General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)

N70-27231
(ACCESSION NUMBER)
194
(PAGES)
CR-109769
(NASA CR OR TMX OR AD NUMBER)
(THRU)
1
(CODE)
05
(CATEGORY)
FACILITY FORM 602

RECEIVED
MAY 1970
NASA STI FACILITY
INPUT BRANCH

DIGITAL SYSTEMS LABORATORY

SCHOOL OF ENGINEERING

CASE WESTERN RESERVE UNIVERSITY

THIS RESEARCH WAS SPONSORED BY THE SPACE NUCLEAR PROPULSION OFFICE-NASA/AEC UNDER RESEARCH GRANT NGR 36-003-042

✓NG-R-36-003-042

A COMPUTER AIDED TELEOPERATOR SYSTEM

✓Final Report
1 - 70 - 80

Investigators
Peter Hammon
Richard Taylor
Jon Beckett
Norman Diederich

Principle Investigator
Professor H. W. Mergler

February 1970

A COMPUTER AIDED TELEOPERATOR SYSTEM

by

NORMAN F. DIEDERICH

Thesis Advisor: Professor H. W. Mergler

School of Engineering

Department of Computing and Information Sciences

CASE WESTERN RESERVE UNIVERSITY

June 1970

A COMPUTER AIDED TELEOPERATOR SYSTEM

Abstract

by

NORMAN F. DIEDERICH

An experimental Teleoperator System is presented which utilizes a general purpose digital computer to assist the operator in performing various tasks. A specific task was selected to test the system. This task involves the use of the teleoperator in the disassembly of a model of a nuclear reactor.

Several pieces of equipment were constructed to permit the operator to communicate with the teleoperator system through the computer. These devices are described and their use is detailed.

The Computer Aided Teleoperator System utilizes a series of algorithms to perform the necessary control functions. These algorithms along with other special functions can be initiated by the operator from the teletype control console. This function is permitted by a teletype executive control routine which handles this selection. By means of this teletype executive routine the operator can indicate the mode of operation desired including manual, semi-automatic, and fully automatic under the control of another executive routine which is controlled by paper tape.

Communication paths have been provided to permit the status of various elements of the manipulator (the prime mover portion of

the teleoperator system) to be communicated to the computer. These statuses are evaluated and under certain conditions special action is taken.

PRECEDING PAGE BLANK NOT FILMED.

TABLE OF CONTENTS

ABSTRACT		ii
ACKNOWLEDGEMENT		iv
TABLE OF CONTENTS		v
LIST OF FIGURES		vii
LIST OF TABLES		
CHAPTER I	INTRODUCTION	1
	Footnotes	6
CHAPTER II	THE EXPERIMENTAL SYSTEM	7
	2.1 Introduction	7
	2.2 Mode Definitions	13
	2.3 The Manipulator	14
	2.4 Analog Control Section	15
	2.4.1 Silicon Controlled Rectifier (SCR) Amplifier	15
	2.4.2 D/A Converters	18
	2.4.3 Amplifiers and Filters	21
	2.4.4 Null Detector	21
	2.5 The Computer	22
	2.6 The Discrete General Purpose Interface Unit (GPI)	22
	2.7 Integrated Circuit General Purpose Interface (ICGPI)	23
	2.8 Manipulator Control Logic	24
	2.8.1 Discrete Control Logic	24
	2.8.2	
	2.9 Programming Notes	42
	2.9.1 GPI and MCI	43
	2.9.2 ICGPI and AMCI	44
	Footnotes	48
CHAPTER III	TELEOPERATOR CONTROL ALGORITHMS	49
	3.1 Introduction	49
	3.2 Executive Control Programs	50
	3.2.1 Teletype Executive Control Program	50
	3.2.2 The Automatic Mode Executive Routine and Punch Routine	52
	3.3 Control Algorithms	55
	3.3.1 Manipulator Position Control	55

3.3.2 Hand Grip and Rotate Control 59
3.3.3 Path Control -- Straight Line 62
3.3.4 Hand Position Control 64
3.3.5 Special Restrictions and Override 98
3.4 Automatic Mode 104
3.4.1 Record Data -- Punch Paper 105
3.4.2 AUTO Exec -- Paper Tape Read Routine 108
3.4.3 Coordinate Transformation 108
3.5 Software A/D Conversion 118
3.6 Miscellaneous Routines 120
3.6.1 Position and Status Typeout 120
3.6.2 Adjust Effective Hand Length 120
3.6.3 Tool Changing 122
3.6.4 Control Special Modes 122
Footnotes 123

