an overall view of the laboratory apparatus used to simulate the subsea teleoperator environment described earlier. Figure 2.2 shows its major components with their interaction. The operator's workstation (Figures 2.3, 2.4) is physically separated from the slave arm environment (Figures 2.5, 2.6) where mechanical activities are carried out: the system provides a communication channel in order to supply information of the remote environment to the operator as well as to direct commands from the operator to the diverse components of the remote site (Figure 1.1).

The only source of information supplied to the operator comes from a video system. A camera, mounted on a pan-tilt unit (Figures 2.7, 2.8), conveys visual information over the communication channel: this information is displayed on a video monitor (Figure 2.9) and is available for the operator to look at. The pan-tilt unit sits in a fixed position relative to the base of the slave arm. It is possible for the operator to modify the orientation of the pan-tilt unit: in addition, the features (i.e. focus, iris and zoom) of the motorized zoom lens (Figure 2.7) are remotely controllable. These controls of the video system can be operated by the operator either manually or through spoken commands. In the former case, modifications are

Figure 2.1: Overall view of experimental setup

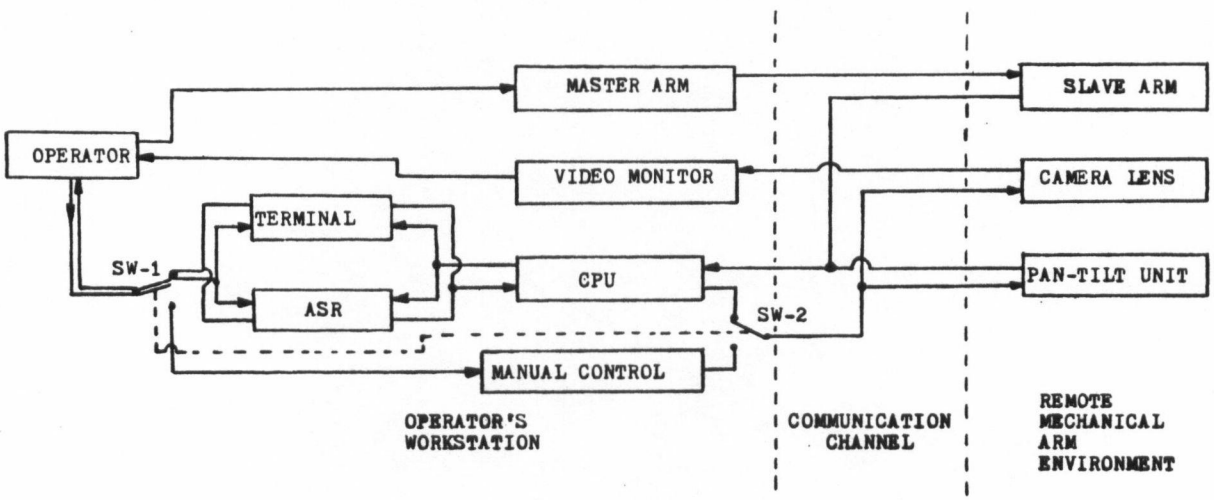


Figure 2.2: Teleoperator system block diagram

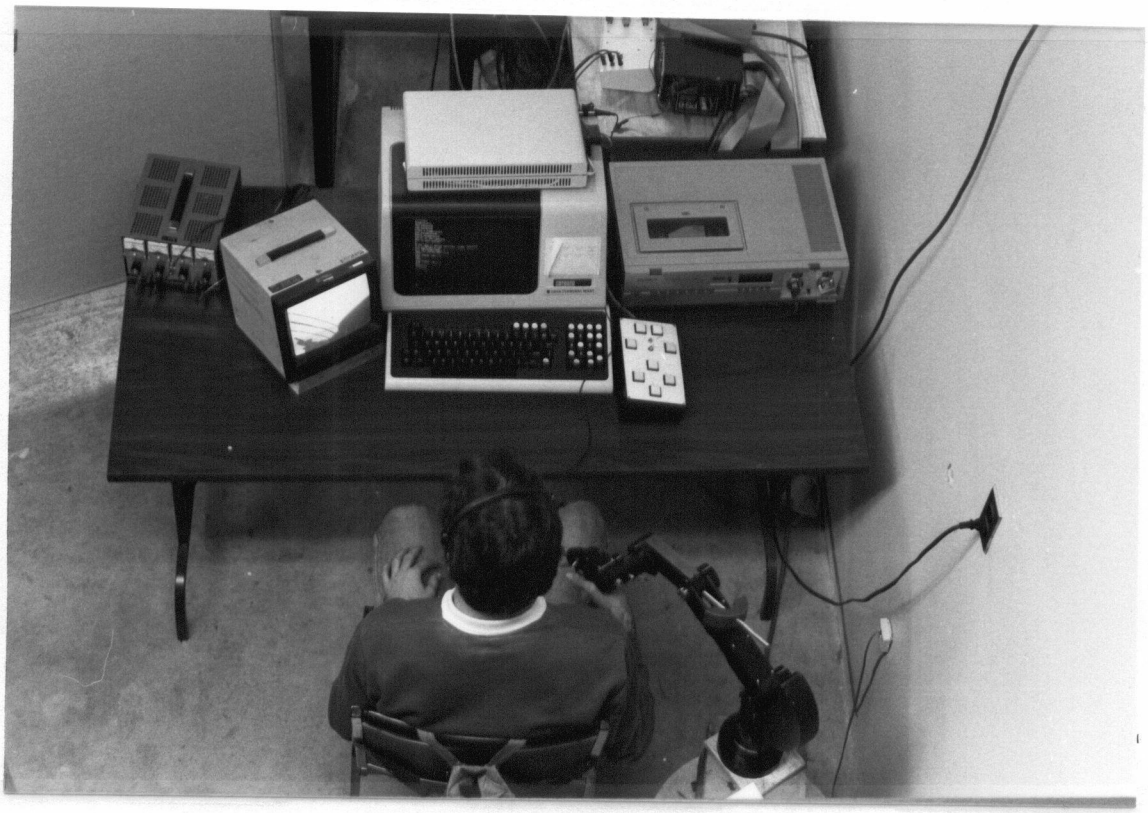


Figure 2.3: Back view of operator's workstation

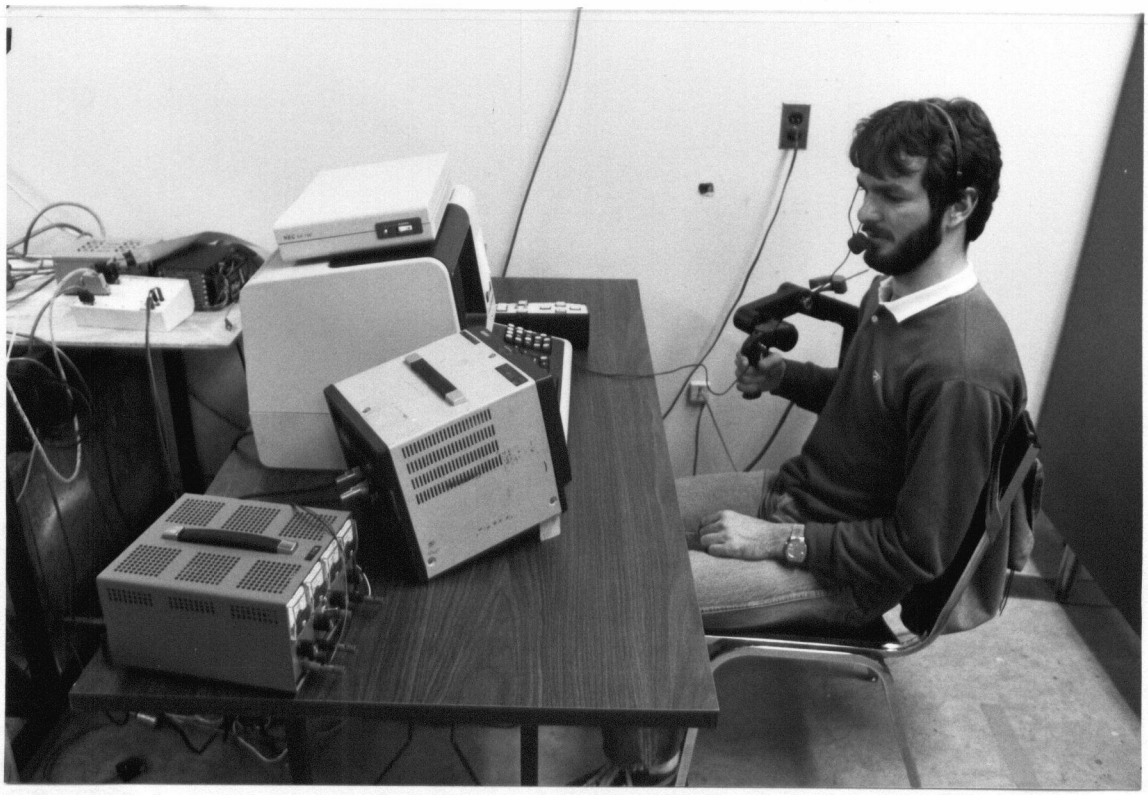


Figure 2.4: Side view of operator's workstation

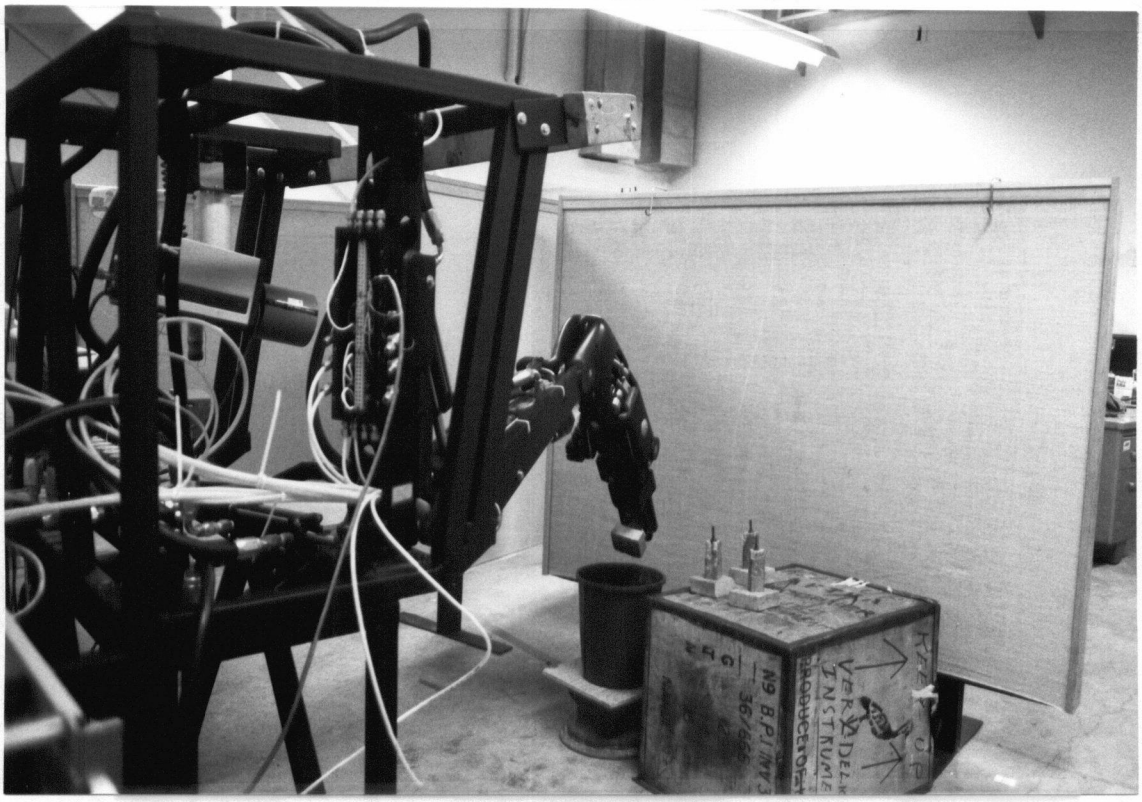


Figure 2.5: Back view of slave arm environment

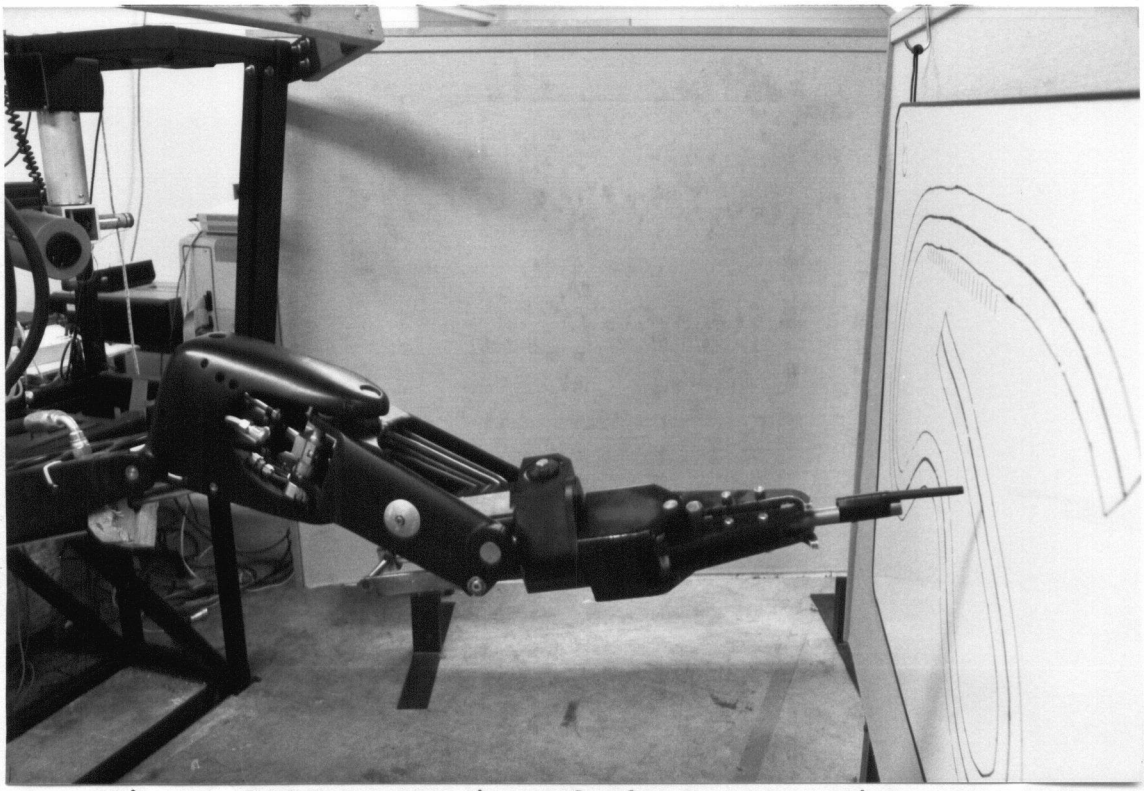


Figure 2.6: Front view of slave arm environment

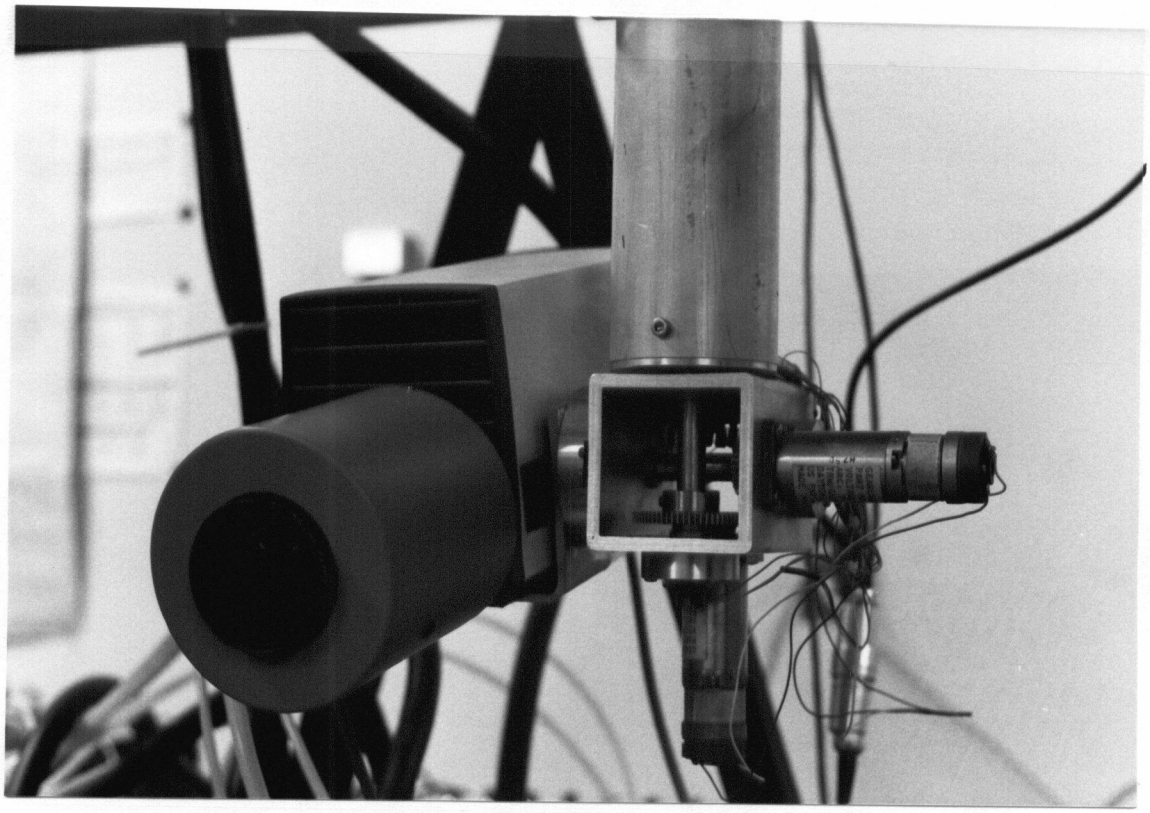


Figure 2.7: Camera/pan-tilt unit (zoom lens)

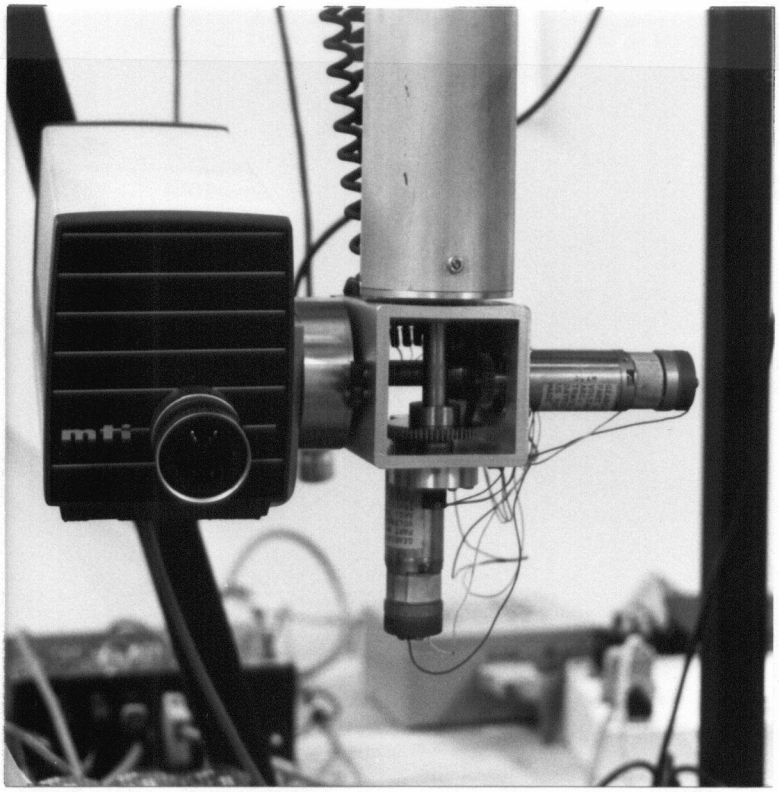


Figure 2.8: Camera/pan-tilt unit (wide angle lens)

Figure 2.9: Video monitor and user's terminal

Figure 2.10: Manual control box

carried out by pressing pushbuttons on the manual control box (Figure 2.10); in the latter case, a computer-based system interfaces a word recognizer (ASR) (Figure 2.4) to the video system which enables the operator to alter the system setting through verbal commands. The choice of controls is determined by the position of the switches shown in Figure 2.2. Note also that a wide angle lens (Figure 2.8) is available for the camera (and was actually used for the experiments): the lens does not possess any remote controls.

This visual information allows one to have some knowledge of the remote site. In particular, one can see the slave arm (if the pan-tilt unit is properly oriented) which enables one to have it perform some mechanical activities. To this end, a master arm (Figures 2.1, 2.3, 2.4) is used, which is a scale model of the slave arm (Figures 2.1, 2.5, 2.6). The master arm and its slave arm are linked together in such a way that any motions of the master arm are replicated onto the slave arm: a spatial correspondence is constantly maintained between the two arms. Thus, the operator can remotely manipulate the slave arm through its master arm and receive visual feedback of the activities through the video system.

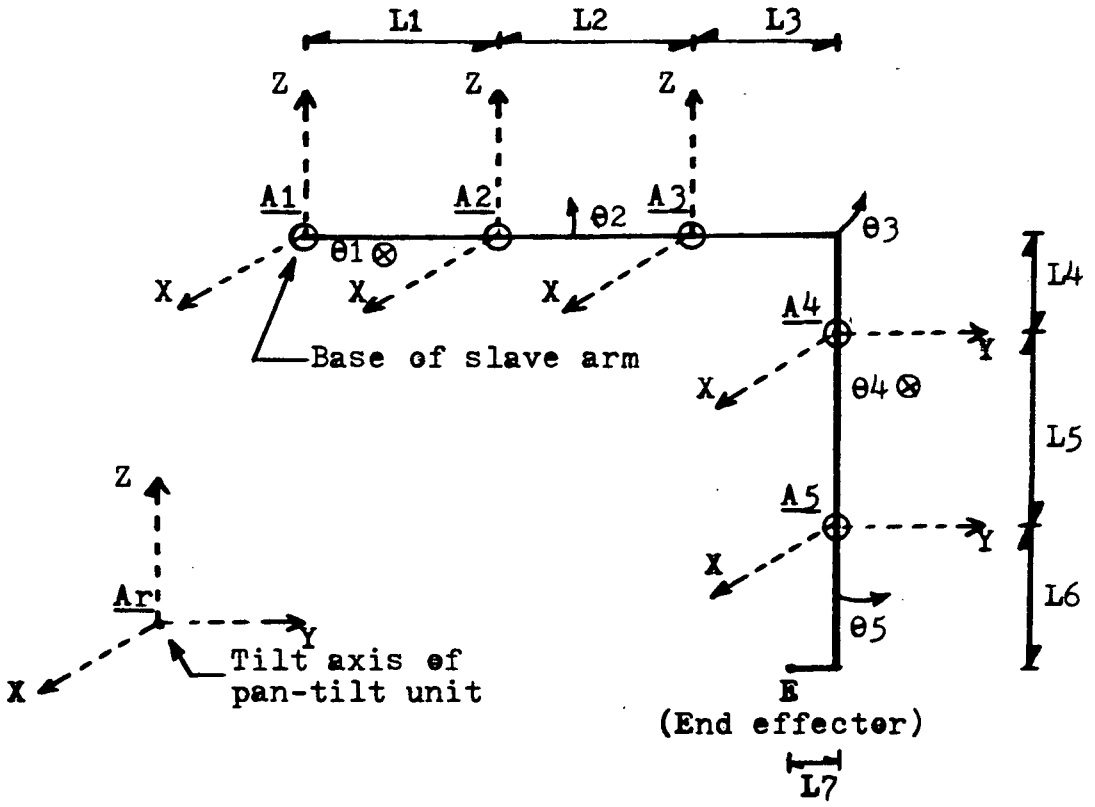
When the operator chooses to operate with the computer-based system, a two-way communication between the operator and the system is established. The operator can utter commands in order to cause changes in the video system. In

addition, the system informs the user of recognition results received from the word recognizer: the information is displayed on a terminal screen (Figure 2.9) and, if the result is a rejection, a buzzer is sounded. The terminal is also used to remind the user of the syntax (e.g. what should come next, violation of syntax, etc.), and to enter some data to initialize the system.

The computer-based system (when used) can implement some computerized functions on the video system which include the following: automatic tracking of the slave arm (more specifically of its end effector) by the camera, "memorization" of pan-tilt setups, discrete motions of the pan-tilt unit. To carry out these functions, current positions of the slave arm (i.e. angle of each joint) as well as of the pan-tilt unit are required by the computer. This information is not needed when operating in the manual mode, since such functions can only be implemented through the use of a computer.