CHAPTER IV SUMMARY AND SUGGESTED FUTURE EXTENSIONS 125
4.1 Introduction 125
4.2 Achievements 125
4.3 Recommendations 126
4.4 Summary and Conclusions 127

BIBLIOGRAPHY 129

APPENDIX I MANIPULATOR COORDINATE SYSTEM, SCALING OF MANIPULATOR VARIABLES AND QUANTIZATION 131

APPENDIX II TELETYPE EXECUTIVE CONTROL ROUTINE 135

APPENDIX III SUBROUTINE LIBRARY FOR EXPERIMENTAL COMPUTER AIDED TELEOPERATOR SYSTEM 143

APPENDIX IV ERROR MESSAGES FROM PROGRAMS IN EXPERIMENTAL COMPUTER AIDED TELEOPERATOR SYSTEM 150

APPENDIX V TRACE FLAGS FOR DEBUGGING SOFTWARE AND HARDWARE 153

APPENDIX VI SAMPLE PROGRAM FOR AUTOMATIC MODE: REMOVAL OF UPPER ENDBELL AND CORE FROM REACTOR MODEL 155

APPENDIX VII DISCRETE GENERAL PURPOSE INTERFACE 162

APPENDIX VIII INTEGRATED CIRCUIT GENERAL PURPOSE INTERFACE 171

APPENDIX IX INSTRUCTION AND DATA FORMATS FOR THE AUXILIARY MANIPULATOR CONTROL INTERFACE AND THE MANIPULATOR CONTROL INTERFACE 178

Figure Number	List of Figures	Pages
2-1	Experimental System as Viewed from Computer Room	8
2-2	Experimental System as Viewed from Laboratory	9
2-3	The Case Manipulator	11
2-4	Experimental System Block Diagram	12
2-5	Manipulator Arm Dimensions	16
2-6	D/A Converter and Bridge Netword	19
2-7	Simplified Bridge Network	19
2-8	D/A System for HR & HG Axes	20
2-9	Typical Information Flow--Manipulator to Computer	29
2-10	Typical flow of Information from Computer to Manipulator Logic	30
2-11	Auxiliary Manipulator Control Interface	32
2-12	Maskable Error Registers and Interrupt Structure	33
2-13	Manual ERROR Sense Amplifier	35
2-14	Flow Chart for Data Transfers in FORCE Mode	38
2-15	HG & HR Axis Block Diagram	40
2-17	Two Programming Hazards	45
3-1	TTY Exec Options	53
3-2	AUTO Exec Options	56
3-3	ITMNIP Flow Diagram	58
3-4	IGRIP Flow Diagram	60
3-5	GRIP Flow Diagram	61

3-6	Vector and Hand Orientation Relationships	63
3-7	EDIST Flow Diagram	65
3-8	VECTOR Flow Diagram	66
3-9	HOPOUT, PHCK, AND FLAGGT Flow Diagram	68
3-10	RTHAND Flow Diagram	69
3-11	ITHAND Flow Diagram	70
3-12	Manipulator and Hand Variable Relationships	71
3-13	Possible Manipulator Configurations for a Specified Change in the Hand Position	73
3-14	Example of SP, EP Grid and the Grid Time Table	76
3-15	Examples of SP, EP, and WP Constraints	78
3-16	Optimized Manipulator Configurations for HX Increments	79
3-17	Paths of Manipulator for Examples in Figure 3-16	80
3-18	Optimized Manipulator Configurations for HZ Increments	81
3-19	Paths of Manipulator for Examples in Figure 3-18	82
3-20	GRID Flow Diagram	84
3-21	Flow Diagrams of Subroutines Called by GRID	85
3-22	Example of a Possible collision between the Manipulator and An Obstacle	86
3-23	Typical Obstacle Bounds	89
3-24	IBOUND Flow Diagram	91
3-25	DTCT Flow Diagram	92
3-26	RDCT Flow Diagram	93

3-27	Obstacle Evasion Paths Tested by SUBG	95
3-28	SUBG Flow Diagram	96
3-29	Photographs of Manipulator Paths Generated to Avoid Obstacles	97
3-30	Values of HX, HY, and HZ for Directing the Manipulator Over or Around an Obstacle	98
3-31	ICONTA Flow Diagram	99
3-32	Contab Flow Diagram	101
3-33	Sample of Paper Tape Block Structure	106
3-34	PUNCH Flow Diagram	107
3-35	Flow Diagram for AUTOEX	109
3-36	Transformation of WP and SR	113
3-37	IREF Flow Diagram	114
3-38	RELHAND Flow Diagram	115
3-39	Flow Diagram For Relative Hand Move	116
3-40	SAD Flow Diagram	119
3-41	Flow Diagram For FSAD	121
A-II.1	TTY Exec Flow Diagram	141
A-VII.1	GPI Block Diagram	165
A-VII.2	Input Line Termination	166
A-VII.3	DIPXX-, SRDYX-, SHIFT- Requirements	166
A-VII.4	OTDNN-, IFRDY-, IFDMC-, IFINX+ Characteristics	167
A-VII.5	OCPNN- Characteristics	167
A-VIII.1	Typical Input/Output Line	176
A-IX.1	Auxiliary Manipulator Control Interface-Integrated Circuit General Purpose Interface Formats	181

A-IX.2 Manipulator Control Interface-GPI Buffer Data Formats

LIST OF TABLES

		Page
2-1	Manipulator Specifications	17
A-I.1	Quanta For Manipulator Axes	134
A-II.1	Teletype Commands	135
A-III.1	Routine Summary	143
A-IV.1	Program Error Messages	150
A-V.1	Trace Flags and Meaning	153
A-VI.1	Preprogrammed Operations for Endbell and Core Removal	157
A-VI.2	Operations Required by Operator To use Program of Table A-VI.1.	160
A-VII.1	Discrete General Purpose Interface Commands	163
A-VII.2	GPI I/O Connector G	168
A-VII.3	GPI I/O Connector H	169
A-VIII.1	Instructions for Integrated Circuit General Purpose Interface	172
A-VIII.2	I/O Connector Signal Pins	174
A-IX.1	OCP and SKS Instructions	178

CHAPTER I

INTRODUCTION

The Man-Machine Communications Project at Case Western Reserve University has as its aim the development of a system to utilize a computer in the control of a remote manipulator. Work on this project includes five aspects:

1. development of a servo-control drive system for the manipulator (6)
2. the development of a digital communication and control system between the computer and the manipulator servo-system (14)
3. the development of a set of algorithms utilizing the above hardware in the performance of various operation (1)
4. the development of a digital servo-control for the hand-rotation and hand-grasp of the manipulator (11)
5. the development of an application system to aid in the disassembly of a model of a nuclear reactor

It is the purpose of this report to summarize the work performed on the Man-Machine Communications Project (MMCP) and to document the development of the fifth aspect of this endeavor. Excerpts from previous reports of the MMCP will of necessity be included. Original refinements of the equipment and of the programming support will form the main body of this report.

The system developed in this project is termed a teleoperator. The term "teleoperator" encompasses "a general purpose, dexterous, cybernetic machine. ...The prefix 'tele' in teleoperator describes

the ability of this class of man-machine systems to project man's innate dexterity not only across distance, but (also) through physical barriers as well."[1]

The manipulator described in this report is a form of teleoperator since it is capable of performing numerous tasks, restricted in class by construction. The system is a true teleoperator because the human operator is required to specify the task and to modify the performance of the task.

Man is designing remote manipulators to perform tasks in a hazardous environment: the handling of dangerous or radioactive materials; and the researching of both suboceanic and extra-terrestrial environments. Much of the current development in manipulators can be related to attempts by personnel associated with the handling of radioactive materials in order to simplify or to enhance their ability to work with these materials. The mechanical design and construction of remote manipulators is dependent upon the specific application; however, the technique required for the mechanical design is well within the state of the art. Central to the MMCP is the control problem and the development of a technique which can be applied to future systems.

Currently manipulators form two groups:

1. Rate-Controlled Manipulators
2. Master-Slave Manipulators.*

Rate-Control Manipulators, while capable of handling heavier loads and working in larger areas than the Master-Slave manipulator, are generally slower and more difficult to control manually than are their counterparts due to the necessity of controlling the rate of each axis independently. This difficulty is particularly significant when specific paths not aligned with a manipulator axis must be traversed.