The automatic tracking mode requires finding the position of the end effector of the slave arm with respect to the pan-tilt unit, given the angle at each joint of the slave arm. After the computation of the position, the camera is moved accordingly so that it points towards the end effector; then, the process is repeated. The joint angles are obtained by reading hall-effect sensors located at the joints of the arm. Technically speaking, one is faced here with the problem of

solving the direct kinematics of the "slave arm/pan-tilt unit" system.



θ1 : ARM SWING
θ2 : SHOULDER
θ3 : ELBOW
θ4 : WRIST YAW
θ5 : WRIST PITCH

Position of base of slave arm with respect to Ar coordinate system = (d,e,f)

Figure 2.11: Solving the direct kinematics problem

Figure 2.11 shows the "slave arm/pan-tilt unit" system. Coordinate systems are drawn at each joint of the arm (A1, being at the base of the slave arm; A2; A3; A4; A5) as well as at the tilt axis of the pan-tilt unit (Ar, the reference coordinate system). Positive direction of motion is indicated at each joint. To determine the position of the end effector (E) with respect to the reference coordinate system (Ar), a series of matrices must be defined which describe relative translations and rotations between two consecutive coordinate systems. These matrices are expressed in terms of homogeneous transformations [18]. Let

aHb: matrix describing the position of the coordinate system Ab with respect to Aa.

T(i,j,k): Translation by (i,j,k)

R(B,θa): Rotation about the B-axis by an angle θa.

cos(θa) --> Ca

sin(θa) --> Sa

,we have:

$$rH1 = \begin{bmatrix} 1 & 0 & 0 & d \\ 0 & 1 & 0 & e \\ 0 & 0 & 1 & f \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{--> } T(d,e,f)$$

$$1H2 = \begin{bmatrix} C1 & -S1 & 0 & -L1xS1 \\ S1 & C1 & 0 & L1xC1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{--> } R(Z,\theta 1) \text{ x } T(0,L1,0)$$

$$2H3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C2 & -S2 & L2xC2 \\ 0 & S2 & C2 & L2xS2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{--> } R(X,\theta 2) \text{ x } T(0,L2,0)$$

$$3H4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C3 & -S3 & (L3xC3)+(L4xS3) \\ 0 & S3 & C3 & (L3xS3)-(L4xC3) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{--> } R(X,\theta 3) \text{ x } T(0,L3,-L4)$$

$$4H5 = \begin{bmatrix} C4 & 0 & S4 & -L5xS4 \\ 0 & 1 & 0 & 0 \\ -S4 & 0 & C4 & -L5xC4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{--> } R(Y,\theta 4) \text{ x } T(0,0,-L5)$$

The position of the end effector (E) with respect to the A5 coordinate system is given by the following matrix:

$$5PE = \begin{bmatrix} 0 \\ -L7xC5 + L6xS5 \\ -L7xS5 - L6xC5 \\ 1 \end{bmatrix} \quad \text{--> } R(X,\theta 5) \text{ x } \quad 0,-L7,-L6,1$$

Finally, the position of the end effector with respect to the Ar reference coordinate system is obtained by multiplying the intermediate matrices:

$$rPE = \begin{bmatrix} Xr \\ Yr \\ Zr \\ 1 \end{bmatrix} = rH1 \text{ x } 1H2 \text{ x } 2H3 \text{ x } 3H4 \text{ x } 4H5 \text{ x } 5PE$$

Figure 2.12 shows the position of the camera pivot point in the reference coordinate system: as seen in the figure, the pivot point is not at the origin of the axes, but has offsets in the X and Y directions (of value "i" and "j" respectively). Figures 2.13 and 2.14 show the angles involved in positioning the camera so that it points towards rPE=(Xr,Yr,Zr). Referring to these 3 figures, PAN and TILT angles are calculated as follows:

$$\phi = \text{ARCTAN } \frac{Xr}{Yr}$$

$$\xi = \text{ARCSIN}\left(\frac{i}{(Xr^2 + Yr^2)^{1/2}}\right)$$

$$PAN = \phi - \xi = \text{ARCTAN}\left(\frac{Xr}{Yr}\right) - \text{ARCSIN}\left(\frac{i}{(Xr^2 + Yr^2)^{1/2}}\right)$$

$$\alpha = \text{ARCTAN}\left(\frac{i}{j}\right) \qquad \text{(constant)}$$

$$Cx = (i^2 + j^2)^{1/2} \, \text{SIN}(\alpha - PAN)$$

$$Cy = -(i^2 + j^2)^{1/2} \, \text{COS}(\alpha - PAN)$$

$$TILT = \text{ARCTAN}\left(\frac{Zr}{[(Xr - Cx)^2 + (Yr - Cy)^2]^{1/2}}\right)$$

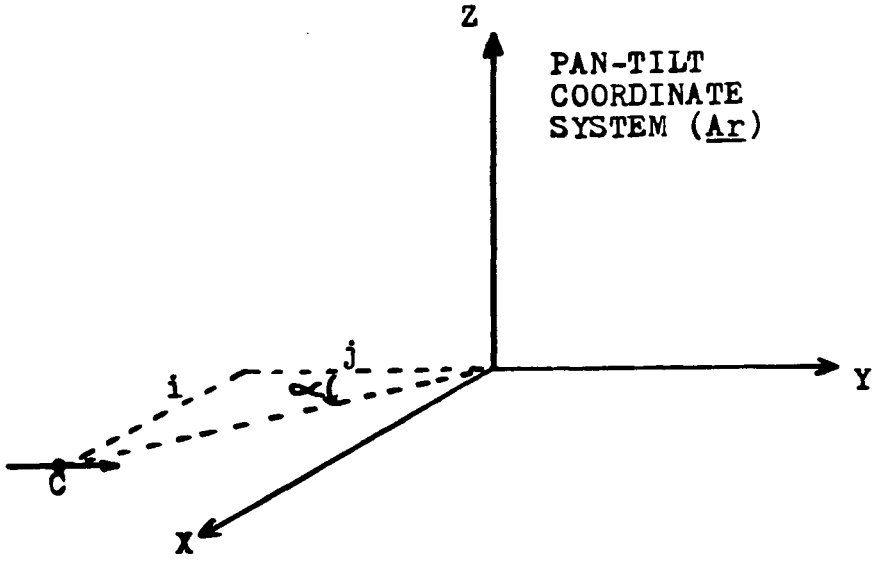


Figure 2.12: Position of camera pivot point in the reference coordinate system

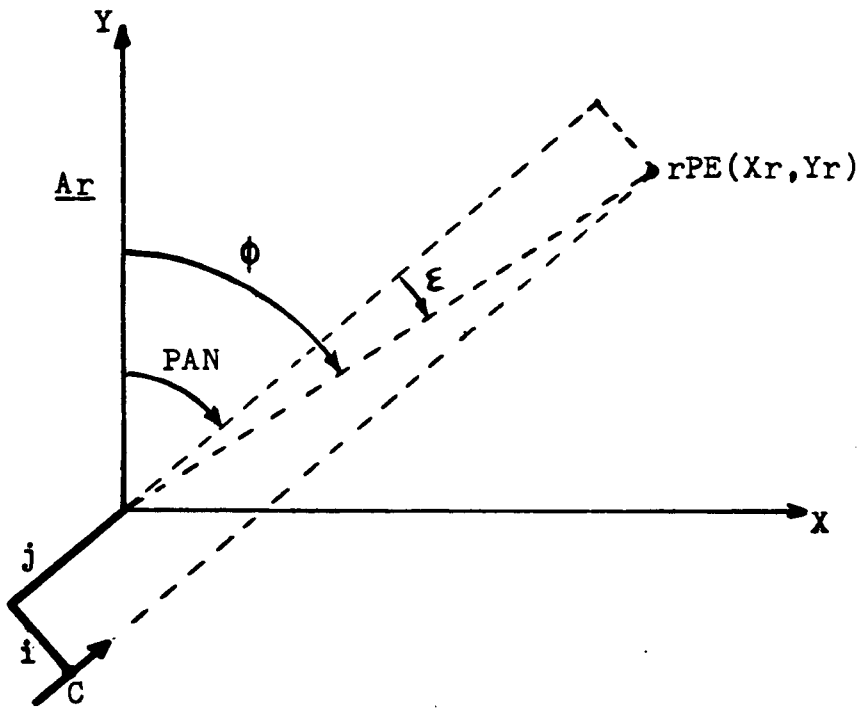


Figure 2.13: Calculating PAN angle

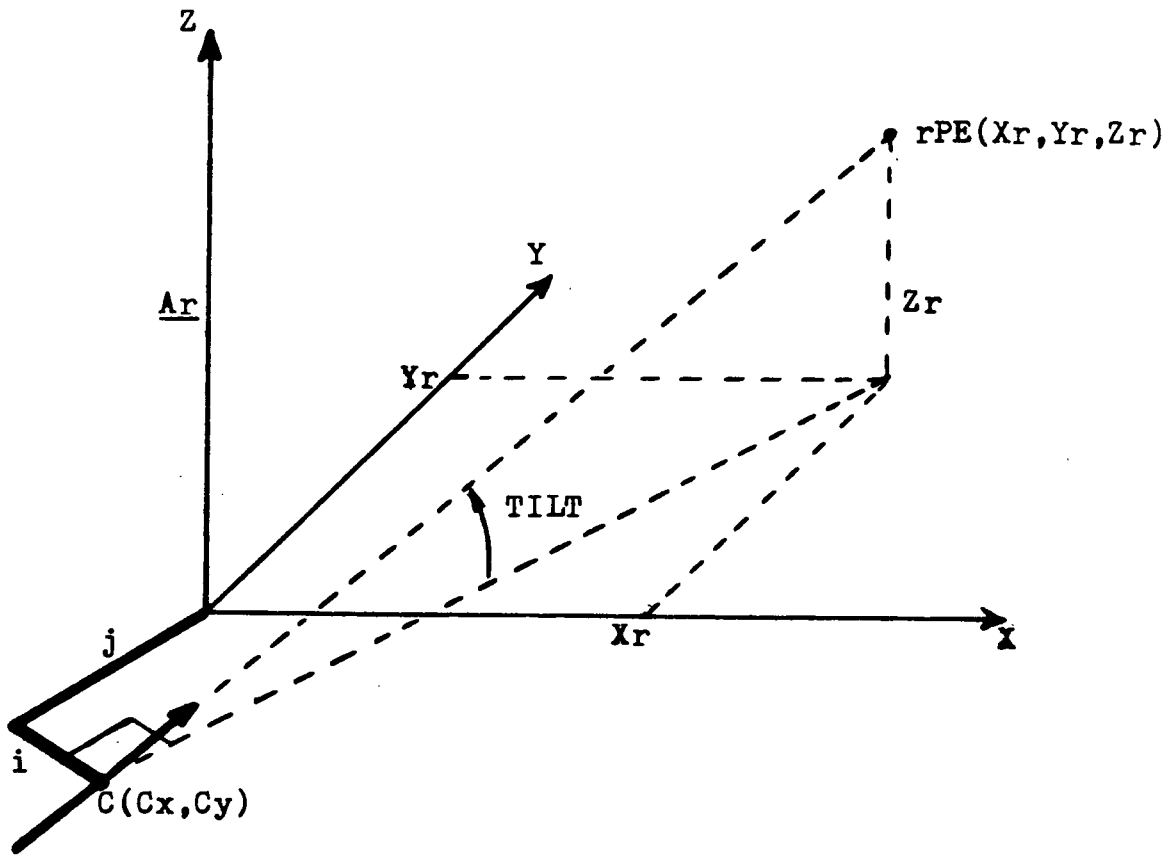


Figure 2.14: Calculating TILT angle

2.2 Tradeoffs Involved In The Design Of The System

The company Robotic Systems International (RSI) expressed the need for quantitative data on systems of the type investigated here. RSI provided all the equipment but the camera and the zoom lens, which belong to the Electrical Engineering department of the University of British Columbia. The choice of the computer system and language was decided in order to match the computer facilities of RSI. Thus, the computer system was designed around the 6809 microprocessor with an STD-bus structure: the C language, well suited for real-time applications, was also selected.

A careful study of speech recognizers available on the market resulted in the selection of the SR-100 NEC speech recognizer. This system offers features that meet the requirements of our actual application:

- It has an RS-232 port which makes it easy to interface to a CPU.
- Its accuracy in normal conditions is comparable to other systems. Moreover, its dynamic programming technique permits the system to obtain good results in a noisy environment.
- Its speed of computation (or, in other words, the minimum pause required between words) was comparable to other systems.
- The training session consists of only one utterance per

template, as opposed to 3 or 4 for certain systems.

- Its price is comparable to other systems, and yet offers, on the whole, more advantageous features.

The automatic tracking feature of the system demands much computation: many trigonometric and arithmetic calculations must be performed to solve the direct kinematics of the arm. Using software algorithms would take too much time and, therefore, only very slow motions of the arm could actually be tracked by the camera. In order to increase the effectiveness of the feature, an arithmetic processor unit was used to do the computation. A survey of the market was done and an STD-bus arithmetic card designed around the 9511 Math chip was selected.

Two options were available for the control of the pan-tilt unit. One consisted in using a position feedback PID controller designed by RSI. The other was to use the computer to perform the whole control: driving the motors of the pan-tilt unit as well as reading its position. The latter was preferred for the following reasons:

- The computer was not being used much as it was and, therefore, could easily handle this operation.
- More control over the pan-tilt unit could be more easily achieved. For instance, a variable speed for the motions could be implemented by simply varying the voltage applied to the motors.
- No need for extra hardware components.

2.3 Description Of Hardware

Referring to Figure 2.2, the teleoperator system hardware will be described according to its two seperate environments, namely the operator's workstation (i.e. the master arm environment) and the slave arm environment.

2.3.1 Operator's Workstation

The master arm (see Figure 2.15) is manufactured by the company Robotic Systems International (RSI). It provides spatially correspondent operator control over the 7 function slave arm manipulator. It has 5 hall-effect sensors which sense the angles of 5 joints (namely ARM SWING, SHOULDER, ELBOW, WRIST YAW, and WRIST PITCH); the wrist of the slave arm can be continuously rotated in either direction by toggling a switch on the master arm; finally, another switch controls the opening and closing of the jaw of the slave arm.

The video monitor is a Panasonic WV-5310 model and has a 5.5X7 inch screen. A Digital Equipment VT-101 keyboad-display terminal was used. Figure 2.16 is a schematic diagram of the manual control box. It consists merely of momentary push buttons: 4 for the control of the pan-tilt unit (UP, DOWN, RIGHT and LEFT) and 4 for the zoom lens (FOCUS-FAR, FOCUS-NEAR, ZOOM-OUT and ZOOM-IN). Since the camera used has a built-in automatic iris control, a manual control over the iris of the

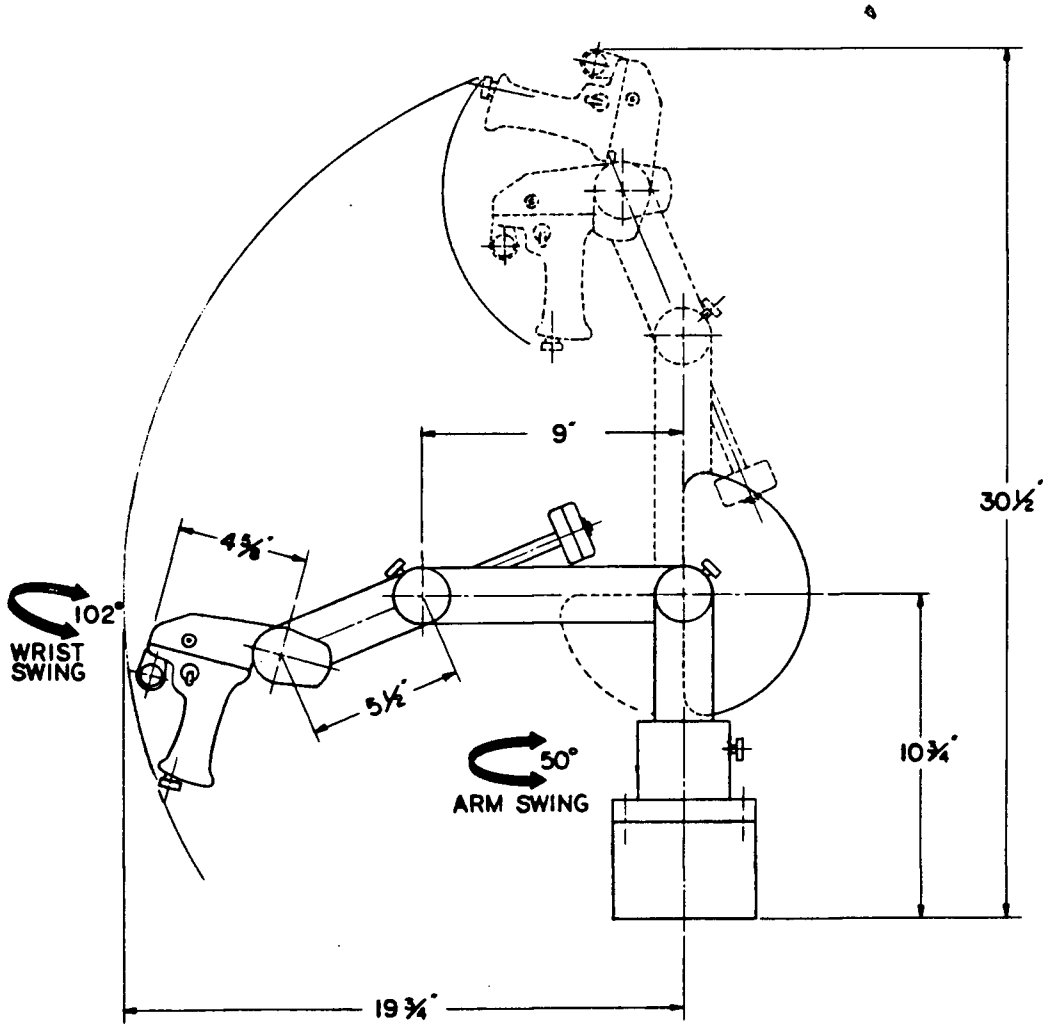


Figure 2.15: Master arm

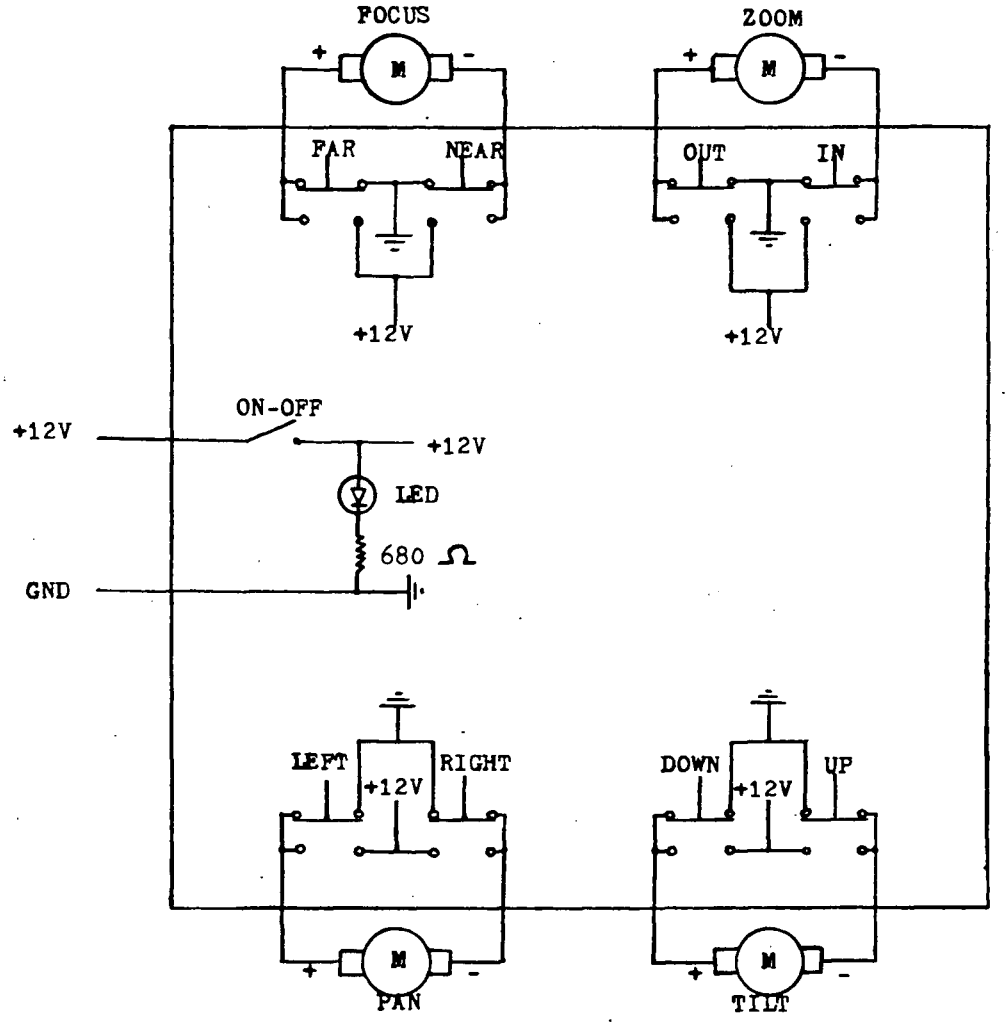


Figure 2.16: Manual control box schematic diagram

zoom lens was not necessary. By pressing a push button, 12 volts are applied to the corresponding motor: the polarity of the voltage depends on which push button of the set of two is pressed; if both of the same set are pressed at the same time, no voltage is apllied to the motor.

An NEC SR-100 speech recognizer with a Shure noise-cancelling microphone were used as a voice interface. The speech recognizer [17] is a speaker-dependent, isolated-word recognizer, which utilizes a filter-bank to analyze the speech signal in the frequency domain: dynamic programming is at the basis of the pattern matching process that follows. Among its characteristics, it provides a maximum vocabulary size of 120 words, which can be divided into a maximum of 100 clusters. The training session consists of only one utterance per word. It is interfaced to a computer through an RS-232 communication port. It also includes a buzzer which may be activated by a switch to sound when the system can not recognize a word (i.e. rejection).

The computer-based system is composed of 6 STD-bus cards and 2 interface modules, as shown in Figure 2.17. The heart of the system is a 6809 microprocessor card : it controls the overall operation of the system, managing its every resource and allowing a proper flow of information to travel among its components. A small on-board monitor, along with a RS232 serial port, establish a link between the CPU and the user's terminal.

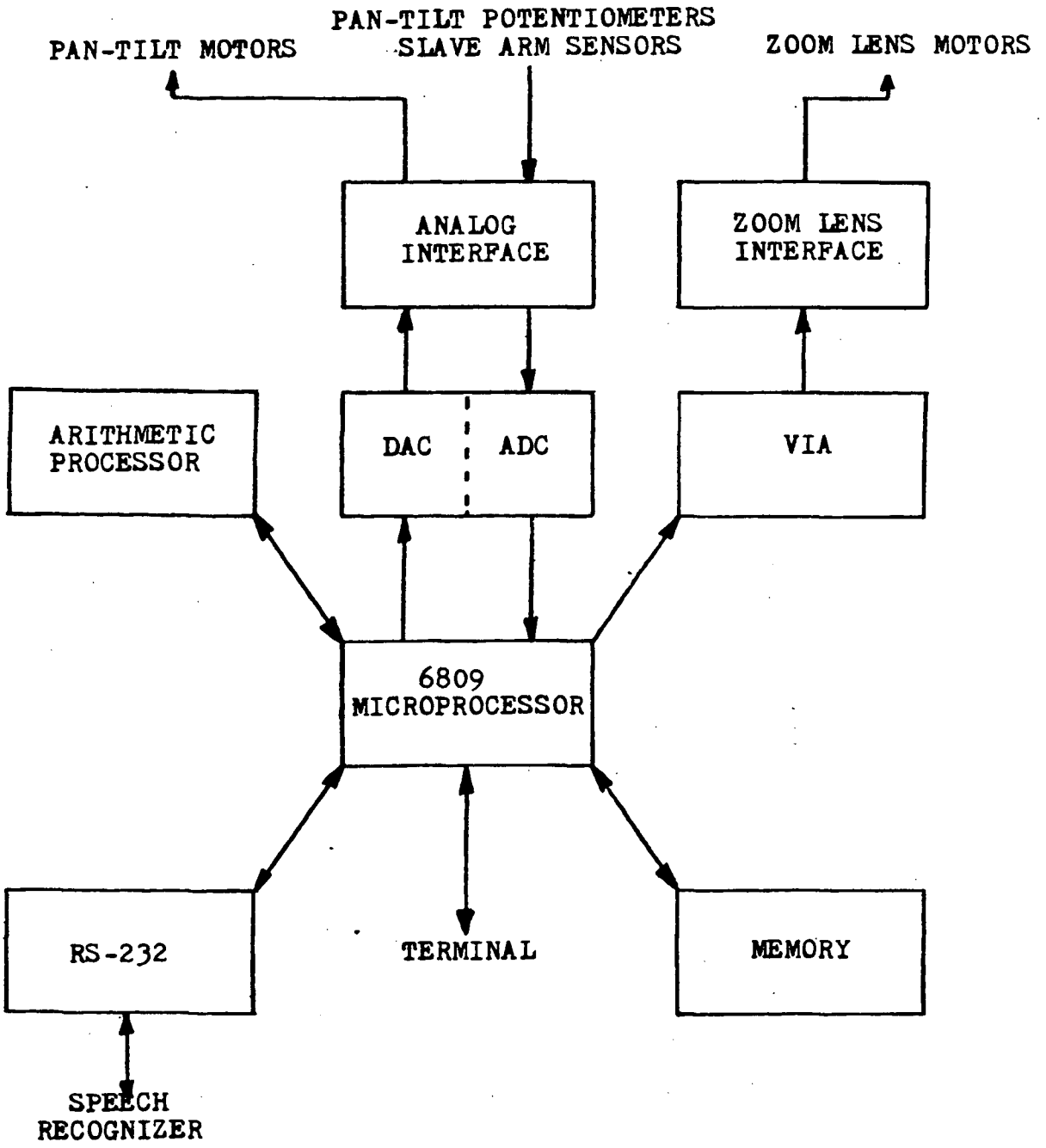


Figure 2.17: Computer-based system block diagram

In addition to supplying a communication channel to the operator, this link is also used for the start-up procedure: memory map selection, downloading of the program and execution.