Master-Slave Manipulators make possible path control operations utilizing the operator's dexterity to control the path; however, the presence of force feedback can result in a high fatigue factor in the human operator. The fatigue factor is particularly high when the manipulator moves at a slow rate. In general, a large working area is required for the operator of the master unit.

*The rate-controlled technique utilizes a control unit which allows the rate of each axis of the manipulator to be independently controlled. The control unit generally contains one lever for each axis and the rate of motion of the axis is proportional to the displacement of the lever.

The master-slave type of manipulator requires a "master" unit which is controlled by the operator and which is similar in shape to the manipulator (termed the "slave"). The two units are mechanically, electrically, or hydraulically linked so that the slave follows the motions of the master. Generally this link is also used to provide force feedback in which load forces at the manipulator are reflected back to the master.[2]

This report presents a control technique in which a general purpose digital computer is utilized to assist the operator in the execution of complex remote manipulation tasks.[3]

The computer's high computation speed and ability to make decisions rapidly coupled with its highly flexible internal and input-output structure enables the following features:

1. A reduction of the amount of conscious attention required of the operator to control the path or position of the manipulator.

2. Enhancement of the performance of positioning operations by optimizing the terminal configurations of the manipulator.

3. Improvement of the speed and accuracy of path control operations.

4. Minimization of the amount of data the operator is required to specify in order to perform a task.

An additional feature is the ability to direct an individual axis manually while maintaining fully automatic control over the other axes.

Numerous small, high-powered computers available make their utilization both feasible and practical.

In order to study Man-Machine Communication it has been necessary to construct an experimental facility by which the various developments can be implemented. The equipment designed and constructed for this facility is detailed in Chapter II and Appendices I, VII, VIII, & IX.

Utilization of the hardware required the development of a system of control programs and routines implementing various forms

of control algorithms. These programs are detailed in Chapter III and Appendices II, III, IV, & V.

At this point it is important to emphasize the fact that even with an extensive programming library, the operator is required since it is impossible to take into account every condition for operation in advance. This system is therefore a genuine teleoperator system since the computer aids (by performing tedious operations) rather than replaces the human operator.

FOOTNOTES

CHAPTER I

1. Edwin G. Johnsen, and William R. Corliss, Teleoperators and Human Augmentation, (NASA SP-5047, Technology Utilization Division, NASA, December, 1967), p. 2.

2. Jon T. Beckett, A Computer Aided Control Technique For a Remote Manipulator, (Report 1-67-44, Digital System Laboratory, Case Institute of Technology), p. 2.

3. Ibid. p. 4.

4. Ibid.

CHAPTER II

THE EXPERIMENTAL SYSTEM

2.1 INTRODUCTION

To permit the evaluation of various control techniques for remote manipulation an experimental facility has been developed in which a Remote Manipulator is coupled to a small general purpose digital computer. Figures 2-1 and 2-2 illustrate the physical equipment utilized in this research. Contained in this chapter is a summary of the physical equipment utilized in this system. Futher details of importance to the user may be found in the Appendices. Hammond [6] details the analog control system utilized for the seven basic manipulator axes; the report by Taylor [14] contains a partial description of the manipulator digital control section in addition to the General Purpose Interface and the Manipulator itself; the report by Gillespie and LePlatney [11] details the electrical and mechanical modifications performed to provide closed loop control over two additional axes (Hand Rotation and Hand Grasp). This Chapter will also contain original work required to complete the control of the manipulator and to modify some of the work performed by Taylor.

The manipulator system utilized in this project is a General Mills* Model 100 manipulator which has been modified by the addition of an upper arm and a forearm to resemble a PaR* Model 3000 manipulator.

*Manufactured and distributed by Programmed and Remote Systems Corporation (PaR), ST. Paul, Minnesota