A 32-Kbyte static RAM memory board is used to receive and store the 16-Kbyte long program. The program resides permanently in a TNIX-operated Tektronix computer, on which the program was entirely developed. A "LOAD" command available from the monitor enables Motorola "MIKBUG" formatted data to be loaded into the system's memory. Thus, a "MIKBUG" version of the program codes can be transferred from the Tektronix system into the system's memory. The use of RAM memory, as opposed to ROM memory, was necessary as the system was under development. The system is still undergoing some further studies; once it gets to its final shape, the program will be stored permanently in ROM memory, which will make the system a stand-alone unit.

The CPU communicates with the speech recognizer through a RS232 port card. In particular, this communication channel permits the CPU to initialize the speech recognizer, to issue commands to it as well as to receive recognition results from it.

The computation required to perform the automatic tracking of the arm by the camera is handled by an Arithmetic card. It is built around the 9511 Math chip, and provides high speed floating-point arithmetic and trigonometric computation

capability.

In conjunction with an analog interface module, an analog I/O board permits the CPU to have access to the pan-tilt unit and mechanical arm. Figure 2.18 is a schematic diagram of one of the two identical channels of the analog interface module. The 10-bit D/A converters are configured to produce a bipolar range of ±5 volts, for a 2's complement input (-512 to 511). Similarly, the 10-bit A/D converter accepts analog inputs between ±5 volts and translates them into 2's complement integer outputs (-512 to 511).

Two D/A converters are used to control the motors of the pan-tilt unit: channel 0 for the pan motor, and channel 1 for the tilt motor. Their outputs, which are not capable of directly driving DC motors, go through an electronic circuitry which is basically 2 similar power op-amps: the DC gain of the op-amps, adjustable through the potentiometers, is set to about 2. The outputs of the op-amps are connected to the motors of the pan-tilt unit. Thus, the input integer value of either D/A converters produces an analog voltage across the corresponding motor proportional to its value: for instance, -512 at the input of the D/A converter yields to a voltage of about 10 volts across the motor.

Two channels of the A/D converter are used by the CPU to determine the position of the pan-tilt unit: channel 5 for the

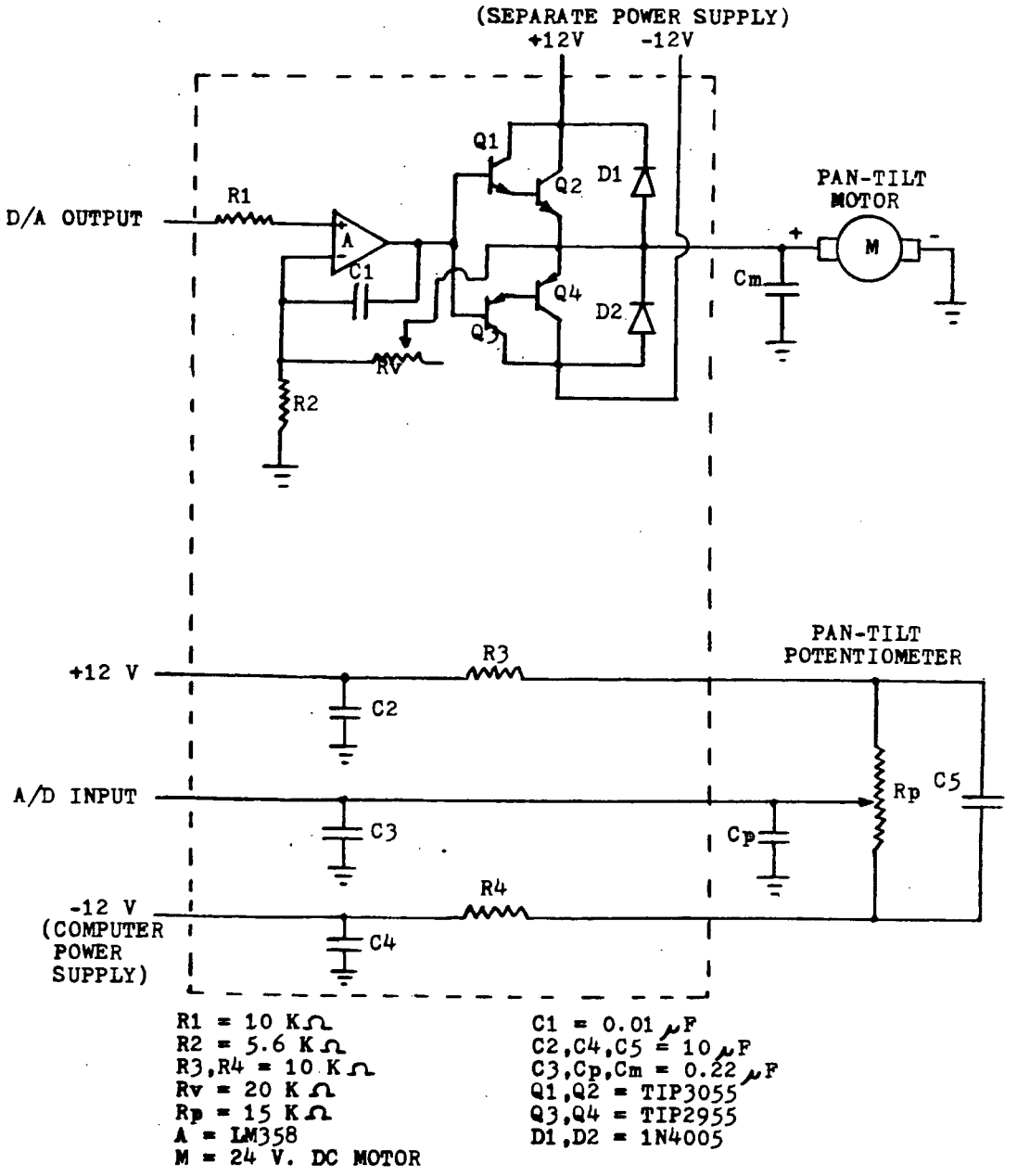


Figure 2.18: Analog interface module

pan, and channel 6 for the tilt. The pan-tilt unit has 2 15-Kohm potentiometers mounted on its shafts: ±12 volts (supplied by the computer power supply) are applied to the potentiometers, and resistors are connected in series with them to bring the voltage range down to ±5 volts. Reading those signals produces integer values (2's complement) proportional to the rotation of the shafts.

The CPU uses 5 channels of the A/D converter to read the signals from the hall-effect sensors of the remote arm. The signals are already conditioned and meet the voltage range requirement of the converters (i.e. ±5 volts). The CPU reads them in the process of computing the automatic tracking of the slave arm.

An interface for the zoom lens has been designed by Tony Leugner of the Electrical Engineering department of UBC. It actually allows both digital and manual (toggle switches) control of the zoom, focus and iris of 2 independent zoom lenses. It consists of 6 similar channels: 3 features per zoom lens, and 2 zoom lenses. The schematic diagram of one such channel is presented in Figure 2.19. Each channel is digitally controlled by 2 bits as follows:

- 0-0: no voltage applied to motor
- 0-1: +12 volts applied to motor
- 1-0: -12 volts applied to motor
- 1-0: no voltage applied to motor

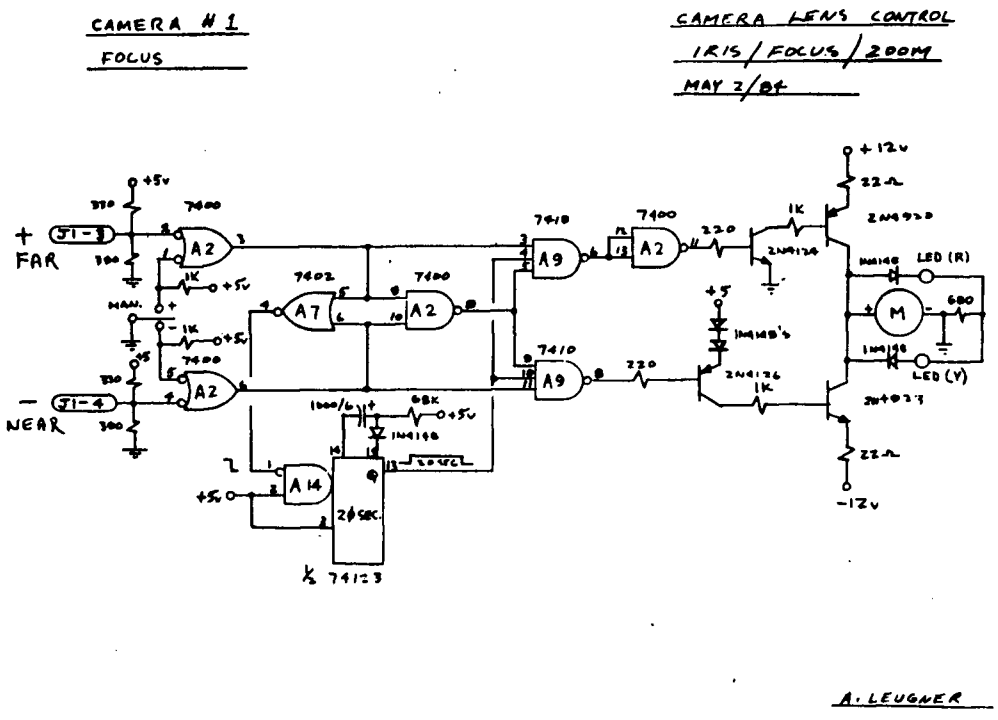


Figure 2.19: FOCUS control schematic diagram

In our system, only one zoom lens is used: a total of 6 bits is then required for the overall control of the zoom lens. A VIA card (Versatile Interface Adapter) supplies the CPU with a parallel I/O port for that very purpose. Bits 1-0 control the iris; 3-2, the focus; 5-4, the zoom. As already mentioned, toglle switches are also available for the user to manually activate the zoom lens motors.

2.3.2 Slave Arm Environment

The slave arm is a hydraulic manipulator which was originally developed by ISE (International Submarine Engineering) for subsea applications. Figure 2.20 shows its design. It consists of a 6 DOF arm plus an OPEN/CLOSE jaw function. It is coupled to a master arm (which is a scale model of the arm itself) through an analog control system. Five of the six joints of the slave arm (ARM SWING, SHOULDER, ELBOW, WRIST YAW and WRIST PITCH) keep a spatial correspondence with the corresponding joints of the master arm: hall-effect sensors on those joints of both arms make this coupling possible. Toggle switches are used for the other joint (continuous WRIST ROTATE) and the opening and closing of the jaw. The arm has a reach of 56 inches and can lift 150 lbs when fully extended.

Description

1. Gripper
2. Wrist rotate
3. Wrist pitch pivot
4. Wrist pitch sensor
5. Wrist yaw pivot
6. Wrist yaw sensor
7. Elbow pivot
8. Elbow sensor
9. Shoulder pivot
10. Shoulder sensor
11. Arm swing sensor
12. Arm swing pivot

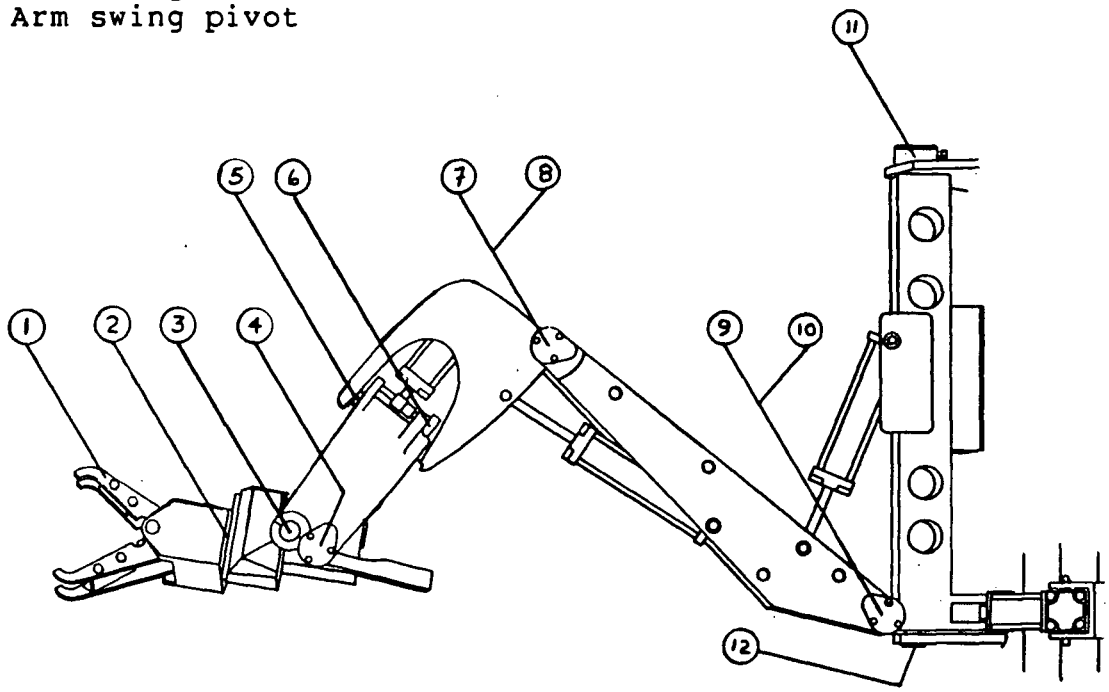


Figure 2.20: Slave arm

The pan-tilt unit, the design of which is shown in Figure 2.21, was designed and assembled by Al Rylandsholm from RSI. Two 24-volt DC motors drive the unit and two 15-Kohm potentiometers are mounted on its shafts for position feedback purposes.

The video system is composed of:

- An MTI DAGE camera : it has VIDICON sensors to capture the image, and features an automatic iris control.
- A FUJINON 17.5-105 mm TV zoom lens (C6X17.5B-MD3): 3 12-Volt DC motors control its focus, zoom and iris features.
- A Cosmicar 8 mm wide angle lens, which does not have any remote controls.

2.4 Description Of Software

2.4.1 Overall Description

The software was developed on a TNIX-operated Tektronix 8560 computer. The C language [22] was used because it is well suited to real-time applications and also for software compatibility with existing RSI software products. Only a small header (which sets the CPU's structure and then calls the main program) and the interrupt handling routine (which is basically a CALL to a C program) were actually written in 6809 assembler. An Introl C cross-compiler compiles the programs and generates

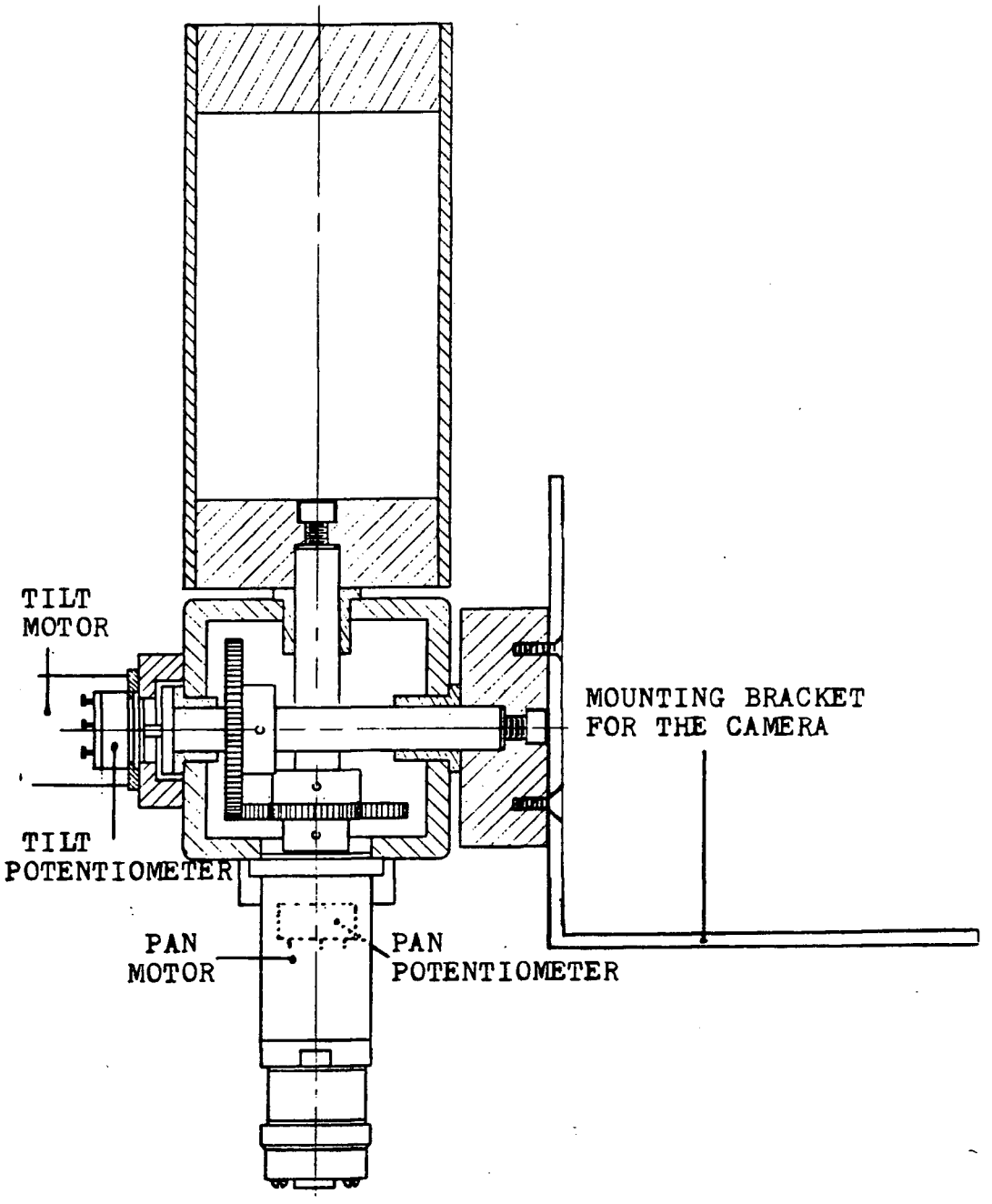


Figure 2.21: Pan-tilt unit

6809 codes: a loader, supplied with the cross-compiler, translates the codes into a Motorola "MIKBUG" format. These formatted codes are downloaded into the 6809 computer-based system through the "LOAD" command of the monitor. The program is then ready to be executed.

The overall software can be divided into 6 major components. The first one is found in the "LIB.C" file: it provides facilities to input/output data from/to the user's terminal keyboard/screen such as string of characters, integers and floating-point numbers.

"HEADER.M09" and "MAIN.C" constitute another portion of the software. "HEADER.M09" enables the interrupt of the CPU and determines the location of its stack. Part of "MAIN.C" is also used for initialization purposes: selection of the I/O page, assignment of the interrupt vector, initialization of variables and configuration of the system's hardware. The other part serves as a monitor, offering a choice of "actions" to the user which includes the initialization of the speech recognizer and passing to the section of the software that allows control of the video system.

"RS232.C", "SR100.C" and RSINT.M09" look after the communication between the speech recognizer and the CPU (through the RS232 card). "SR100.C" contains all the commands that the recognizer accepts: training command, recognition command,

cluster_definition command, etc. "RS232.C" and "RSINT.M09" provide the facilities to:

- send commands to the recognizer,
- receive messages from the recognizer,
- manage the reserved section of the memory where messages are stored into and retrieved from.

"CAMERA.C" handles the interpretation of uttered commands to control the video system. Using a pre-defined syntax, it analyzes recognition results of the speech recognizer and directs actions accordingly.

"PANTILT.C", "POS.C", "DAC.C" and "ZOOM.C" oversee the operation of the pan-tilt unit and the zoom lens: driving their motors, and reading the position of the pan-tilt unit and of the arm sensors.

Finally, "LIBMATH.C" and "MATH_COMMAND" supply the necessary arithmetic and trigonometric computation capability for the automatic tracking feature. This feature is computed in the "ARM.C" file.

2.4.2 Communication Between Speech Recognizer And CPU

A half-duplex RS232 channel is set up between the CPU and the speech recognizer as a means of communication: the common language consists of alphanumeric characters and others (e.g. "*", ",", "<CR>"), coded in ASCII. A message, originating from either end, is always terminated by a <CR> character (i.e. Carriage Return).

Figure 2.22 shows the different steps involved in the initialization process of the speech recognizer. First, the speech recognizer is told by the CPU to reserve a section of its memory which will eventually contain the templates of say "M" words: this group of reference templates constitutes a cluster and is given a specific label (which is a number). Thus, step #1 defines a cluster #"L" which is "M"-word long.

The next step consists in training the speech recognizer, that is producing the reference templates for the cluster previously defined. To this end, the CPU sends a command to the speech recognizer specifying that the user desires to produce the template of the word #"X" of cluster #"L". Upon reception of the command, the speech recognizer waits until it "hears" an utterance over its microphone; it then analyzes the speech signal (feature extraction) and stores within its memory a template which characterizes how the word #"X" of cluster #"L" was just uttered. This process is repeated for each word of the

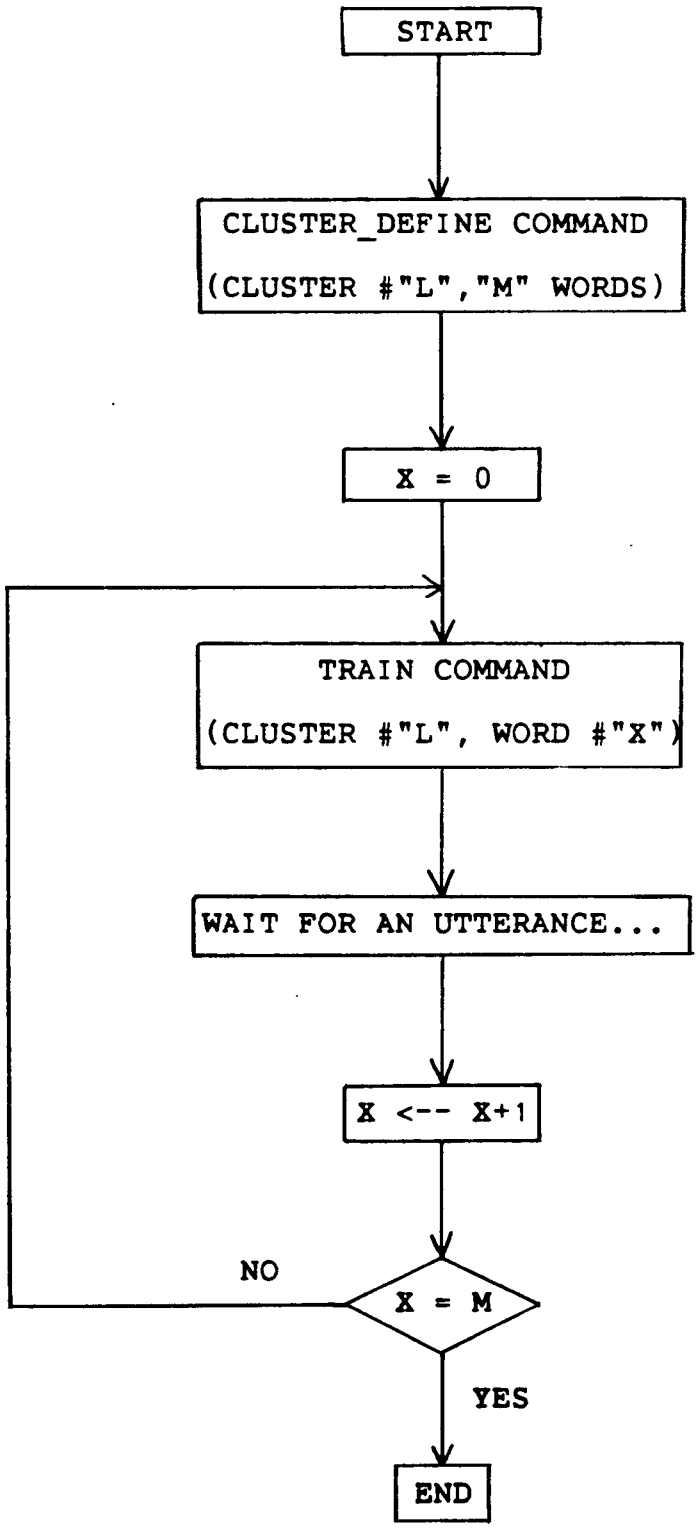


Figure 2.22: Initialization process of the speech recognizer

cluster, by simply varying "X" between 0 and "M-1" and sending the commands for training. Note that the system can be re-trained for any word of any cluster at any time: the proper command for training for that word is sent and the speech recognizer (after receiving the speech signal) replaces the old template with the new one.

After the initialization process, the speech recognizer is now ready to operate, that is to recognize the user's voice. To get the recognizer's attention, the CPU sends a command to have the recognizer "listen". After detecting an utterance, the recognizer analyzes the speech signal and produces a template the same way as in the training session: this template is then compared with the reference templates of the cluster specified in the command. The result of the matching process is then sent over the communication channel to the CPU. This procedure, presented in Figure 2.23, must be followed every time the user wants the speech recognizer to recognize a word: it is not a continuous process.

Two possibilities can result from the recognition process. Either none of the reference templates matches close enough the template of the word that was just uttered, or there is at least one such reference template. In the former case, the letter "J" is sent to signal a REJECTION , i.e. the recognizer could not figure out what was just said. In the latter case, the result consists of the letter "R" followed by the number of the

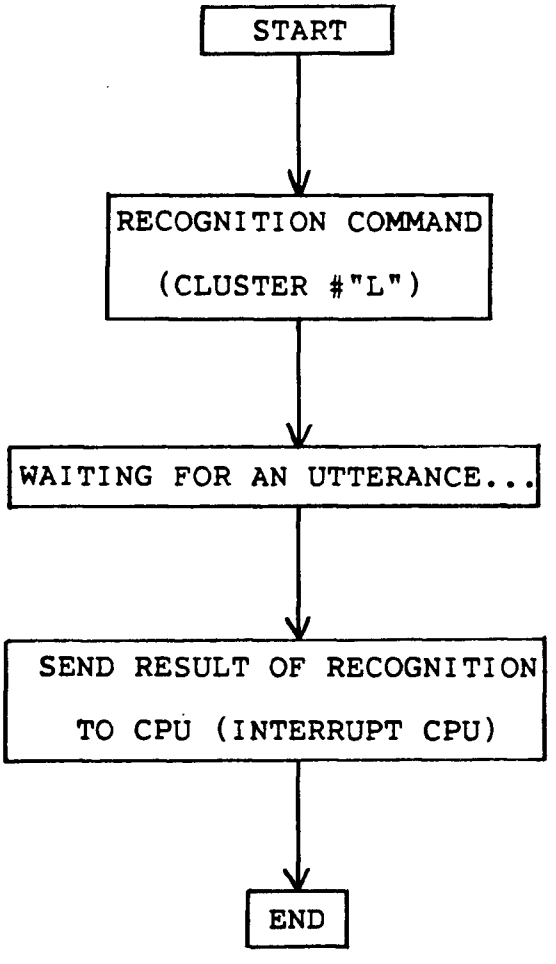


Figure 2.23: Recognition process

reference template that matches the closest. If the chosen reference template corresponds to what was actually said, it is a RECOGNITION ; if not, it is a MISRECOGNITION, that is the recognizer misunderstood.

No hand-shaking between the speech recognizer and the CPU exists: there is no direct way of making sure that a message sent from either device will be properly received by the other; also, messages can not cross each other since the recognizer handles only a half-duplex communication. To allow proper communication between the two, the CPU must do two things. First, it must always keep track of the state of the recognizer and of its protocol of communication. For instance, no commands can be received by the recognizer while it is waiting for an utterance. Also, it always provides an answer to a command after it has been executed: the CPU must wait for that reply to come in before carrying on with another command.

The other requirement of the CPU is that it must always be alert and ready to read any messages that would be coming in: the recognizer does not signal the CPU before sending any data. To this end, the interrupt capability of the CPU (and of the RS232 card) is used. As soon as the first byte of a message is received by the RS232 card, the CPU is interrupted so that the data coming in can actually be read in. The interrupt routine is shown in Figure 2.24. First, the CPU reads the byte that already came in and stores it in a reserved section of the

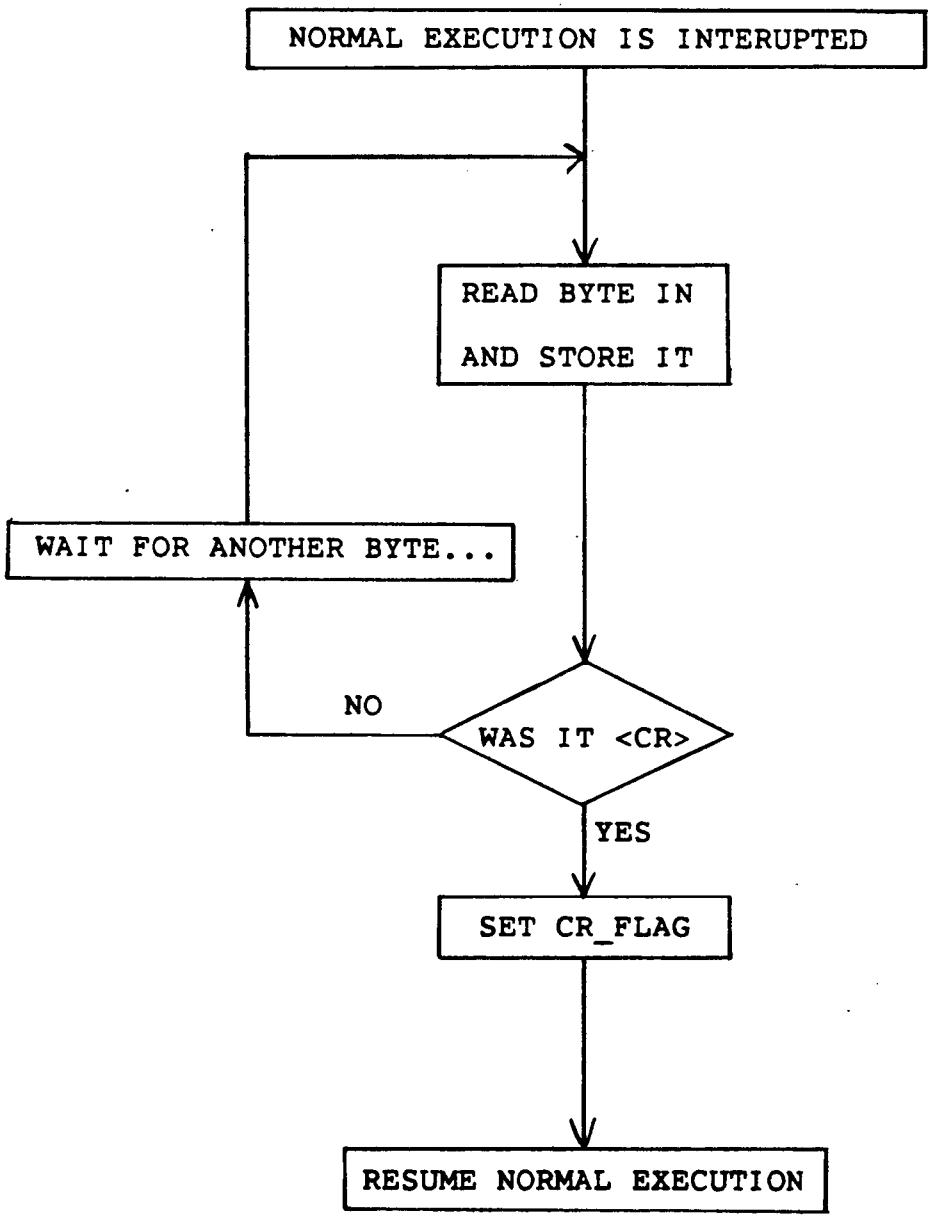


Figure 2.24: Interrupt routine

memory (called RS_BUFFER). It then waits until another byte comes in: when it eventually arrives, it is read in and stored. This process continues until the last byte of data, that is a <CR>, is read in and stored. The CPU then sets a flag (called CR-flag) and resumes its operation.

Since the process of receiving data from the recognizer is performed by interrupt, the CPU is not "aware" of this. The only way the CPU can find out if a message has actually been received from the recognizer is by looking at CR-flag. If it is set, the CPU goes to the specific place in its memory, retrieves the message, analyzes it and responds accordingly: the flag is also reset. This is an operation performed constantly during the execution of the program.

2.4.3 Vocabulary And Syntax

The speech recognizer operates on a vocabulary that has a specific number of words. The words are chosen to suit the needs of the application. In our case, 33 words were selected in the control of the video system: they are listed in Table 2.1 along with their respective word number.

Before one can use the system, one must initialize the speech recognizer to one's voice, by going through the two steps mentioned earlier. During the training session, it is important to stick to the word numbers, since the recognizer deals with

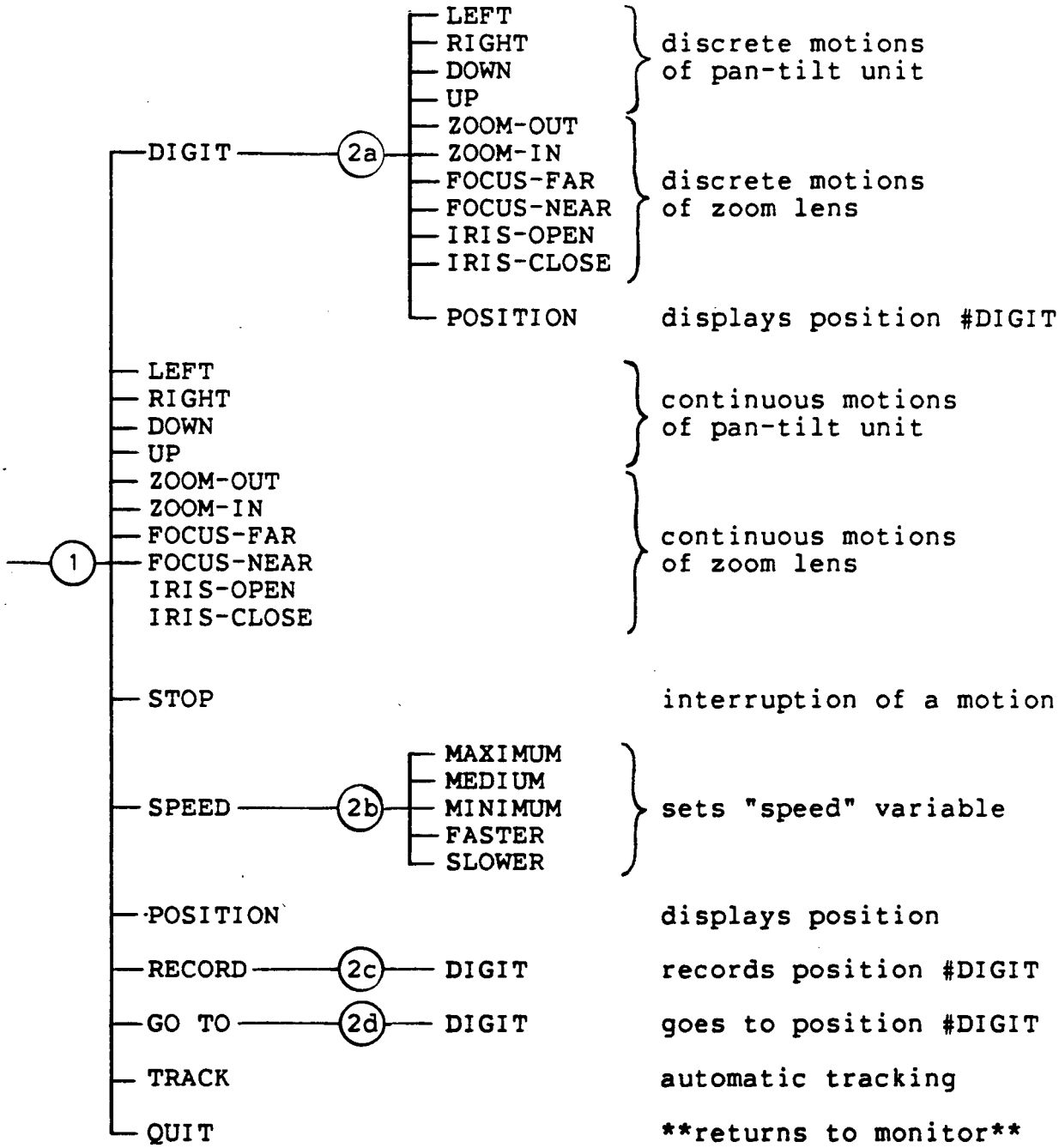
WORD NUMBER	WORD
0	ZERO
1	ONE
2	TWO
3	THREE
4	FOUR
5	FIVE
6	SIX
7	SEVEN
8	EIGHT
9	NINE
10	TEN
11	LEFT
12	RIGHT
13	DOWN
14	UP
15	ZOOM-OUT
16	ZOOM-IN
17	FOCUS-FAR
18	FOCUS-NEAR
19	IRIS-OPEN
20	IRIS-CLOSE
21	STOP
22	SPEED
23	MAXIMUM
24	MEDIUM
25	MINIMUM
26	FASTER
27	SLOWER
28	POSITION
29	RECORD
30	GO TO
31	TRACK
32	QUIT

Table 2.1: System's vocabulary

word numbers and not with the words as such: therefore, when the TRAIN command is sent for word #0, the word ZERO must be uttered and no other.

After the training, the operator can then carry on with the control of the video system by uttering commands. Every recognition process produces a result which is received by the CPU: it can be either a rejection or a recognition and, in the latter case, it gives the word number of the word believed to have been uttered by the operator. In receiving those results, the CPU must have rules to go about to know how to interpret them in order to proceed with proper actions. The set of rules that the speech recognizer follows constitutes the syntax of the language. For instance, the simplest syntax that may be used is a one-level syntax whereby each word uttered leads immediately to an action. —

Figure 2.25 shows the syntax used in this video system control. Certain actions take on immediately after one utterance is received: for example, if the CPU receives word #28 (i.e. POSITION) as a recognition result, the current position of the pan-tilt unit is read and is displayed on the user's terminal. Other actions require two consecutive utterances: for instance, to have the pan-tilt unit go to a pre-defined position #"X", the CPU must receive word #30 (i.e. GO TO) followed by word #"X" (X is any digit) as recognition results. Thus, it is a two-level syntax: normally, after passing the first level, the



N.B.: DIGIT=0,1,....,10

Rules:

- Return to level 1 if syntax is violated and no action is taken.
- Any rejection result is ignored and the level (or sub-level) remains the same.

Figure 2.25: System's syntax

CPU proceeds with an action or goes into one of the sub-levels of level 2 where it will wait for another utterance before any action is taken.

A syntax must also include rules on how to deal with rejection results and also when the syntax itself is violated. The syntax is violated when a recognition result gives the number of a word that was not expected in the current level or sub-level of the syntax. For instance, if the CPU is currently in level 1 and receives an acknowledgement that FASTER was uttered (or more precisely word #26 is received); or if it receives GO TO (word #30) followed by SPEED (word #22). The syntax specifies that, upon violations of the syntax, the CPU returns to the first level of the syntax, and no actions are taken; however, if a rejection result is received by the CPU, it is ignored and the CPU remains in the same level of the syntax.

The operator is kept informed by the CPU (through the terminal) of recognition results (or rejection which, in this case, is accompanied with the sound of a buzzer) and of the status of the syntax (e.g. error in syntax, what should come next, etc.).

2.4.4 Zoom Lens Motions

Six commands are used to modify the setting of the zoom lens:

- ZOOM-OUT: increases the focal length
- ZOOM-IN: decreases the focal length
- FOCUS-OUT: focuses on a point that is closer
- FOCUS-NEAR: focuses on a point that is further
- IRIS-OPEN: opens wider the iris
- IRIS-CLOSE: diminishes the opening of the iris

Three similar functions are related to these 3 pairs of commands: zoom(time), focus(time) and iris(time). Figure 2.26 shows their flow chart. The polarity of the "time" parameter passed to the function determines which 2-bit pattern is sent to the zoom lens interface module (0-1 or 1-0) which, in turn, determines the polarity of the voltage that is applied across the motor: 0-1 produces -12 volts; 1-0, +12 volts. For example, the ZOOM-IN command yields to a call to the zoom(time) function with "time" sets to a positive value; for the ZOOM-OUT command, "time" is set to a negative value.

The magnitude of the "time" value determines the maximum amount of time a motor is turned on during the execution of its respective function. In the case where the function is not ended prematurely by an utterance (see below), the function called by the main program energizes the motor (according to the

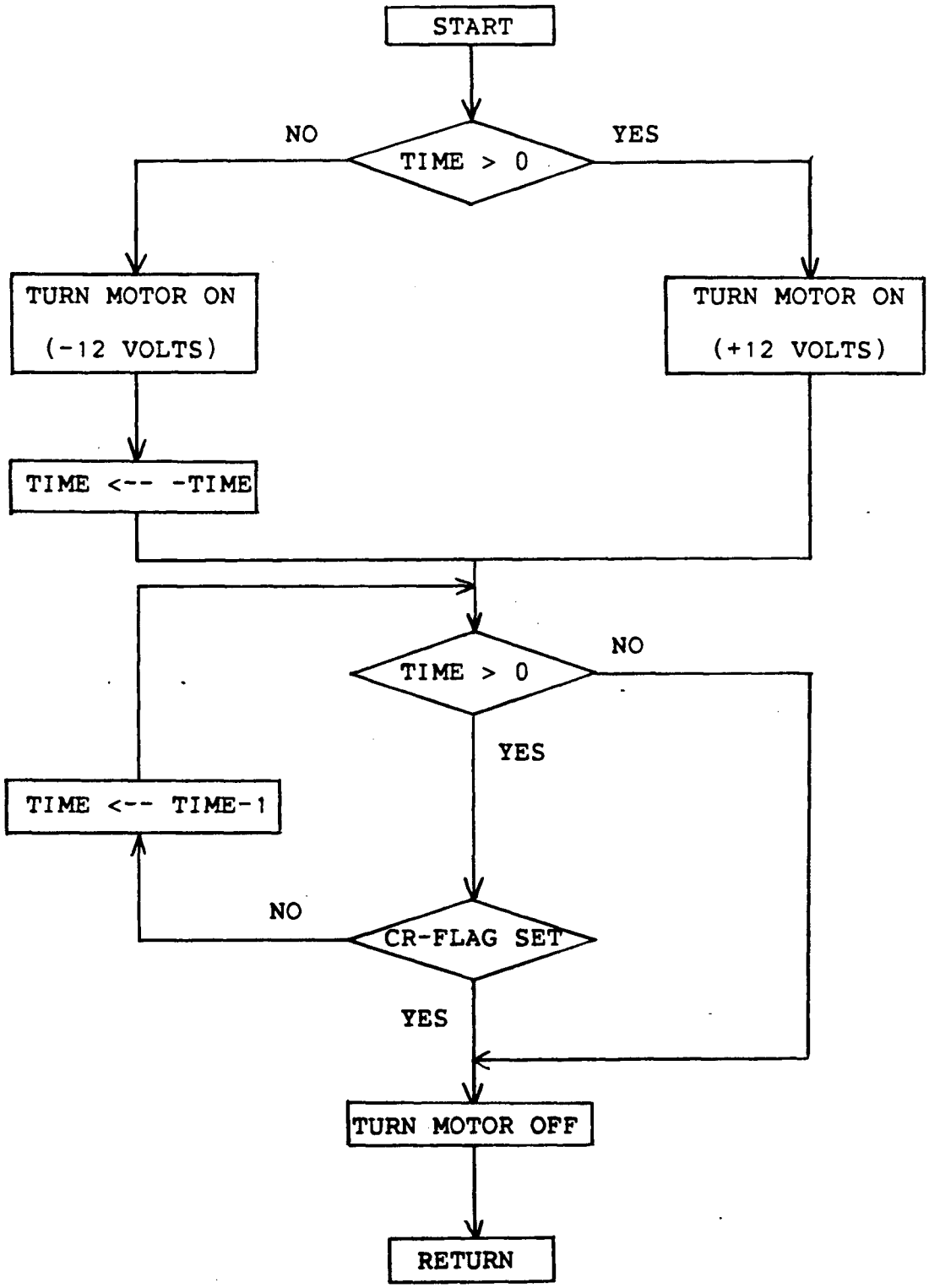


Figure 2.26: ZOOM(TIME), FOCUS(TIME), IRIS(TIME) functions

polarity of the "time") for a period of time determined by the magnitude of "time"; then, it de-energizes the motor (sends 0-0 bit pattern to the interface module) and returns to the main program.

Continuous and discrete motions are available. A discrete motion is obtained by uttering a digit (in level 1 of the syntax) followed by one of the six commands: the magnitude of "time" takes on a value that is proportional to the digit that was uttered. The multiplicative constant (called "ZFI_UNIT") is such that the digit 10 corresponds to approximately a third of the motor span. On the other hand, uttering one of the commands while in level 1 of the syntax leads to a continuous motion: the "time" parameter is set to its maximum value (in magnitude) so that one of the extremities of the motor span may be reached before returning to the main program.

Before calling a function, the CPU re-enables the speech recognizer; during the execution of the function, the CR-flag is constantly checked to see whether or not the operator has uttered anything. If the flag is set, the normal process is interrupted: the motor is turned off and the CPU returns to the main program (level 1 of the syntax) to figure out what was said. Even a rejection result from the recognizer causes the function to terminate and to return to the main program. Both continuous and discrete motions can be stopped before their time by any utterances (properly recognized or not). The STOP

command would normally be used by the operator.

Continuous motions are useful for gross motions of the zoom lens: the operator turns on the motor by uttering the corresponding command, and then turns it off when it reaches approximately the desired region. Alternatively, fine motions are performed by discrete motions: the operator figures out the amount of time the motor should be energized for, and proceeds with the two words (digit and command). An accurate adjustment of the focus between two points far apart from each other would require first a gross change of the focus followed by fine discrete motions.

2.4.5 Pan-tilt Unit Motions

Using the commands RIGHT, LEFT, UP and DOWN, the control of the pan-tilt unit is similar to that of the zoom lens. It allows for continuous and discrete motions, which can be interrupted before their time according to the same procedure as in the zoom lens case.