Figure 2-1. Experimental System as Viewed from Computer Room

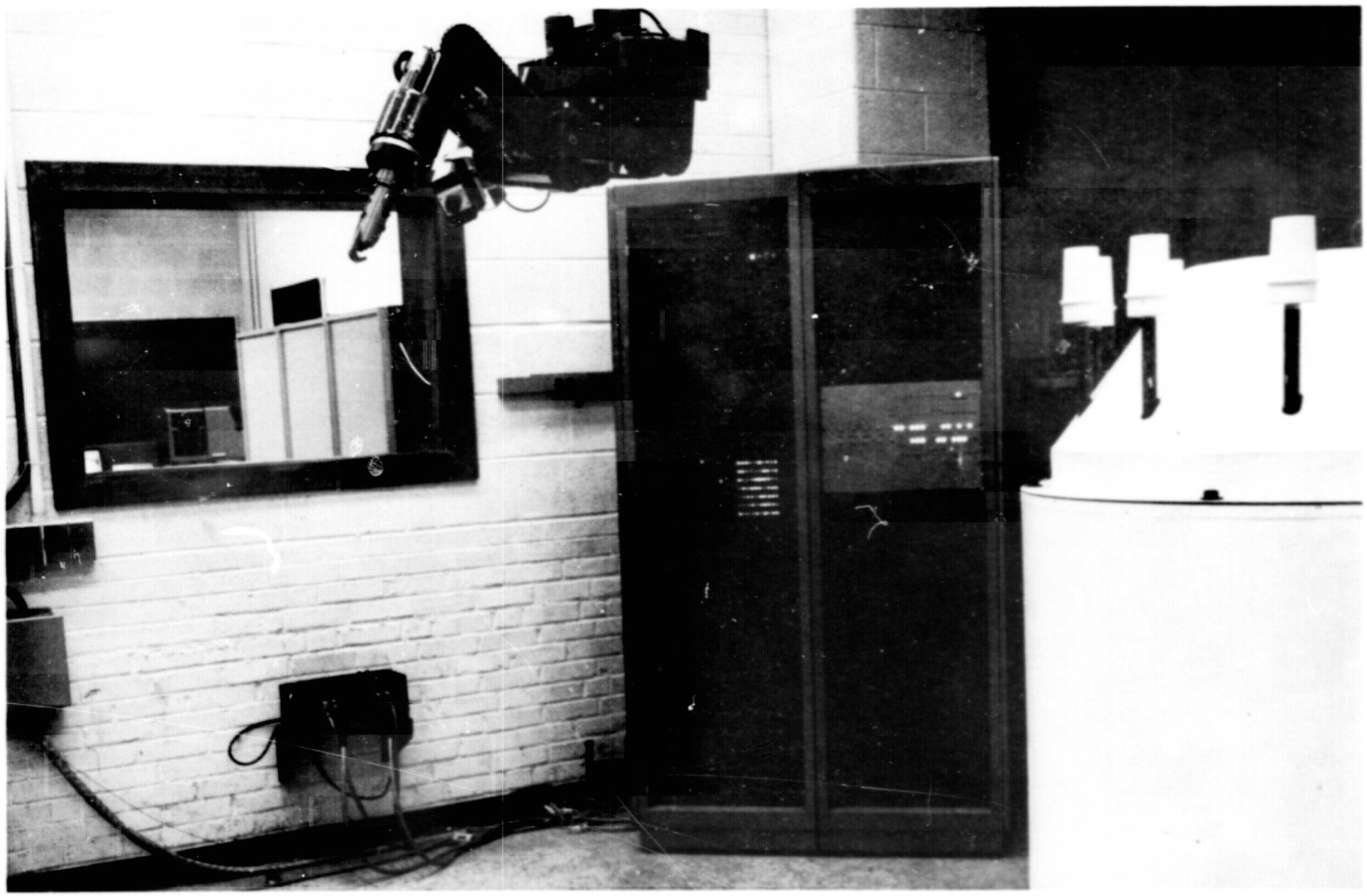


Figure 2-2. Experimental System as Viewed from Laboratory

This device is shown in Figure 2-3 and has nine degrees of freedom referred to as bridge traverse, carriage traverse, hoist, tube (or shoulder) rotation, shoulder pivot, elbow pivot, wrist pivot, hand rotation, and hand grasp. These have been abbreviated as X, Y, Z, SR, SP, EP, WP, HR, and HG for future references. The control for the first seven of these axes utilizes analog control loops while the last two utilize digital loops with analog drive sections.