Two similar functions are associated with the four commands: pan(adc_unit) and tilt(adc_unit). Figure 2.27 shows their flow chart. The "adc_unit" parameter refers to the maximum amount of change that must be executed (either on the pan, or on the tilt); its polarity determines the direction of the motion. The function reads the current position of the pan

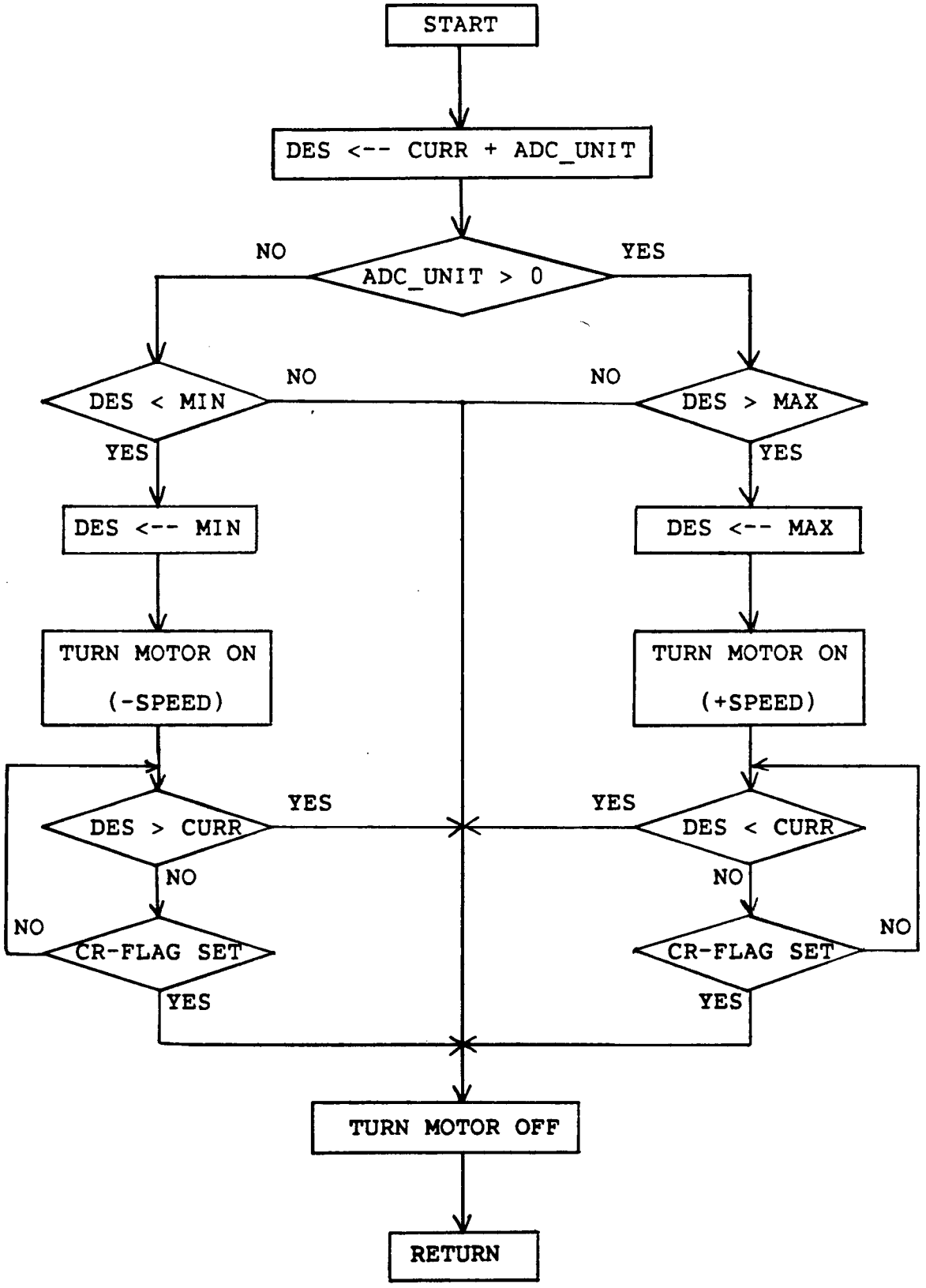


Figure 2.27: PAN(ADC_UNIT), TILT(ADC_UNIT) functions

(or of the tilt): "adc_unit" is then added to that current position and the result becomes the desired position of the pan (or of the tilt). This desired position must not be outside a permissible range of ±90 degrees: if it is, the desired position is set to the maximum position allowed. The motor is then turned on, and the CPU starts reading continually the current position as well as checking the CR-flag: as soon as the desired position is reached or the CR-flag is found set, the motor is turned off and the CPU returns to the main program.

The speed of the motion is determined by the input integer value of the D/A converter: the resulting voltage applied across the motor is proportional to that value. A global variable called "speed" is kept by the CPU for that purpose: the value sent to the converter to energize the motor is of the same magnitude as "speed", and its polarity is the same as of the "adc_unit" parameter. For instance, to produce a motion to the LEFT, "adc_unit" is set to a negative value, which results in sending -"speed" to the D/A converter channel 0.

Uttering one of the commands while the CPU is in level 1 of the syntax calls for a continuous motion: the "adc_unit" parameter is set to reflect a 180-degrees motion. This causes the desired position to automatically become one of the maximum positions. As in the case of the zoom lens, a discrete motion is obtained by uttering a digit (while in level 1) followed by a command: "adc_unit" takes on a value which is the multiplication

of the digit by a constant (called "PT_UNIT"). This multiplicative constant is such that one "adc_unit" corresponds to approximately 1.5 degrees of rotation.

The operator can change the value of the "speed" variable by uttering the word SPEED while the CPU is in level 1 of the syntax. Then, MAXIMUM, MEDIUM or MINIMUM causes "speed" to take on preset values (namely "SPEED_MAX", "SPEED_MED" and "SPEED_MIN" respectively). The current value of "speed" can also be increased by a certain amount (specified by the constant "SPEED_INC") by uttering FASTER (after SPEED); SLOWER causes the opposite effect.

2.4.6 Memorization Feature

At any time, the operator can ask for the current position of the pan-tilt unit by uttering the command POSITION. Upon reception of a recognition result giving word #28 (POSITION) while in level 1 of the syntax, the CPU reads the two potentiometers of the pan-tilt unit through the A/D converter and displays the results (i.e. two integers) on the screen.

A total of 11 positions of the pan-tilt unit can be memorized by the system at all time. A current position is memorized as position #x (x being a digit between 0 and 10) by uttering RECORD (while in level 1) followed by X. Upon receiving the two words, the CPU reads the current position of

the pan-tilt unit and stores it in its memory. A position can be re-defined at any time. It is also possible for the user to have the CPU display on the screen the values currently memorized under a certain position. To this end, a digit is first uttered (while in level 1) followed by POSITION: the CPU, then, reads from its memory the values of the position and displays them on the screen.

The purpose of memorizing positions is to allow the operator to have the pan-tilt unit go to a memorized position automatically. This is particularly useful for situations where the camera must often point towards certain defined directions. After a position has been memorized by the system as position #x, uttering GO TO (in level 1) followed by X causes the pan-tilt unit to move back into that position.

The function pan_tilt1(pan_pos,tilt_pos,speed), the flow chart of which is shown in Figure 2.28, oversees the motion of the unit. The "speed" parameter determines the speed at which the motion is executed; more precisely, it is directly proportional to the voltage that is applied across the motors. The "speed" global variable kept by the CPU (see section 2.4.5) is the value passed to the function as the "speed" parameter (for the GO TO command). The "pan_pos" and "tilt_pos" parameters constitute the desired position of the pan-tilt unit. In the case of a GO TO command, the pair of values memorized under position #x become those two parameters.

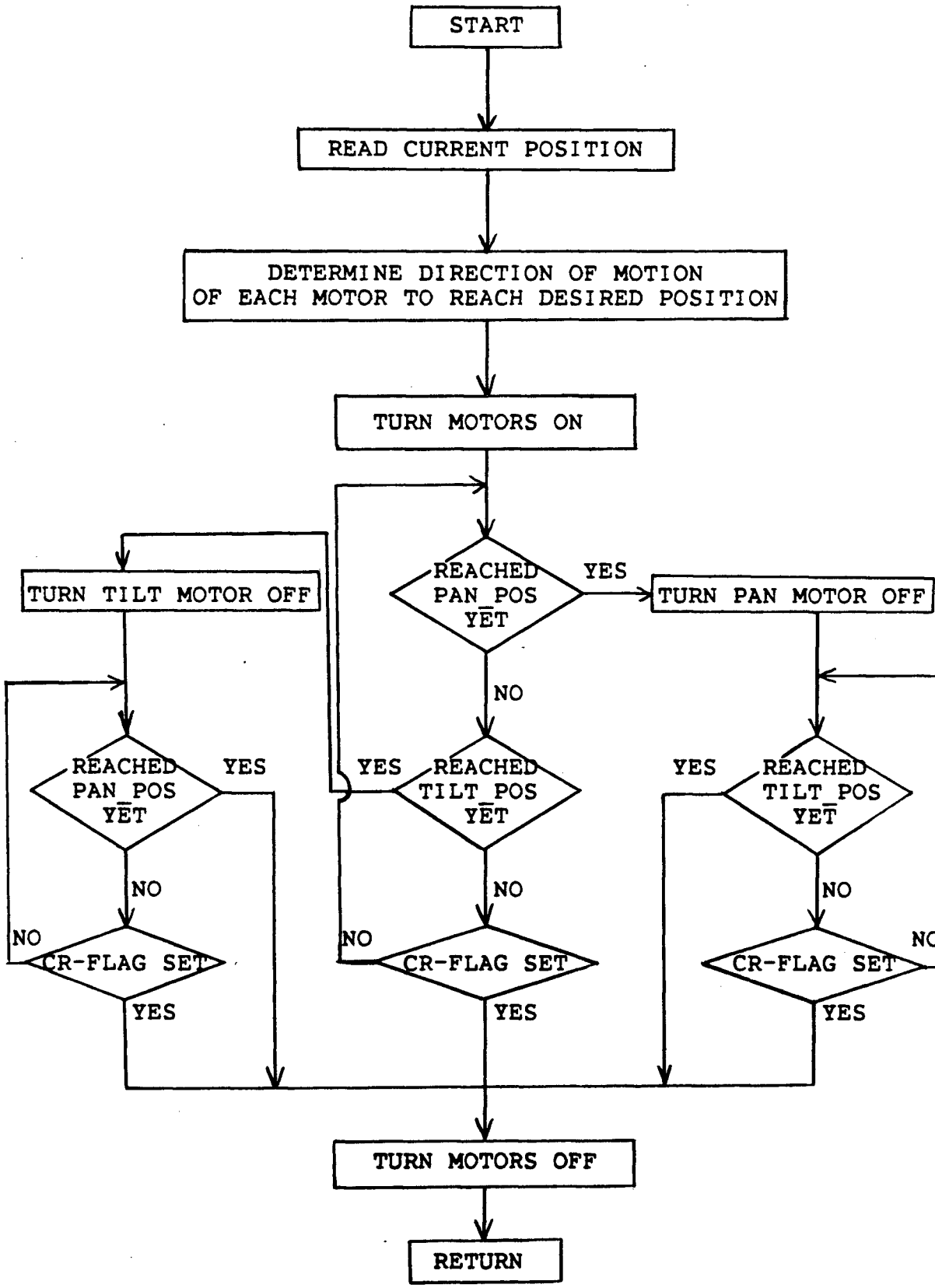


Figure 2.28: PAN_TILT1(PAN_POS,TILT_POS,SPEED) function

The first step involves finding out which direction each motor should turn in order to reach the desired position from the current position. Both motors are then energized accordingly: either the "speed" parameter (which is the same as the "speed" global variable in this case) or its negation is sent independently to each D/A converter. In the event that the CR-flag never gets set, the CPU lets a motor energize until the corresponding desired position is reached, and then turns it off; eventually,both motors will reach their respective desired position and will be de-activated, after which the CPU returns to the main program. The recognizer having been enabled before entering the function, the CR-flag is set if the recognizer hears something; if such is the case, the motion is interrupted (the motors are turned off) and the control is returned to the main program.

2.4.7 Automatic Tracking

The TRACK command, when uttered while the CPU is in level 1 of the syntax, sets the system into its automatic tracking mode whereby the camera automatically tracks the slave arm. Upon receiving the command, the CPU re-enables the speech recognizer, and passes the control to the track(speed) function. Figure 2.29 shows the flow chart of the function.

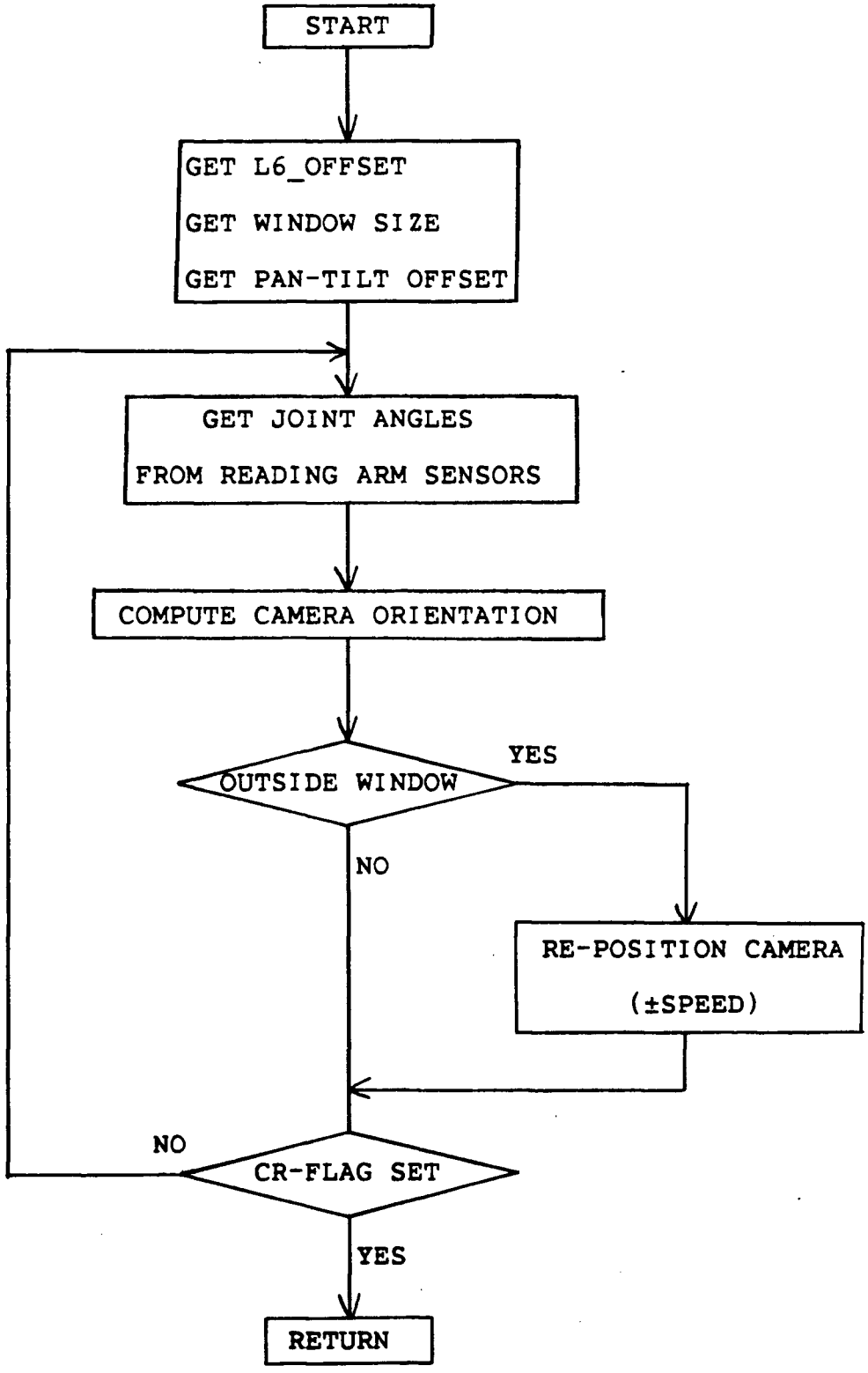


Figure 2.29: TRACK(SPEED) function

Three features are included in the implementation of the automatic tracking mode. A variable focal point is possible through the "L6_offset" variable: it permits tracking of a point distant by L6_offset from the end effector of the slave arm. This is particularly useful when the operator desires to track a tool held by the gripper of the slave arm, and not the gripper itself.

A round window of size specified by the "window" variable is defined around the focal point. After computing what the orientation of the camera should be in order to have the camera point to the focal point, this orientation is compared to the current position of the camera: if it lies within the limits of the window, the camera remains in its current position; if it does not, the camera is moved into the computed position and the window is re-defined around that new position. For instance, setting the window to zero causes the camera to compensate for any motions of the slave arm (as well as for the noise).

The positioning of the focal point on the screen is determined by the PAN-TILT OFFSET values, composed of the "pan_offset" and "tilt_offset" variables. If the values are set to zero, the current focal point of the tracking appears in the middle of the screen. Setting "tilt_offset" to a positive causes a positive value offset in tilt_position calculation which, in turn, lowers the position of the focal point from the middle of the screen.

Those four variables are entered by the operator through the terminal. After this is done, the CPU reads the hall-effect sensors of the slave arm (through the A/D converter), and determines the angle at each joint. Then, it proceeds with the computation of the position of the camera (to have it point to the focal point), taking into accout "L6_offset", "pan_offset" and "tilt_offset": the derivation of the equations is presented in section 2.1. If a change in the position of the camera is required (depending on "window", as discussed above), the CPU calls for the pan_tilt1(pan_pos,tilt_pos,speed) function, which is defined in section 2.4.6. The computed values become the "pan_pos" and "tilt_pos" parameters; "speed" is set to its maximum value, that is 511. The CPU then repeats the cycle as shown in Figure 2.29, until the operator utters anything: when the operator does so, the CR-flag gets set, which eventually causes a return to the main program.

2.4.8 Arithmetic Card

The computation required to perform the automatic tracking is handled by the Arithmetic card. Two functions are used by the CPU to have the card execute operations on floating-point numbers: mathf1(op1,command) and mathf2(op1,op2,command). The two differ only in the number of operands needed for the operation: mathf1(op1,command) requires one operand (e.g. sine) whereas mathf2(op1,op2,command) requires two (e.g. multiplication).

Figure 2.30 shows the steps involved in performing an arithmetic or trigonometric function. The operand(s) is(are) pushed onto an internal stack and the appropriate command is issued. Then, the CPU reads the status register to find out when the operation is completed; when it is, the CPU retrieves the result from the stack ond returns to the calling program.

The floating-point format handled by the card is different from the IEEE format used by the C cross-compiler. Conversions between the two formats are carried out by the am_to_ieee(ieee) and ieee_to_am(ieee) functions. In particular, a conversion is necessary for constants which are defined at the compilation time; also, the library facilities to input/output floating-point numbers from/to the terminal/screen handle only the IEEE format.

Finally, flts(op1) and fixs(op1) are two functions used to convert 16-bit integers into floating-point numbers, and vice versa.

2.4.9 Monitor And QUIT Command

When the system's program is executed, the control is first managed by a monitor. The monitor offers the user a set of actions that can be implemented. Among the choices are:

- CLUSTER_DEFINITION (2): This reserves a portion of the recognizer's memory for templates of a group of words (see

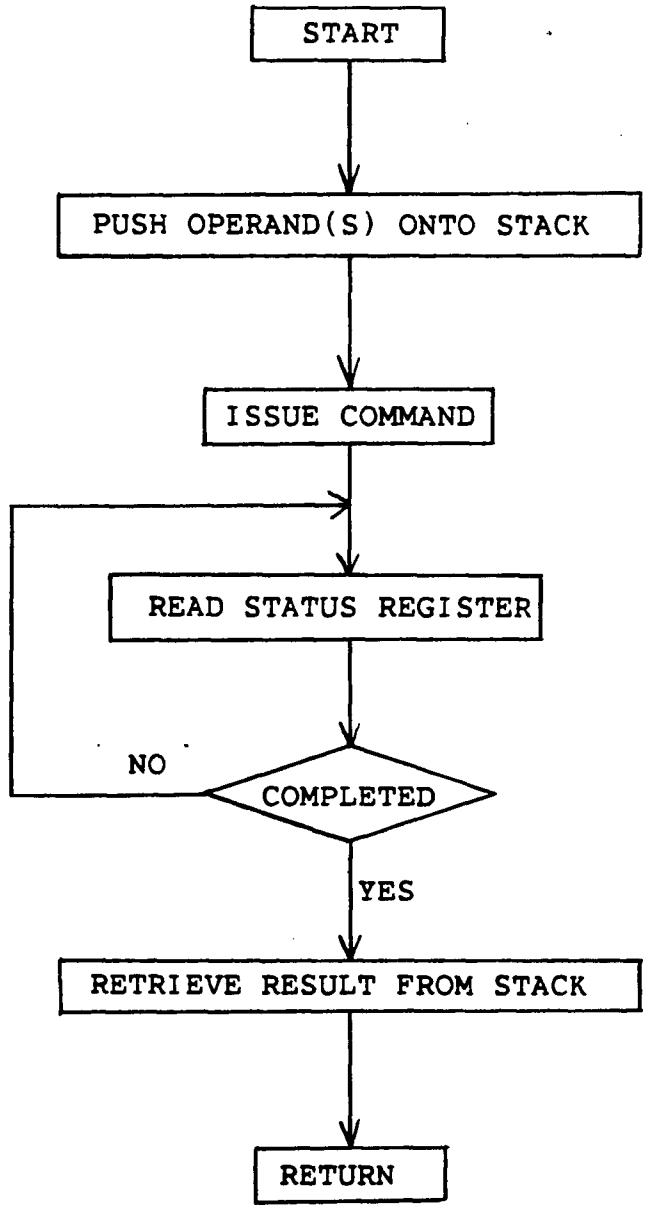


Figure 2.30: MATHF1(OP1,COMMAND), MATHF2(OP1,OP2,COMMAND) functions

section 2.4.2).

- TRAINING (3): To train (or re-train) the recognizer for an entire cluster or for a particular word of the cluster.
- RECOGNITION (4): To test the recognizer.
- CAMERA CONTROL (A): To pass the control over to the main program which interprets and executes commands to modify the video system.

At first, the operator must initialize the recognizer (choice-2) and train the speech recognizer for all the words of the vocabulary (choice-3). The operator may want to test the recognizer by uttering words of the vocabulary (choice-4 each time): if the recognizer can not properly recognize certain words, the operator can re-train the recognizer for those words (choice-3). Then, the operator can go on with the control of the video system (choice-A).

The QUIT command, when uttered in level 1 of the syntax, causes a return to the monitor. The command can be used to simply de-activate the video sytem control, or to re-train the recognizer for certain words that have been causing problems.

III. SYSTEM PERFORMANCE

This chapter gives details of the performance of the system described in Chapter II.

3.1 Pan-tilt Unit

Table 3.1 gives the angular speed (degrees/sec) of the two pan-tilt motors at different voltages (magnitude): the measures were taken while the camera and the zoom lens were sitting on the pan-tilt unit. It is noticed that a particular voltage does not produce the same speed for the two motors, as well as for the 2 directions of a same motor. In particular, the tilt motor runs faster (for V > 7 volts) than the pan motor, but exhibits a considerable difference in speed between the two directions of motion. A limited torque of the motors is the main cause of this effect.

VOLT (MAGN.)	12	11	10	9	8	7	6	5
PAN MOTOR(RIGHT)	52.0	46.5	40.0	34.3	28.2	23.0	17.7	12.8
PAN MOTOR(LEFT)	51.0	44.2	39.0	33.0	27.3	21.7	16.5	11.0
TILT MOTOR(UP)	57.4	52.5	45.6	39.4	33.3	27.4	21.0	14.8
TILT MOTOR(DOWN)	53.0	47.0	41.0	34.8	29.0	22.8	17.0	10.6

Table 3.1: Pan-tilt motors angular speed (DEG/SEC)

The operator may make use of three preset values to set the speed at which pan-tilt motions (using the RIGHT, LEFT, UP and DOWN commands) are executed. These values are: "SPEED_MIN", "SPEED_MED" and "SPEED_MAX". Similarly, the TRACK command uses the "SPEED_TRACK" value. These four values are integer constants sent to the D/A converters to produce the following voltages (magnitude) across the motors:

- SPEED_MIN = 230 ==> 4.43 volts
- SPEED_MED = 335 ==> 6.46 volts
- SPEED_MAX = 400 ==> 7.71 volts
- SPEED_TRACK = 511 ==> 9.85 volts

Another characteristic of a pan-tilt unit is the amount of backlash. The pan-tilt unit does not exhibit a backlash greater than 1 degree in amplitude of rotation for both motors.

3.2 Slave Arm Sensors

Readings were taken from the 5 slave arm hall-effect sensors. Then, equations relating voltages to angles were derived (according to a least square criterion) to fit the experimental data. The results, presented in Tables 3.2 to 3.6, show that the CPU can determine the joint angles of the slave arm within ±1.5 degrees accuracy by simply reading the sensors and using the equations.

SENSOR OUTPUT (VOLT)	OBSERVED ANGLE (DEG)	COMPUTED ANGLE (DEG)
2.19	-6.43	-6.70
2.02	-4.28	-4.68
1.83	-2.14	-2.40
1.64	0.00	-0.13
1.46	2.14	2.02
1.26	4.28	4.41
1.08	6.43	6.56
0.91	8.57	8.59
0.73	10.71	10.74
0.53	12.85	13.13
0.36	14.99	15.16
0.16	17.14	17.55
-0.02	19.28	19.65
-0.12	21.42	20.87
-0.37	23.56	23.89
-0.55	25.70	26.01
-0.73	27.85	28.12
-0.90	29.99	30.16
-1.07	32.13	32.25
-1.24	34.27	34.28
-1.39	36.41	36.07
-1.56	38.56	38.10
-1.72	40.70	39.95

EQUATION: ANGLE(DEG) = (-11.95 X VOLT) + 19.46

ACCURACY: ±0.75 DEGREE

Table 3.2: ARM SWING sensor

SENSOR OUTPUT (VOLT)	OBSERVED ANGLE (DEG)	COMPUTED ANGLE (DEG)
1.85	0.00	-0.38
1.56	2.24	1.83
1.23	4.48	4.34
0.94	6.72	6.55
0.63	8.98	8.92
0.33	11.25	11.20
0.00	13.54	13.72
-0.33	15.85	16.23
-0.63	18.19	18.52
-0.94	20.56	20.89
-1.24	22.97	23.17
-1.57	25.42	25.69
-1.89	27.92	28.13
-2.25	30.49	30.87
-2.55	33.12	33.16
-2.90	35.83	35.83
-3.30	38.64	38.87
-3.60	41.56	41.16

EQUATION: ANGLE(DEG) = (-7.62 X VOLT) + 13.72

ACCURACY: ±0.50 DEGREE

Table 3.3: SHOULDER sensor

SENSOR OUTPUT (VOLT)	OBSERVED ANGLE (DEG)	COMPUTED ANGLE (DEG)
4.37	55	53.69
4.10	49	49.19
3.86	45	45.31
3.55	40	40.50
3.25	35	36.01
2.86	30	30.47
2.46	25	25.13
2.02	20	19.63
1.55	15	14.20
1.08	10	9.24
0.63	5	4.93
0.11	0	0.47
-0.61	-5	-4.77
-1.11	-10	-10.02
-1.55	-15	-14.43
-2.12	-20	-20.16
-2.65	-25	-25.48
-3.15	-30	-30.50
-3.60	-35	-35.02
-4.11	-40	-40.14
-4.55	-45	-44.56

EQUATION:

ANGLE(DEG) = (1.05 X VOLT X VOLT) + (7.80 X VOLT) - 0.40 IF VOLT >= -0.61

ANGLE(DEG) = (10.04 X VOLT) + 1.13 OTHERWISE

ACCURACY: ±1.5 DEGREES

Table 3.4: ELBOW sensor

SENSOR OUTPUT (VOLT)	OBSERVED ANGLE (DEG)	COMPUTED ANGLE (DEG)
3.88	75	73.43
3.68	70	70.01
3.43	65	65.74
3.15	60	60.94
2.82	55	55.30
2.56	50	50.85
2.20	45	44.69
1.87	40	39.58
1.36	35	34.11
0.91	30	29.29
0.54	25	25.32
0.14	20	21.04
-0.40	15	15.25
-0.82	10	10.75
-1.92	0	1.03
-2.85	-10	-11.00
-3.33	-15	-16.14
-3.70	-20	-20.10
-3.76	-21	-20.75

EQUATION:

ANGLE(DEG) = (17.11 X VOLT) + 7.04 IF VOLT > 1.87

ANGLE(DEG) = (10.71 X VOLT) + 19.54 OTHERWISE

ACCURACY: ±1.5 DEGREES

Table 3.5: WRIST YAW sensor

SENSOR OUTPUT (VOLT)	OBSERVED ANGLE (DEG)	COMPUTED ANGLE (DEG)
3.37	53.14	51.72
3.08	48.71	48.19
2.73	44.29	43.92
2.45	39.86	40.50
2.07	35.43	35.87
1.73	31.00	31.73
1.37	26.57	27.34
0.97	22.14	22.46
0.62	17.71	18.19
0.20	13.29	13.07
0.15	8.86	8.80
-0.53	4.43	4.17
-0.90	0.00	-0.34
-1.26	-4.43	-4.73
-1.60	-8.86	-8.87
-1.92	-13.29	-12.78
-2.22	-17.71	-16.43

EQUATION: ANGLE(DEG) = (12.19 X VOLT) + 10.63

ACCURACY: ±1.5 DEGREES

Table 3.6: WRIST PITCH sensor

3.3 Camera Lenses

The resolution of the zoom lens (at f = 45mm) and the wide angle lens were tested and found to be 550 and 450 lines respectively. It was felt that the difference in resolution would not affect the results of the experiments and, therefore, we proceeded with the experiments using those two lenses.

3.4 Speech Recognizer

Accuracy and minimum pause were two aspects of the speech recognizer that were tested.

3.4.1 Accuracy

NEC claims that the SR-100 speech recognizer typically exhibits an accuracy greater than 99% [17]. However, no details are given on the setup of their tests: size and choice of the vocabulary, background noise, how familiar the subjects were with the speech recognizer, etc.

No direct measures of the system accuracy could be taken during the experiments because only a very small portion of the vocabulary was actually used. Separate tests were then performed on the speech recognizer to estimate its accuracy. Eight subjects were asked to speak each word of the vocabulary presented in Table 2.1 (with the word READY added to it) twice,

and at random. The subjects had had some previous experience with the recognizer through the experiments (using the digits and GO TO), but had never used the rest of the words of the vocabulary. The background was noiseless. Defining the accuracy of the recognizer by the equation

$$A = \frac{\text{words properly recognized}}{\text{words uttered}}$$

, the tests give the following results:

- best performance = 98.5%
- worst performance = 85.3%
- overall performance = 92.6%

This is particularly a good performance considering the little training of the subjects prior to the tests, and also that no special study has been done to select an optimum vocabulary.

It is generally believed that concatenated words increase the accuracy of recognition [15]. However, the tests revealed the opposite effect. This can be explained by the fact that concatenated words put tighter constraints (e.g. speech rate, intonation) on the user than regular words, and that only after the user gets familiar with uttering those words can the benefits of using them be felt.

3.4.2 Minimum Pause

The minimum pause is specified by the amount of time between the end of an utterance and when the recognizer is re-enabled by the CPU for another utterance. A word partly uttered before the recognizer is enabled leads to a rejection or misrecognition, and nothing results from a word that is completely uttered before the recognizer is enabled.

Figure 3.1 shows the steps involved in the process of re-enabling the recognizer after an utterance is spoken. When the recognizer detects the end of an utterance, it starts comparing the template of the speech signal with the reference templates. This recognition process takes approximately 300 ms (according to the instruction manual [17]). The result of the process, which consist of 20 bytes in all, is sent over the RS232 communication channel to the CPU. With a baud rate of 9600 and a 10-bit/byte format, 21 ms elapse to complete the transmission.

The reception of the result is accomplished by the CPU through interrupt (see section 2.4.2). After the last byte is stored, the CPU sets a flag (CR-flag) and resumes its normal operation. While in its normal operation, the CPU constantly checks the CR-flag to find out if the recognizer has sent anything. When it is found set, all current actions are immediately terminated and the CPU returns to the main program

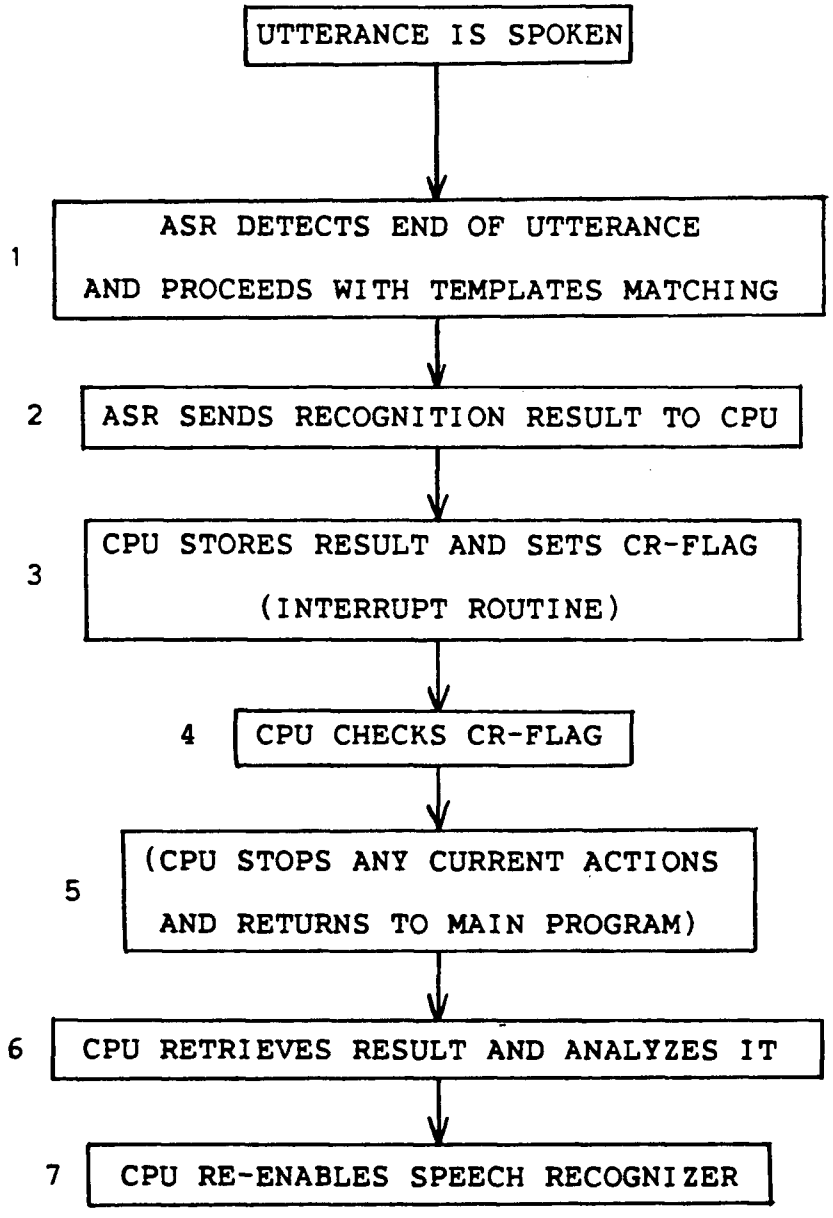


Figure 3.1: Steps involved before re-enabling the speech recognizer

(if not already in it). The result is then retrieved and analyzed. Finally, the speech recognizer is re-enabled and actions are taken according to the recognition result.

The amount of time required to go through steps 4-6 is not constant, but depends on what the CPU was doing just before the reception of the result. In some instances, the CPU may just be waiting for an utterance; in others, the CPU may be driving the motors, computing the automatic tracking, etc. Thus, the amount of time between two consecutive checks of the CR-flag differs within the program. However, a typical minimum pause for the system is estimated to be approximately 1.5 seconds.

Comparing this time (i.e. 1.5 seconds) with the 321 ms taken by the recognizer to give the result of a recognition, one can conclude that the speech recognizer is not the predominantly limiting factor in the rate words can be uttered.

3.5 Accuracy In Positioning The Pan-tilt Unit

The CPU uses the pan_tilt1(pan_pos,tilt_pos,speed) function to position the pan-tilt unit into any desired position. The flow chart of the function is shown in Figure 2.28 (section 2.4.6). When called, this function moves the pan-tilt unit into a position specified by the "pan_pos" and "tilt_pos" parameters, at a speed determined by the "speed" parameter. It involves driving the pan-tilt motors, reading its potentiometers, and

stopping the motors when the reading of their respective potentiometer coincide with the desired position. This function finds two applications in the system:

- with the GO TO command, when a certain position of the pan-tilt unit (which was previously memorized) is recalled.
- with the automatic tracking mode, when the pan-tilt unit is moved into a position that has been computed.

Figure 3.2 shows the setup of the tests that were implemented to evaluate the accuracy of the function. A board with equally-distant concentric circles was standing some distance in front of the camera. The video monitor was marked with cross-hairs in the middle of the screen. The camera was set in a positon such that the center of the circles on the board coincided with the cross-hairs on the screen: this position was memorized by the system (using the RECORD...ZERO command). The camera was then moved into anyone of four positions that make an angle of 45 degrees with the memorized position, and all equally distant from it. The memorized position was then recalled (using the GO TO...ZERO command): pan_tilt1() would move the camera back into position ZERO, or as close to it as possible. By looking at the position of the center of the circles with respect to the cross-hairs (and knowing the distance between the camera and the board, as well as between the circles), one can calculate the deviation from the original position ZERO.

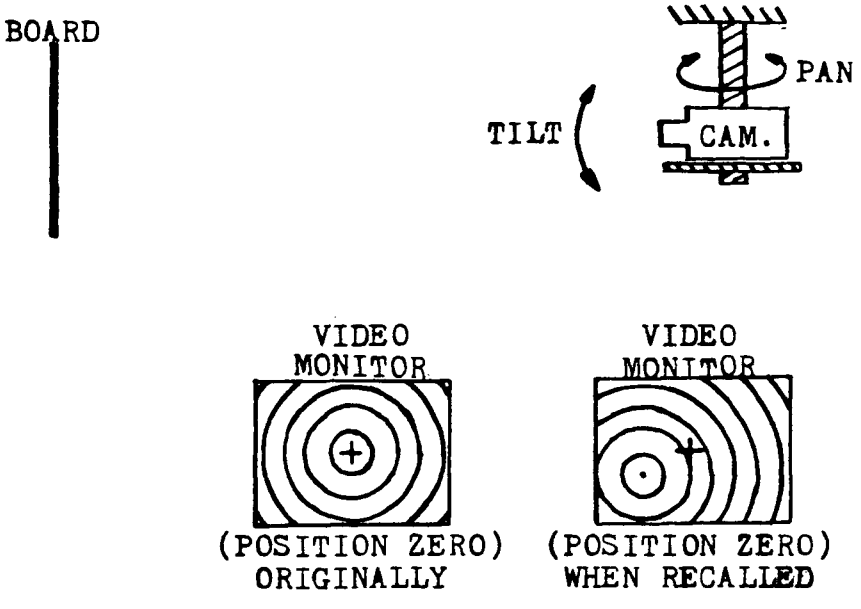


Figure 3.2: Setup for evaluating the system's positioning accuracy

SPEED	BEST ACCURACY (DEG)	WORST ACCURACY (DEG)	OVERALL ACCURACY (DEG)
SPEED_MIN	0.6	1.0	0.7
SPEED_MED	0.4	0.8	0.5
SPEED_MAX	0.4	1.5	0.9
SPEED_TRAC	0.2	2.3	1.3

Table 3.7: PAN_TILT1() function accuracy

The function was tested under 4 different speeds, which are specified by the "SPEED_MIN", "SPEED_MED", "SPEED_MAX" and "SPEED_TRACK" constants (see section 3.1). For each speed, the position ZERO was recalled 6 times from each of the 4 positions. The results, summed up in Table 3.2, confirm that the system can at least achieve a ±2.5 degrees accuracy in positioning the pan-tilt unit, in any circumstances.

3.6 Automatic Tracking Characteristics

The track(speed) function manages the automatic tracking capability of the system. Figure 2.29 (see section 2.4.7) shows its flow chart, and section 2.1 presents the derivation of the equations relating the orientation of the camera to the joint angles of the slave arm.

3.6.1 Maximum Tracking Speed

The fastest motion of the slave arm that the camera can track depends on two factors:

- the speed that the pan-tilt unit can achieve,
- the amount of time required to compute what the position of the camera should be.

Every time the pan-tilt unit needs to be re-positioned, the track(speed) function calls the pan_tilt1() function to carry out the task. The "speed" parameter is set to its maximum

value, that is 511 ("SPEED_TRACK"). As discussed in section 3.1, a value of 511 produces a voltage of 9.85 volts across the motors: according to Table 3.1, an angular speed of approximately 40 deg/sec results from such a voltage.

Much computation is required to calculate what the position of the pan-tilt unit should be in order to have the camera pointing towards the desired focal point (normally the gripper of the slave arm). The use of an Arithmetic card permits the computation to be performed in approximately 58 ms.

The system is thus capable of tracking any motions without any noticeable lag of angular speed smaller than 20 deg/sec. Our experience shows that this is more than sufficient for normal operation of the slave arm.

3.6.2 Tracking Accuracy

Three factors are responsible for the accuracy in the tracking of the slave arm by the camera:

- Accuracy in reading the slave arm sensors to obtain the joint angles.
- Accuracy in solving the mathematic equations for the computation of the position of the pan-tilt unit.
- Accuracy in moving the camera into the computed position.

As can be seen from Tables 3.2 to 3.6 (section 3.2), the joint angles can be obtained with a ±1.5 degrees accuracy.

All trigonometric and arithmetic functions are implemented on 32-bit floating-point numbers, and exhibit relative errors in the order of 10 EXP -7. The result is converted to an integer: the round-off error in the conversion process represents less than 0.3 degree of rotation.

As it was discussed in section 3.5, test showed that the pan-tilt unit could be moved into any desired position with an accuracy of ±2.5 degrees.

The effect of the second factor is thus negligible compared to the two other factors. It becomes very tedious to try to evaluate the overall accuracy of the tracking mode, because of the interaction that exists between the two significant factors.

An easier solution to estimate the accuracy was obtained through the following procedure. The zoom lens was set to a focal length of 45 mm. The tracking mode was enabled and the focal point was programmed to be the tip of the gripper. Then, the slave arm was moved all around its work volume and the motion was recorded. The playback of the test showed that the tip of the gripper always remained at a distance less than 4 cm from the middle of the screen (which was marked with cross-hairs). The screen, which is 17.78 cm wide, covers a

14-degrees field of view when the focal length is set to 45 mm. Therefore, one estimates the accuracy of the tracking to be in the order of ± 3 degrees.

IV. DESCRIPTION OF EXPERIMENTS

Experiments were carried out to compare the effects of 4 different modes of operating the video system for two simulated subsea tasks. The four modes are manual control mode, automatic tracking mode, voice-operated mode and fixed-camera-position mode.

The zoom lens was utilized for the first three modes, but its focal length was set to a certain value and remained at that value (45 mm) throughout the experiments: it was therefore used as a close-up lens more than as a zoom lens. Since the field of vision is limited with a close-up lens, the operator needed to re-orient the camera with the pan-tilt unit as the slave arm was moved. In contrast, the last mode freed the operator from having to control the pan-tilt unit since it made use of the 8-mm wide angle lens.

Manual control mode

In the manual control mode, the subject modified the orientation of the pan-tilt unit via a control box. The control box, which consists of 4 momentary push buttons, remained in a constant position to the right of the subject. The subject's right hand was used to manipulate the master arm as well as to access the control box. When a change on the camera orientation was required, the subject first held the master arm still (in order to immobilize the slave arm) with his left hand; then, his

right hand reached to the control box and pressed the appropriate push buttons (Figure 2.10); when done, the right hand returned to the master arm and the left hand let go.

The reason for such a scheme for accessing the control box is two-fold. First, the subsea teleoperator environment simulated in these experiments is one where the operator has both hands busy: one with the control of the submarine, the other with the control of the slave arm. Second, there was no "freezing" capability on the master arm: this was then accomplished using the subject's left hand.

Automatic tracking mode

In the automatic tracking mode, the camera automatically tracked the slave arm through the automatic tracking feature of the computer-based system (see section 2.4.7). The subject could then concentrate uniquely on the task of moving the slave arm. The focal point was set to suit the needs of each experiment: the tip of the pointer was the focal point in the first experiment; in the second experiment, the camera tracked the gripper with a "TILT_OFFSET" of approximately -3 degrees so that the whole object (when carried by the gripper) could be seen on the screen as well as the gripper. In both experiments, "WINDOW" was set to zero.

Voice-operated mode

In the voice-operated mode, the pan-tilt unit orientation

was verbally controlled. After training the speech recognizer to his own voice, the subject could then utter commands to move the pan-tilt unit. The "memorization" feature of the system (see section 2.4.6) was used, and appropriate positions of the pan-tilt unit were memorized by the system beforehand. In particular, 8 positions labelled from ZERO to EIGHT along the whole path were memorized for the first experiment; in the second experiment, the locations of the bucket and of the stool (with the objects on it) were memorized and given the labels BUCKET and OBJECT (in lieu of NINE and TEN). By uttering the command GO TO followed by one of the position labels, the pan-tilt unit would immediately move into the specified position.

Fixed-camera-position mode

In the fixed-camera-position mode, the position of the pan-tilt unit was held constant, since the field of vision exhibited by the wide angle lens allowed the subject to see everything (i.e. the whole path, or the bucket and the stool) all at once. Thus, there was no need to operate the pan-tilt unit, and the subject's attention could be uniquely on the task.

4.1 Experiment #1

4.1.1 Object

The object of the first experiment was to compare the effects of the 4 camera control modes mentioned above for a simulated subsea tracking task.

4.1.2 Method

Task

A line, resembling the letter "G", was drawn on a board and a "follower" line was drawn X/2 cm (where X is the path bandwidth) on each side (Figure 4.1). A pointer was attached to the end of the gripper of the slave arm. The subject's task was to have the slave arm follow the path from one end to the other, back and forth: by looking at the video monitor (Figure 2.9), the subject moved the slave arm (through the master arm) and tried to keep the tip of the pointer inside the path, without going outside the follower lines. Sixteen subjects participated in the experiment.

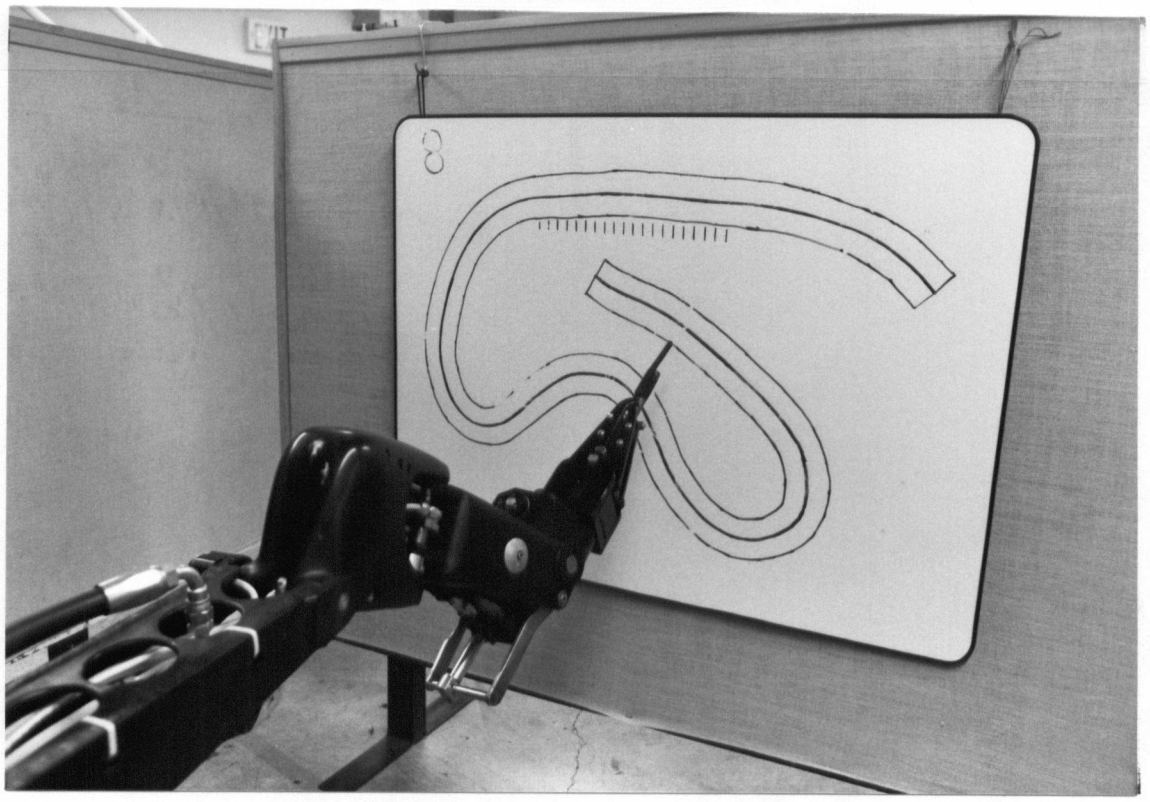


Figure 4.1: Experiment #1 --> Tracking task

Experimental design

The first factor was the camera control mode:

- manual control mode (a)
- automatic tracking mode (b)
- voice-operated mode (c)
- fixed-camera-position mode (d)

The order of presentation of each of the four modes (labelled a, b, c and d respectively) was randomized according to a Latin square scheme as indicated below [19]:

a b c d	d c b a	c a d b	b d a c
b c d a	c b a d	a d b c	d a c b
c d a b	b a d c	d b c a	a c b d
d a b c	a d c b	b c a d	c b d a

Using such a scheme, each mode was presented first as often as the second, third or fourth. Thus, the first subject would be presented with mode a, then b, then c and finally d; the second subject would see mode b first, then c, then d and finally a; etc.

The second factor was the path bandwidth, that is X. Bandwidths of 2, 4, 8 and 16 cm were chosen to simulate different degrees of fineness for a tracking task. They too were presented in a random order according to a Latin square:

2 4 8 16	16 8 4 2	8 2 16 4	4 16 2 8
4 8 16 2	8 4 2 16	2 16 4 8	16 2 8 4
8 16 2 4	4 2 16 8	16 4 8 2	2 8 4 16
16 2 4 8	2 16 8 4	4 8 2 16	8 4 16 2

Specifically, subject 1 would see mode a with bandwidth 2, 4, 8 and 16; mode b with bandwidth 4, 8, 16 and 2; mode c with bandwidth 8, 16, 2 and 4; mode d with bandwidth 16, 2, 4 and 8. Subject 2 would see mode b with bandwidth 16, 8, 4 and 2; mode c with bandwidth 8, 4, 2 and 16; etc. The overall experimental design was therefore a 4X4 factorial design [19,20].

Measurements

Elapsed time taken to travel along the path back and forth was measured using a stopwatch.

A measure of error was obtained by considering the amount of travel outside the follower lines. To this end, lines distant by 2 cm from each other were also drawn on the board (Figure 4.1), and the visual information displayed on the video monitor was being recorded on tape at the same time. The lines gave the necessary perspective needed for figuring distances for the playbacks: using a caliper, one could determine the amount of travel outside the bands.

The combination of the two measures provided information on the speed-accuracy tradeoffs involved.

4.1.3 Analysis

The analysis consisted of a fixed-effect two-way ANOVA with Multiple Classification Analysis (MCA) [19,20]. Effects of the 2 factors (i.e. camera control mode and path bandwidth), as well as their interactions, were then obtained in terms of speed (through the elapsed time measure) and accuracy (through the error measure). Scattergrams of elapsed time vs error were also examined for the different modes and bandwidths.

Note that the mode factor is fixed while the bandwidth factor is random. Therefore, a mixed-model analysis (as opposed to a strictly fixed model) would have been appropriate. Unfortunately, SPSS (Statistical Package for the Social Sciences) available on the university MTS computer system

provides summary tables for fixed-effect two-way models only [21]. Although this analysis slightly overestimated the significance, it was deemed adequate and marginal results were discarded.

4.2 Experiment #2

4.2.1 Object

The object of the second experiment was to compare the effects of the 4 camera control modes for a simulated subsea pick-and-drop task.

4.2.2 Method

Task

A stool (with 4 objects on it) and a bucket were placed in front of the slave arm (Figures 4.2, 2.5). The subject used the manipulator to pick up each object and drop it into the bucket. If an object did not fall into the bucket, it was placed back onto the stool by the experimenter and the subject had to try again. The same number of subjects as in experiment #1 participated, since the two experiments were run concurrently.

Experimental design

The only factor involved in the experiment is the camera control mode. As before, each of the 4 levels of the factor was randomized according to a Latin square. Thus, the overall

experimental design was a 4-level single-factor design.

Measurements

Only elapsed time taken to successfully drop the 4 objects into the bucket was considered.

4.2.3 Analysis

The analysis consisted of a one-way fixed-effect ANOVA with Multiple Classification Analysis (MCA). Scattergrams of elapsed time vs mode were also examined.

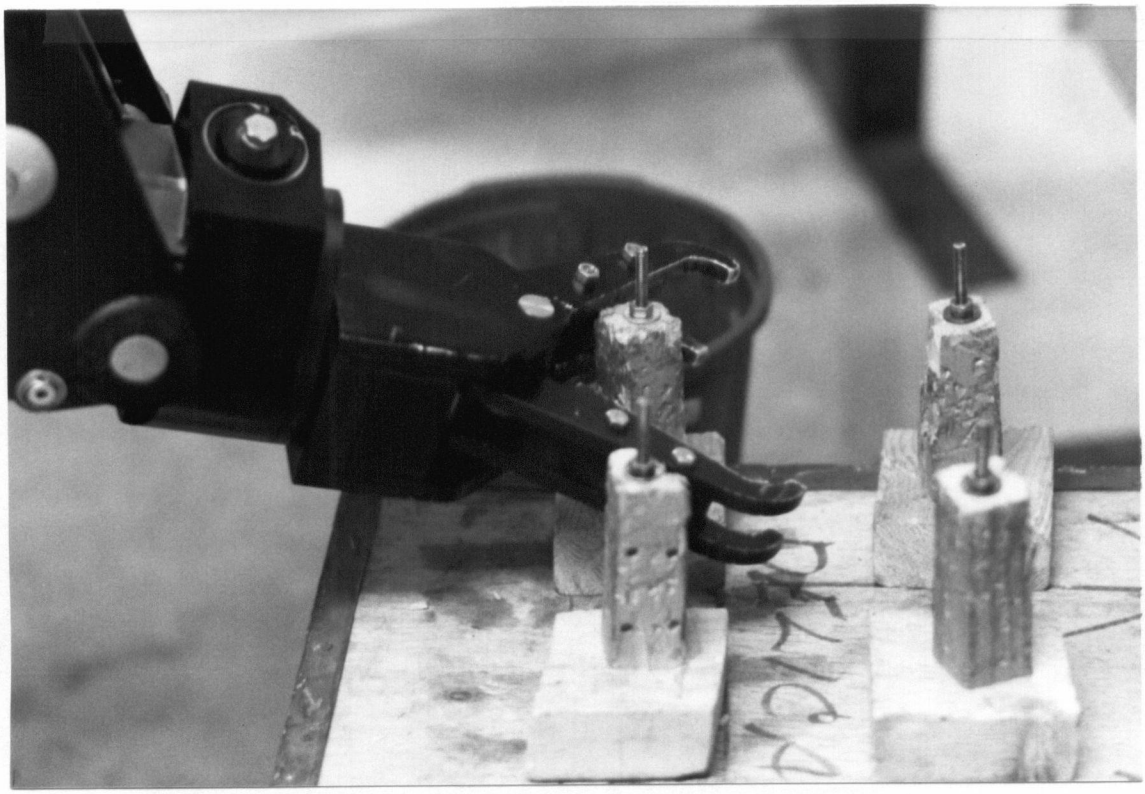


Figure 4.2: Experiment #2 --> Pick-and-drop task

V. DATA AND RESULTS

5.1 Experiment #1

The effects of the two factors of the experiment (i.e. camera control mode and path bandwidth) were analyzed with respect to each measure (i.e. elapsed time and error) using two-way fixed-effect ANOVA's, with Multiple Classification Analysis (MCA) [19,20].

ANOVA utilizes an F test statistic: F represents the ratio of the between-groups variance to the within-groups variance, and is also a measure of differences between group means. Specifically, the larger the ratio, the more different the means. The degree of difference is expressed by P, which is obtained from the sampling distribution of F: a larger value of F leads to a smaller value of P. The group means are said to be significantly different when the value of P is smaller than a certain level of significance (determined by the experimenter). A level of 0.05 was selected for the experiment. Thus, when P is smaller than 0.05, one can assert that at least two group means are significantly different and that the difference is due to the corresponding modes or bandwidths. Note that the ANOVA analyzes the two factors separately as well as the combination of the two factors: the latter is referred to as the main effects .

Once the means are found to be significantly different, a multiple comparison procedure (MCA) is used to determine how they differ. To this end, the grand mean is first calculated by averaging all the scores of a measure over all possible combinations of the different levels of the two factors. Then, the average of the scores of the measure over each level of each factor is computed separately and subtracted from the grand mean: these values are referred to as the unadjusted deviations.

Measures of elapsed time and error taken during the experiment are compiled in Table 5.1. Table 5.2 summarizes the results of the ANOVA of ELAPSED TIME by CAMERA MODE and PATH BANDWIDTH. Table 5.3 presents the results of the MCA performed on the same parameters. The complete results of the ANOVA and MCA analyses are found in Appendix A.

DAY	MODE	T-1	T-2	T-3	T-4	E-1	E-2	E-3	E-4
2 0	1	273	178	162	135	025	019	000	000
2 0	2	100	064	045	031	017	003	000	000
2 0	3	107	094	071	062	046	002	006	000
2 0	4	058	042	031	026	194	026	011	000
2 1	1	176	115	089	087	101	054	013	012
2 1	2	109	085	061	037	020	007	000	000
2 1	3	084	068	068	046	066	022	009	001
2 1	4	074	051	035	023	118	064	013	007
1 2	1	142	154	115	095	071	014	002	004
1 2	2	061	037	027	020	023	014	016	024
1 2	3	133	065	107	072	013	017	005	000
1 2	4	055	034	025	018	077	045	031	000
2 2	1	224	195	142	137	067	027	004	000
2 2	2	095	064	048	025	062	035	000	000
2 2	3	103	078	056	048	070	023	013	000
2 2	4	039	038	029	020	115	102	046	008
1 3	1	141	119	076	076	023	008	007	000
1 3	2	063	046	036	026	035	023	006	000
1 3	3	144	095	065	067	031	002	001	000
1 3	4	089	069	054	032	203	069	030	000
2 3	1	145	182	135	096	078	079	006	000
2 3	2	054	038	035	023	078	061	008	000
2 3	3	124	084	070	068	016	008	000	000
2 3	4	051	042	030	023	100	050	023	021
1 4	1	272	180	136	188	044	011	000	000
1 4	2	100	101	051	034	098	012	041	000
1 4	3	124	142	101	094	016	007	001	000
1 4	4	095	076	064	048	169	030	011	000
1 5	1	245	247	130	129	036	018	002	000
1 5	2	123	069	049	026	010	013	004	000
1 5	3	154	078	089	058	047	011	007	004
1 5	4	125	073	041	030	078	021	006	000
1 6	1	202	164	140	095	007	004	003	000
1 6	2	071	046	034	029	070	033	004	002
1 6	3	197	092	072	095	008	011	007	000
1 6	4	072	036	037	016	040	038	002	004
2 6	1	195	267	114	144	050	026	013	003
2 6	2	082	073	046	032	051	011	031	020
2 6	3	104	087	073	091	075	026	005	000
2 6	4	105	062	048	021	075	057	034	019
2 7	1	165	132	117	088	053	028	006	000
2 7	2	066	046	030	024	095	059	046	007
2 7	3	097	067	058	060	057	014	005	000
2 7	4	092	045	035	022	065	051	007	017
1 1	1	204	157	148	138	038	006	001	001
1 1	2	229	161	051	040	004	002	002	000
1 1	3	194	122	089	067	001	001	000	000
1 1	4	112	068	043	032	034	002	000	000
1 8	1	262	215	179	159	020	008	008	000
1 8	2	135	064	046	031	062	042	005	000
1 8	3	194	127	091	128	009	006	001	000

(Table 5.1: cont'd next page)

DAY	MODE	T-1	T-2	T-3	T-4	E-1	E-2	E-3	E-4
		(cont'd from previous page)							
1 8	4	097	054	038	025	061	062	018	000
2 8	1	230	153	144	091	018	001	000	000
2 8	2	101	060	037	028	028	001	014	002
2 8	3	124	193	066	068	018	000	000	000
2 8	4	104	072	048	043	045	023	015	017
1 9	1	353	224	210	144	001	000	000	000
1 9	2	201	086	051	045	001	001	000	000
1 9	3	205	133	092	097	001	000	000	000
1 9	4	133	101	057	030	040	000	000	000
2 9	1	143	127	099	087	067	008	008	000
2 9	2	134	094	072	058	026	008	000	000
2 9	3	185	139	123	083	034	022	000	007
2 9	4	082	045	038	028	057	004	018	002

DAY
- First digit ==> 1=Morning, 2=Afternoon
- Second digit ==> day of the experiment (0-9)

MODE
- 1-> Manual control mode
- 2-> Automatic tracking mode
- 3-> Voice-operated mode
- 4-> Fixed-camera-position

T-1,T-2,T-3,T-4
- Elapsed time measures (seconds) for bandwidths of 2cm, 4 cm, 8cm and 16 cm respectively.

E-1,E-2,E-3,E-4
- Error measures (cm) for bandwidths of 2 cm, 4 cm, 8 cm and 16 cm respectively.

Table 5.1: Data of experiment #1

ANOVA(ELAPSED TIME by CAMERA MODE,PATH BANDWIDTH)

SOURCE OF VARIATION	F	P
Main effects	105.320	<0.0005
• camera mode	144.061	<0.0005
• path bandwidth	66.579	<0.0005
2-way interactions	0.921	0.507

Table 5.2: Results of ANOVA(TIME by MODE,BANDWIDTH) of experiment #1

MCA(ELAPSED TIME by CAMERA MODE,PATH BANDWIDTH)

GRAND MEAN = 94.16 s

	UNADJUSTED DEVIATION
CAMERA MODE:	
• Manual control	65.31 s
• Automatic tracking	-30.32 s
• Voice-operated	6.34 s
• Fixed-camera-position	-41.33 s
PATH BANDWIDTH:	
• 2 cm	42.59 s
• 4 cm	7.64 s
• 8 cm	-19.18 s
•16 cm	-31.05 s

Table 5.3: Results of MCA(TIME by MODE,BANDWIDTH) of experiment #1

The ANOVA indicates that the variability between scores of elapsed time is attributable to the camera mode as well as the path bandwidth: $P < 0.0005$ for both factors taken separately as well as taken together. Also, we note that the two factors do not interact, as indicated by $P = 0.507$ (which is greater than the 0.05-level of significance). This non-interaction is best seen by considering the mean values of elapsed time for all combinations of the two factors. These values, compiled in Table 5.4, are shown in Figure 5.1.

	2-cm	4-cm	8-cm	16-cm
Manual control	211	176	134	118
Automatic tracking	108	71	45	32
Voice-operated	142	104	81	75
Fixed-camera-position	86	57	41	27

Table 5.4: Mean values of elapsed time (s) for all combinations of the two factors

According to the MCA, the manual control mode is the slowest mode (with an average of 65.31 seconds above the grand mean); then the voice-operated mode comes second; and finally the automatic tracking mode and the fixed-camera-position mode are the fastest modes, the latter being slightly faster. These results were obtained by considering the four bandwidths together; however, similar conclusions hold for each bandwidth taken separately, since the ANOVA did not find any interactions between the two factors (see also Figure 5.1).

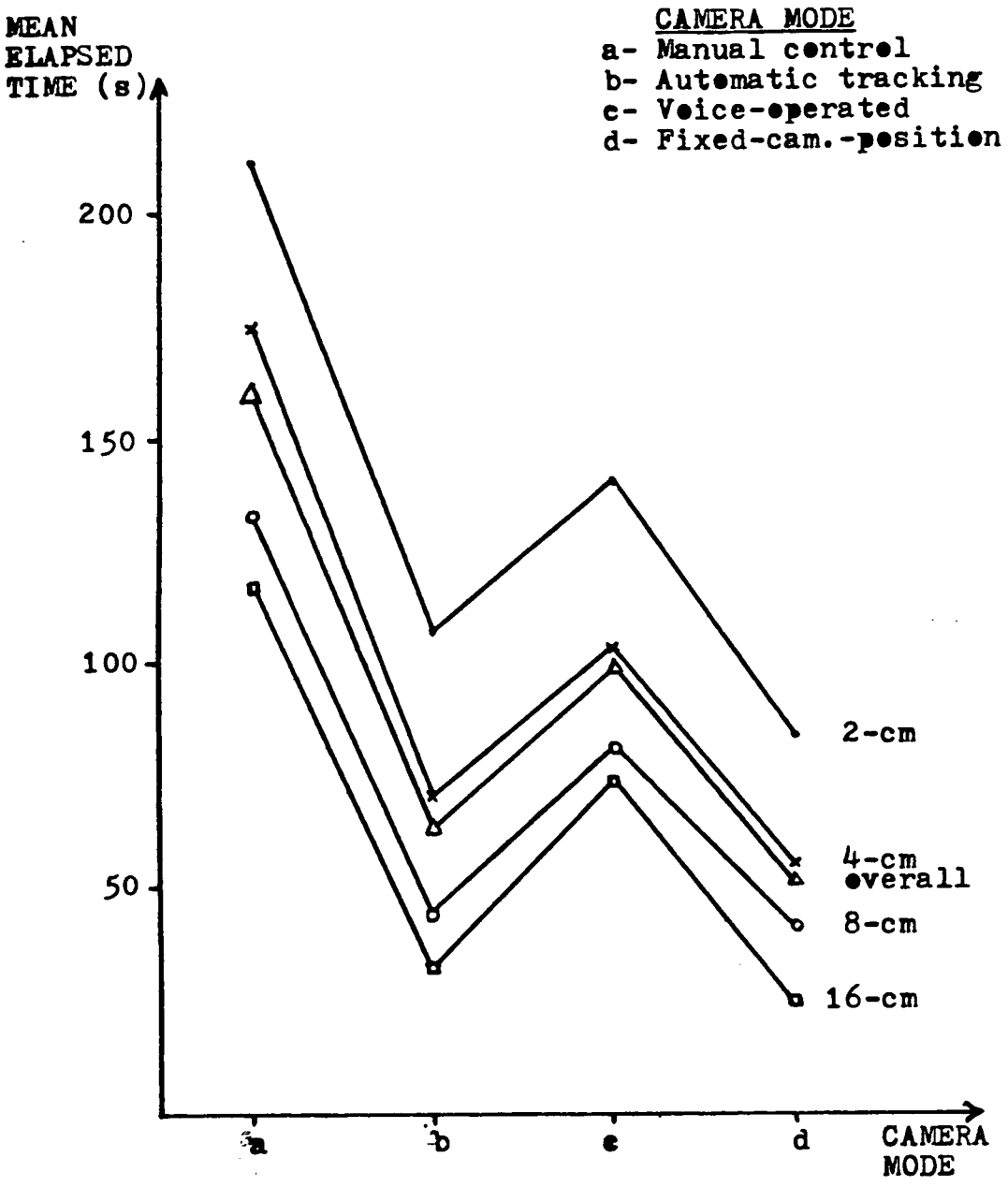


Figure 5.1: Non-interaction between MODE and BANDWIDTH (in terms of elapsed time)

As far as the path bandwidth is concerned, the results obtained from the MCA were expected: as the bandwidth of the path increases, the elapsed time decreases. In other words, the more precision required by a task, the longer it takes to accomplish the task. We note however that the difference between the results for a bandwidth and for its next one in size decreases as the bandwidth increases: specifically, a difference of 34.95 seconds is observed between the results for the the 2 cm and the 4 cm bandwidths, as opposed to 26.82 seconds between 4 cm and 8 cm, and 11.87 seconds between 8 cm and 16 cm. This suggests that the path bandwidth (i.e. the degree of fineness of a task) loses its significance in terms of elapsed time as the bandwidth increases (i.e. as the degree of fineness decreases).

Tables 5.5 and 5.6 summarize the results of the ANOVA and the MCA conducted on ERROR by CAMERA MODE and PATH BANDWIDTH. Complete results of the analyses are found in Appendix A.

Again, the ANOVA indicates that the camera mode and the path bandwidth are significant factors with respect to the error measure: specifically, the significance of both factors is maximum (i.e. $P < 0.0005$), as before. However, a strong interaction between the two factors is observed in this case ($P < 0.0005$). The mean values of error for all combinations of the two factors are compiled in Table 5.7. Figure 5.2, which shows these mean values, displays the interaction.

ANOVA(ERROR by CAMERA MODE,PATH BANDWIDTH)

SOURCE OF VARIATION	F	P
Main effects	41.909	<0.0005
• camera mode	18.420	<0.0005
• path bandwidth	65.398	<0.0005
2-way interactions	4.114	<0.0005

Table 5.5: Results of ANOVA(ERROR by MODE,BANDWIDTH) of experiment #1

MCA(ERROR by CAMERA MODE,PATH BANDWIDTH)

GRAND MEAN = 21.75 cm

	UNADJUSTED DEVIATION
CAMERA MODE:	
• Manual control	-4.51 cm
• Automatic tracking	-2.42 cm
• Voice-operated	-10.00 cm
• Fixed-camera-position	16.93 cm
PATH BANDWIDTH:	
• 2 cm	30.72 cm
• 4 cm	0.94 cm
• 8 cm	-12.76 cm
•16 cm	-18.90 cn

Table 5.6: Results of MCA(ERROR by MODE,BANDWIDTH) of experiment #1

	2-cm	4-cm	8-cm	16-cm
Manual control	44	19	5	1
Automatic tracking	43	20	11	3
Voice-operated	32	11	4	1
Fixed-camera-position	92	40	17	6

Table 5.7: Mean values of error (cm) for all combinations of the two factors

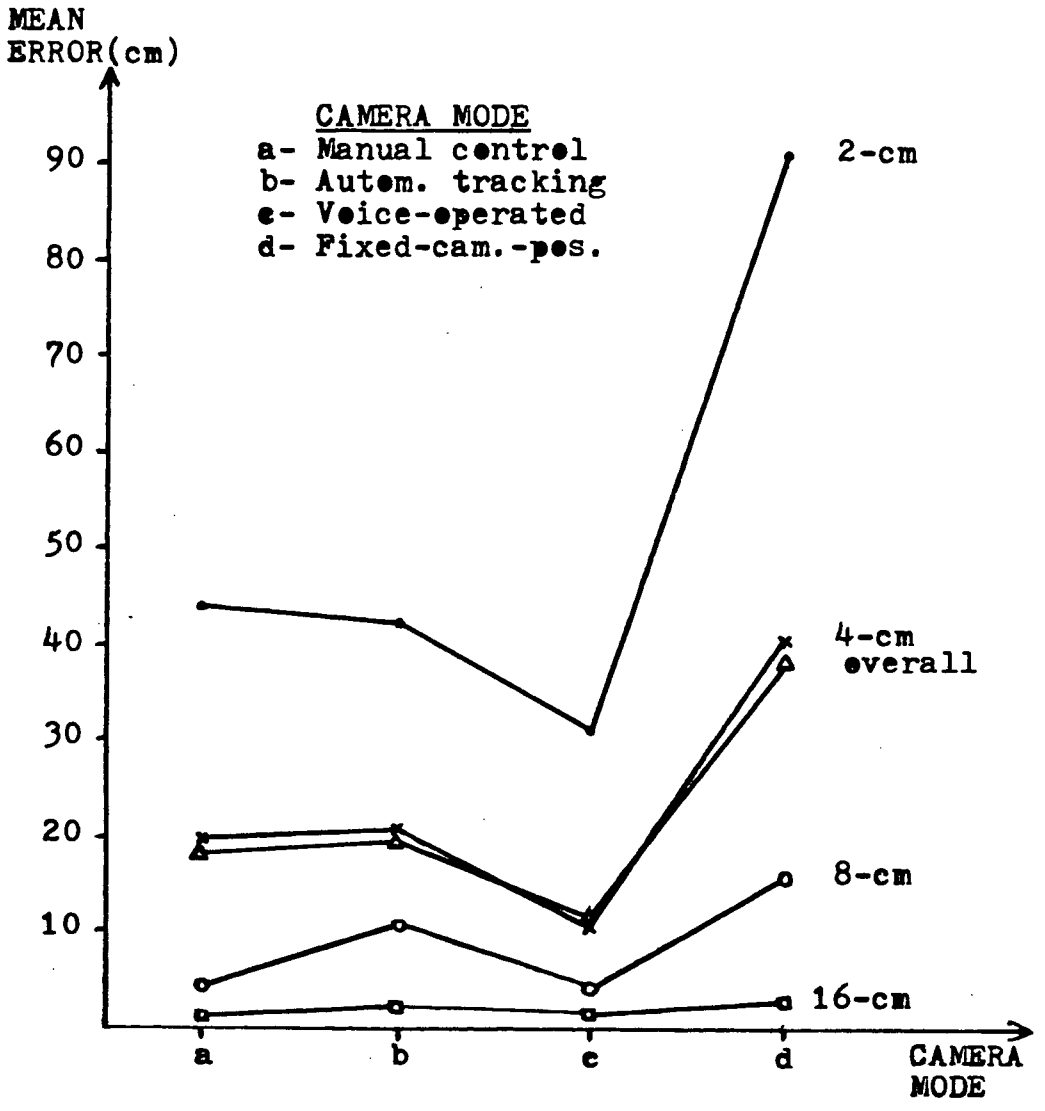


Figure 5.2: Interaction between MODE and BANDWIDTH (in terms of error)

The MCA reveals that the fixed-camera-position mode is the least precise mode, followed by the automatic tracking mode and the manual control mode (the latter being slightly more accurate), and voice-operated mode is by far the most precise mode. For instance, the fixed-camera-position mode exhibits an average error of 16.93 cm above the grand mean, as opposed to an average of 10.00 cm below the grand mean for the voice-operated mode.

Because the ANOVA showed the evidence of interactions between the two factors, it is important to keep in mind that the rating (in terms of accuracy) previously mentioned is valid only as one considers the four bandwidths together: the rating is not necessarily valid under each bandwidth, or the degree of difference may vary and lose its significance for different bandwidths. However, Figure 5.2 shows that the rating remains the same for each bandwidth taken separately. It is also noticed that as the bandwidth increases, the performance of each mode becomes more and more alike.

Finally, the MCA confirms what was expected concerning the effects of the bandwidth on the measure of error: the amount of error decreases as the bandwidth gets larger. In addition, one may notice from Table 5.4 that the average error for the largest bandwidth is 2.8 cm, which is an enormous reduction from the 52.5 cm average for the smallest bandwidth. Thus, the bandwidth factor loses its significance, both with respect to accuracy and

speed (as was indicated before), as the bandwidth increases.

Note that the experiment was conducted over a period of 10 days, with a maximum of 2 subjects per day: one in the morning and one in the afternoon. The effects of each of these 2 variants, that is day and time of the day, were also analyzed through ANOVA's. The results indicated that the day did not cause any significant variations on either measures: levels of 0.146 and 0.133 were obtained with respect to the elapsed time measure and the error measure respectively. A similar conclusion was derived for the time of the day with respect to the elapsed time measure ($P = 0.131$). However, it was found that the effects of the time of the day with respect to the error measure were significant ($P = 0.032$): specifically, the subjects were more accurate in the morning than in the afternoon.

In the final analysis, the time of the day was ignored (in spite of the fact that it was a significant factor), and only the camera mode and the path bandwidth were considered. There was no need to include the time of the day as another factor since the way it influenced the measures was already known and was not of any particular interest. By so doing, "noise" was introduced in the analysis which caused an underestimation of the significance of the two factors that were being examined: nevertheless, the two factors were found to be extremely significant.

5.2 Experiment #2

Measures of elapsed time taken during the experiment are compiled in Table 5.8. A one-way fixed-effect ANOVA with Multiple Classification Analysis was performed on the data to examine the effects of the camera mode on the elapsed time measures. The results are summarized in Table 5.9, and the complete results of the analysis are found in Appendix B. A level of significance of 0.05 was selected (as in experiment #1).

DAY	T-1	T-2	T-3	T-4
2 0	159	064	089	114
2 1	162	075	126	058
1 2	136	050	111	061
2 2	215	176	157	178
1 3	157	100	094	097
2 3	200	124	136	109
1 4	347	154	131	127
1 5	232	140	165	108
1 6	159	064	110	084
2 6	263	147	155	175
2 7	147	116	098	092
1 1	171	121	117	106
1 8	240	202	148	124
2 8	224	088	130	167
1 9	183	103	125	106
2 9	118	097	113	135

DAY
- First digit ==> 1=Morning, 2=Afternoon
- Secong digit ==> day of the experiment (0-9)

T-1,T-2,T-3,T-4
- Elapsed time measures (seconds) for bandwidth of 2 cm, 4 cm, 8 cm and 16 cm respectively.

Table 5.8: Data of experiment #2

ONEWAY(ELAPSED TIME by CAMERA MODE)**

SOURCE OF VARIATION	F	P
• Between groups	13.760	<0.00005

MODE	MEAN(s)	STANDARD DEVIATION(s)
• Manual control	194.5625	57.7026
• Automatic tracking	113.8125	42.4299
• Voice-operated	125.3125	22.8756
• Fixed-camera-position	115.0625	35.7071
• TOTAL	137.1875	52.7684

COCHRANS TEST FOR HOMOGENEITY OF VARIANCES
• C = MAX.VAR./SUM(VAR.) = 0.4806 ==> P=0.017

Table 5.9: Results of ONEWAY(ELAPSED TIME by CAMERA MODE) of experiment #2

ANOVA is based on the assumption that the variances of the diverse groups are homogeneous. This assumption however was not verified for experiment #2, as indicated by the Cochrans C test for homogeneity of variances. A P value of 0.017 is less than the 0.05-limit of acceptability.

The conclusion of the ANOVA is that the camera mode is a highly significant factor on the measures of elapsed time: in fact, it shows a maximum degree of significance ($P < 0.00005$). Specifically, a significant difference exists between the manual control mode and the three other modes: however, the performance of the three other modes can not be distinguished at a significant level. Although the test for homogeneity of variances was negative, the result of the ANOVA is so strong as to be conclusive.

The manual control mode is found to be the slowest mode of all, followed by the voice-operated mode; then the fixed-camera-position mode and the automactic tracking mode are the fastest modes, the latter being slightly faster. The only difference between these results and the results obtained from experiment #1 is the reversed order between the automatic tracking mode and the fixed-camera-position mode: however, in both cases, their performances are comparable with each other.

The differences in variance between the camera modes reveal an important aspect. Compared to the other modes (especially the manual control mode), the voice-operated mode exhibits a more constant performance between the subjects. This indicates that the mode is less subject-dependent than the other modes and does not require any special skills from the user.

5.3 Rating Of The Camera Modes By The Subjects

In any man-machine system, the long-term performance of the system is strongly dependent on the user's attitude towards it. Thus, this study would not have been complete without considering the preference of the subjects over the different camera control modes. To this end, two questions were asked to the subjects after having performed the two experiments:

- Q.1 : Suppose that you had a job to do which was an equal mix of tracking and pick-up and drop tasks. Which camera mode would be your first, second, third and last choice?
- Q.2 : Suppose that your employer explained to you that he would provide the best equipment that he could afford, but what he could afford depended on what you were paid. To determine which equipment he should buy, he asked: "What is the minimum wage you would accept to do an equal mix of tracking and pick-up and drop tasks with each of the four modes?".

The second question uses hourly wages as a means to compare between the camera modes. This was preferred over the standard 0-10 rating because wages have a real meaning which is common to each subject as opposed to numbers from 0 to 10. The answers of the subjects are gathered in Table 5.10.

SUBJECT NUMBER	CHOICE 1st	2nd	3rd	4th	WAGE($/hr) a	b	c	d
1	b	c	d	a	20	12	15	18
2	d	b	c	a	11	8	9	8
3	b	c	d	a	12	9.5	10	11
4	b	c	d	a	16	12	12	14
5	b	c	d	a	20	16	17	18
6	b	d	c	a	22	15	18	17
7	c	b	d	a	20	14	12	16
8	b	c	d	a	16	10	11	14
9	b	c	d	a	20	10	11	14
10	b	c	d	a	40	20	30	35
11	b	c	d	a	100	40	50	60
12	d	b	c	a	20	15	18	16
13	b	c	d	a	15	10	10	15
14	c	b	d	a	10	7	6	10
15	b	c	d	a	9	5	7	8
16	d	b	a	c	65	62	70	60

MODE

- a ==> manual control mode
- b ==> automatic tracking mode
- c ==> voice-operated mode
- d ==> fixed-camera-position mode

Table 5.10: Rating of the camera control modes by the subjects

One-way fixed-effect ANOVA's were conducted to analyze the effects of the camera mode on the two measures, that is the choice and the wages. Note that the wages were normalized for the analysis in such a way that the maximum wage for each subject was set to $10 per hour. Tables 5.11 and 5.12 summarize the results of the ONEWAY's with respect to CHOICE and WAGE respectively.

ONEWAY(CHOICE by CAMERA MODE)**

SOURCE OF VARIATION	F	P
• Between groups	50.330	<0.00005

MODE	MEAN	STANDARD DEVIATION
• Manual control	3.9375	0.2500
• Automatic tracking	1.3125	0.4787
• Voice-operated	2.1875	0.7500
• Fixed-camera-position	2.5625	0.8139
• TOTAL	2.5000	1.1269

COCHRANS TEST FOR HOMOGENEITY OF VARIANCES
• C = MAX.VAR./SUM(VAR.) = 0.4368 ==> P=0.059

Table 5.11: Results of ONEWAY(CHOICE by MODE)

ONEWAY(WAGE by CAMERA MODE)**

SOURCE OF VARIATION	F	P
• Between groups	28.558	<0.00005

MODE	MEAN($/hr)	STANDARD DEVIATION($/hr)
• Manual control	9.9556	0.1775
• Automatic tracking	6.6469	1.2822
• Voice-operated	7.4075	1.3386
• Fixed-camera-position	8.4300	1.0590
• TOTAL	8.1100	1.6287

COCHRANS TEST FOR HOMOGENEITY OF VARIANCES
• C = MAX.VAR./SUM(VAR.) = 0.3905 ==> P = 0.185

Table 5.12: Results of ONEWAY(WAGE by MODE)

A distinctive preference between the camera modes was confirmed by both analyses, with maximum levels of significance ($P < 0.00005$). The order of preference, starting with the one that is most preferred, is as follows: automatic tracking mode, voice-operated mode, fixed-camera-position mode and manual control mode.

As the subjects answered the questions, remarks concerning the different modes were gathered. These are also important to consider when designing a proper man-machine interface. Following is a summary of these remarks:

- All subjects but one selected the manual control mode as their last choice. As the subjects could not directly access the control box with their left hands, this mode was found to be very tedious and awkward to use, increasing significantly the workload. The performance of the subject who selected the voice-operated mode as his last choice was slightly better in the mamual control mode than in the voice-operated mode.
- The close-up lens was in general preferred over the wide angle lens: the subjects felt more comfortable at executing a task with a closer view of the arm. However, some subjects preferred the wide angle lens because they found it easier to move the slave arm with an overall view of the environment. Yet, all agreed that a zoom lens would be very useful and would help in the execution of tasks requiring both gross motions of the arm and delicate

operations.

- Automatic tracking mode was preferred (in general) over voice-operated mode for the reason that one is freed from controlling the pan-tilt unit in the automatic tracking mode. However, it was remarked that the motion of the camera would have to be smoother in the automatic tracking mode in order for this mode to be fully appreciated: the drawback of jerky motions of the camera would become more and more burdensome as time went by.
- All subjects but one were satisfied with using speech recognition as a means to control the video system. The one subject who selected the voice-operated mode as his last choice felt too much restrained in his speech, which explains his relatively poorer performance in that mode.

VI. CONCLUSIONS

A microprocessor-based system has been designed to control the video-camera of a teleoperator system. Through the use of a speaker-dependent isolated-word recognizer, the system enables an operator to remotely control the video-camera using spoken commands. In addition, the operator can have the video-camera automatically track the end effector of the slave arm.

Four different modes of operating the video-camera have been evaluated through two experiments. The first experiment consisted of a tracking task with four different path bandwidths: measures of elapsed time and error were taken and analyzed. In the second experiment, a pick-and-drop task was performed by subjects, and measures of elapsed time were also taken and analyzed. The situation considered in the experiments was one where the operator does not have a free hand for the control of the video-camera.

Results of experiment #1 have shown the significant effects of the camera mode and the path bandwidth on the measures: maximum levels of significance ($P < 0.0005$, ANOVA) were observed. In terms of speed, the manual control mode was found to be the slowest mode; then the voice-operated mode came second; and finally, the automatic tracking mode and the fixed-camera-position mode were the fastest modes, the latter being slightly faster. In terms of accuracy, the

fixed-camera-position mode was the least precise, followed by the automatic tracking mode and the manual control mode (the latter being slightly more accurate), and voice-operated mode was by far the most accurate mode.

The effects of the path bandwidth were as expected: as the bandwidth increases, the elapsed time and error decrease. In other words, the more precision required by a task, the longer it takes to accomplish the task and the more likely errors can be made. It was also noticed that the bandwidth factor loses its significance in terms of both accuracy and speed as the bandwidth increases.

The camera mode has also been found to be a significant factor in experiment #2: a maximum degree of significance ($P < 0.00005$, ONEWAY) was observed. The manual control mode was the slowest mode, followed by the voice-operated mode; then the fixed-camera-position mode and the automatic tracking mode were the fastest modes, the latter being slightly faster.

The differences in variance between the camera modes have revealed an important aspect. Compared to the other modes, the voice-operated mode exhibited a more constant performance between the subjects. This indicates that the mode is less subject-dependent than the other modes and does not rely on any special skills from the users.

Combining the results of the two experiments, the following conclusions can be drawn concerning the camera control modes:

- The manual control mode , although being the slowest mode of all, is not the most accurate as one would expect from a normal speed-accuracy tradeoff. In fact, the accuracy is quite inferior to that of the voice-operated mode. Thus, this mode does not present any advantages when the user does not have direct access to the manual control. However, in situations where the operator has a free hand for the control of the camera, the utilization of this mode would most likely be justified by an excellent performance.

- The fixed-camera-position mode is by far the least precise mode. It performs particularly well for large bandwidths, providing high speed and good accuracy (this conclusion is drawn from inspection of the data); however, for small bandwidths, the high speed of this mode is accompanied with a very poor accuracy.

- In terms of speed, the performance of the automatic tracking mode compares with the fixed-camera-position mode: in fact, these two modes are the fastest modes. However, the automatic tracking mode exhibits a much greater accuracy. Thus, for comparable speed, much accuracy is gained by choosing the automatic tracking mode over the fixed-camera-position mode.

- In terms of speed, the performance of the voice-operated mode is closer to that of the automatic tracking mode (and the fixed-camera-position mode) than to that of the manual control mode. In terms of accuracy, the voice-operated mode was far superior than any of the other modes.

Therefore, automatic tracking mode and voice-operated mode offer the best compromises between speed and accuracy. If speed is more important than accuracy, automatic tracking mode should be selected; on the other hand, if emphasis must be placed on accuracy, voice-operated mode should be preferred. In addition, these two modes were the two most preferred modes by the subjects.

In the voice-operated mode, the "memorization" feature of the control system was utilized, that is the experiments allowed for camera positions to be memorized by the system. Thus, the camera could be moved to a memorized position by uttering "GO TO" followed by the position number. Under this condition, the voice-operated mode performed very well, as indicated by this research. However, in situations where the "memorization" feature cannot be used, such as in field searching and following an unknown path, the performance of the voice-operated mode will be poorer for the following reasons:

- The current vocabulary and software limit movement to X or Y directions, one at a time.
- The necessity of pauses between commands means that quick

changes in direction cannot be executed.

Further studies

While furnishing useful data, this research has provided excellent groundwork for further studies. The conclusions drawn from this work encourage deeper investigation into the area of voice-operated and computer-aided control of a video-camera in remote applications.

Many facets of the video-camera control system have been purposely ignored in the experiments in order to limit the scope of this research: e.g. variable WINDOW size in the automatic tracking mode, the ability to control a motorized zoom lens, and discrete and contimuous motions. Experiments to determine the best size for the WINDOW would provide some important results. A study of the focal length (in terms of speed and accuracy in the voice-operated mode) could be carried out. Also, the automatic tracking of the arm by the camera should be looked at to render smoother motion. Finally, the interaction between the voice-operated mode and the automatic mode should be carefully studied in order to take full advantage of both modes, so as to optimize the use of the zoom lens.

Gradually, through all these experiments and studies, a better man-machine interface between the operator and the video-camera can be designed, optimizing the capabilities of man and machine.

REFERENCES

1. A.K. Bejczy, "Remote Applications of Robots", Proceedings of the 1983 International Conference on Advanced Robotics, 1983, Vol. 2, pp. 1-11.

2. A.K. Bejczy, "Sensors, Controls and Man-Machine Interface for Advanced Teleoperation", Science, No 4450, Vol 208, 20 June 1980, pp. 1327-1335.

3. E. Heer & A.K. Bejczy, "Control of Robot Manipulators for Handling and Assembly in Space", Mechanism and Machine Theory (GB), Vol. 18, No 1, pp 23-35, 1983.

4. A.K. Bejczy, R.S. Dotson, J.W. Brown & J.L. Lewis, "Voice Control of the Space Shuttle Video System", Proceedings of the 17th Annual Conference on Manual Control (June 16-18), UCLA, Los Angeles, CA, 15 Oct 81, pp. 627-640.

5. A.K. Bejczy, R.S. Dotson & F.P. Mathur, "Man-Machine Speech Interaction in a Teleoperator Environment", Proceedings of Symposium on Voice Interactive Systems, DOD Human Factors Group, Dallas, TX, May 13-15, 1980.

6. G. Streiff, P. Auchapt & J. Vertut, "Association of remote dexterity and remote lifting for maintenance in fuel reprocessing industry", Proceedings of the 1984 National Topical Meeting on Robotics and Remote Handling in Hostile Environments, Gatlinburg, USA, 1984.

7. J. Vertut, "Advances in Computer Aided Teleoperation Systems (CATS) in the frame of the French Advanced Robotics and Automation (ARA) Project", 1983 ICAR, Tokyo, Japan.

8. P. Marchal, J.L. Rouyer & J. Vertut, "Computer Aided Teleoperation applications in offshore operations", Journees Mediteraneennes de l'Offshore, Marseille, May 17-18, 1985.

9. J. Vertut, R. Fournier, B. Espiau & G. Andre, "Advances in a Computed Aided Bilateral Manipulator System", Proceedings of the 1984 National Topical Meeting on Robotics and Remote Handling in Hostile Environments, pp. 367-374.

10. D.C. Smith, "Summary of the Advanced Teleoperator Technology Conference", Naval Ocean Systems Center, AD-A085-187, San Diego, CA, April 80.

11. G.P. Starr, "Supervisory control of remote manipulation: a prelimanary evaluation", Proceedings of the 17th Annual Conference on Manual Control (June 16-18), UCLA, Los Angeles, CA, 15 Oct 81, pp. 95-107.

12. N. Shields Jr., F. Piccione, M. Kirkpatrick III & T.B. Malone, "Human operator performance of remotely controlled tasks: A summary of Teleoperator Research conducted at NASA'S George C. Marshall Space Flight Center between 1971 and 1981", Essex Corporation, Contract NAS8-31848, March 1982.

13. R. Nishijo, "Voice control of an unmanned submersible", Naval Ocean Systems Center, NOSC-TD-560, AD-A125-523, San Diego, 13 Jan 83.

14. R.S. Stoughton, H.L. Martin & R.R. Bentz, "Automatic Camera Tracking for Remote Manipulators", Proceedings of the 1984 National Topical Meeting on Robotics and Remote Handling in Hostile environments, pp. 383-389.

15. P.E. Van Hemel, S.B. Van Hemel & W.J. King, "Training implications of airborne applications of automated speech recognition technology", Technical report: NAVTRAEQUIPCEN 80-D-0009-0155-1, N61339, AD-A098-625, Oct 80.

16. T.G. McGinty & R.S. Shirley, "How to talk to your computer and enjoy it", Proceedings of the 17th Annual Conference on Manual Control, 15 Oct 81, pp. 691-702.

17. Voice Input Terminal SR-100 Instruction Manual, NEC Corporation.

18. R.P. Paul, Robot Manipulators: Mathematics, Programming and Control , Cambridge, Mass.: Mit Press, 1981.

19. R.E. Kirk, Introductory Statistics , Brooks/Cole Publishing Compagny, Monterey, CA.

20. W. Lee, Experimental Design and Analysis , W.H. Freeman and Co., San Francisco, c1975.

21. C. Lai, UBC SPSS: Statistical Package for the Social Sciences Version 9.00 (Under MTS), Computing Centre, The University of British Columbia, June 1983.

22. B.W. Kernighan & D.M. Ritchie, The C Programming Language , Prentice-Hall, Englewood Cliffs, NJ, c1978.

APPENDIX A - RESULTS OF THE ANALYSES OF EXPERIMENT # 1

* * * * * * * * * * A N A L Y S I S O F V A R I A N C E * * * * * * * * * *

TIME
BY MODE
WIDTH

* *

SOURCE OF VARIATION	SUM OF SQUARES	DF	MEAN SQUARE	F	SIGNIF OF F
MAIN EFFECTS	648761.750	6	108126.938	105.320	0.000
MODE	443700.938	3	147900.313	144.061	0.000
WIDTH	205060.813	3	68353.563	66.579	0.000
2-WAY INTERACTIONS	8514.438	9	946.049	0.921	0.507
MODE WIDTH	8514.438	9	946.049	0.921	0.507
EXPLAINED	657276.188	15	43818.410	42.681	0.000
RESIDUAL	246396.938	240	1026.654		
TOTAL	903673.125	255	3543.816		

256 CASES WERE PROCESSED.
0 CASES (0.0 PCT) WERE MISSING.

* * * M U L T I P L E C L A S S I F I C A T I O N A N A L Y S I S * * *

TIME
BY MODE
WIDTH

* *

GRAND MEAN = 94.16

VARIABLE + CATEGORY	N	UNADJUSTED DEV'N	UNADJUSTED ETA	ADJUSTED FOR INDEPENDENTS DEV'N	ADJUSTED FOR INDEPENDENTS BETA	ADJUSTED FOR INDEPENDENTS + COVARIATES DEV'N	ADJUSTED FOR INDEPENDENTS + COVARIATES BETA
MODE							
1	64	65.31		65.31			
2	64	-30.32		-30.32			
3	64	6.34		6.34			
4	64	-41.33		-41.33			
			0.70		0.70		
WIDTH							
1	64	42.59		42.59			
2	64	7.64		7.64			
3	64	-19.18		-19.18			
4	64	-31.05		-31.05			
			0.48		0.48		
MULTIPLE R SQUARED					0.718		
MULTIPLE R					0.847		

* * * * * * * * * * A N A L Y S I S O F V A R I A N C E * * * * * * * * * *

ERROR
BY MODE
WIDTH

* *

SOURCE OF VARIATION	SUM OF SQUARES	DF	MEAN SQUARE	F	SIGNIF OF F
MAIN EFFECTS	120162.125	6	20027.020	41.909	0.000
MODE	26406.789	3	8802.262	18.420	0.000
WIDTH	93755.375	3	31251.789	65.398	0.000
2-WAY INTERACTIONS	17692.875	9	1965.875	4.114	0.000
MODE WIDTH	17692.875	9	1965.875	4.114	0.000
EXPLAINED	137855.000	15	9190.332	19.232	0.000
RESIDUAL	114688.563	240	477.869		
TOTAL	252543.563	255	990.367		

256 CASES WERE PROCESSED.
0 CASES (0.0 PCT) WERE MISSING.

* * * M U L T I P L E C L A S S I F I C A T I O N A N A L Y S I S * * *

ERROR
BY MODE
WIDTH

* *

GRAND MEAN = 21.75

VARIABLE + CATEGORY	N	UNADJUSTED DEV'N	UNADJUSTED ETA	ADJUSTED FOR INDEPENDENTS DEV'N	ADJUSTED FOR INDEPENDENTS BETA	ADJUSTED FOR INDEPENDENTS + COVARIATES DEV'N	ADJUSTED FOR INDEPENDENTS + COVARIATES BETA
MODE							
1	64	-4.51		-4.51			
2	64	-2.42		-2.42			
3	64	-10.00		-10.00			
4	64	16.93		16.93			
			0.32		0.32		
WIDTH							
1	64	30.72		30.72			
2	64	0.94		0.94			
3	64	-12.76		-12.76			
4	64	-18.90		-18.90			
			0.61		0.61		
MULTIPLE R SQUARED					0.476		
MULTIPLE R					0.690		

APPENDIX B - RESULTS OF THE ANALYSES OF EXPERIMENT # 2

- O N E W A Y -

VARIABLE TIME
BY VARIABLE MODE

ANALYSIS OF VARIANCE

SOURCE	D.F.	SUM OF SQUARES	MEAN SQUARES	F RATIO	F PROB.
BETWEEN GROUPS	3	71501.0000	23833.6641	13.760	0.0000
WITHIN GROUPS	60	103922.6836	1732.0447		
TOTAL	63	175423.6250			

GROUP	COUNT	MEAN	STANDARD DEVIATION	STANDARD ERROR	MINIMUM	MAXIMUM	95 PCT CONF INT FOR MEAN
GRP1	16	194.5625	57.7026	14.4257	118.0000	347.0000	163.8150 TO 225.3100
GRP2	16	113.8125	42.4299	10.6075	50.0000	202.0000	91.2032 TO 136.4218
GRP3	16	125.3125	22.8756	5.7189	89.0000	165.0000	113.1230 TO 137.5020
GRP4	16	115.0625	35.7071	8.9268	58.0000	178.0000	96.0356 TO 134.0894
TOTAL	64	137.1875	52.7684	6.5960	50.0000	347.0000	124.0064 TO 150.3686
	FIXED EFFECTS MODEL		41.6178	5.2022			126.7815 TO 147.5935
	RANDOM EFFECTS MODEL			19.2977			75.7746 TO 198.6004

RANDOM EFFECTS MODEL - ESTIMATE OF BETWEEN COMPONENT VARIANCE 1381.3511

TESTS FOR HOMOGENEITY OF VARIANCES

COCHRANS C = MAX. VARIANCE/SUM(VARIANCES) = 0.4806, P = 0.017 (APPROX.)
BARTLETT-BOX F = 3.951, P = 0.008
MAXIMUM VARIANCE / MINIMUM VARIANCE = 6.363

- O N E W A Y -

VARIABLE TIME
BY VARIABLE MODE

MULTIPLE RANGE TEST

LSD PROCEDURE
RANGES FOR THE 0.050 LEVEL -

2.83 2.83 2.83

THE RANGES ABOVE ARE TABLE RANGES. THE VALUE ACTUALLY COMPARED WITH MEAN(J)-MEAN(I) IS..
29.4283 * RANGE * SQRT(1/N(I) + 1/N(J))

HOMOGENEOUS SUBSETS (SUBSETS OF GROUPS, WHOSE HIGHEST AND LOWEST MEANS DO NOT DIFFER BY MORE THAN THE SHORTEST SIGNIFICANT RANGE FOR A SUBSET OF THAT SIZE)

SUBSET 1

GROUP	GRP2	GRP4	GRP3
MEAN	113.8125	115.0625	125.3125

SUBSET 2

GROUP	GRP1
MEAN	194.5625