Figure 2-4 gives a detailed block diagram showing all of the major components of the system. There are several modes of control. In the most basic mode information is transferred in sequence to the seven basic manipulator axis buffers from the computer through the General Purpose Interface (GPI) and through the Manipulator Control Interface (MCI). In an alternate mode the computer would select the desired axis to receive the data using the Integrated Circuit General Purpose Interface (ICGPI) and the Auxiliary Manipulator Control Interface (AMCI). The data would then be transferred to the manipulator axis buffers through the GPI in a special mode. The control information for the two hand axes is transmitted via the ICGPI to registers contained in the AMCI. For the first seven axes the digital position as contained in the axis buffer registers, is converted to an analog setpoint. This is then compared with the actual position as indicated by a potentiometer on the structure and the resulting difference or error is used as input to the SCR drivers for the motors. For the Hand Axes the position is maintained in a

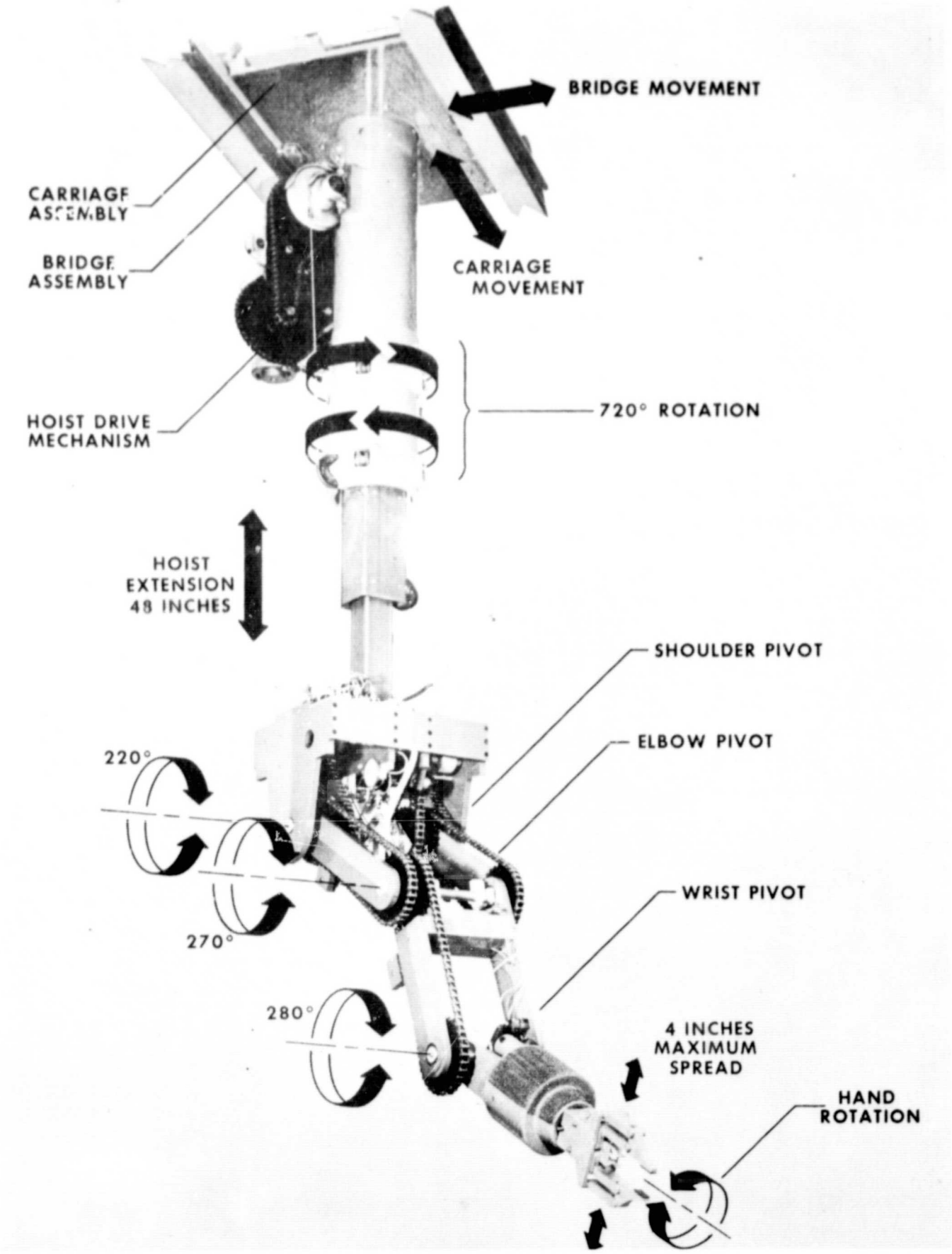


Figure 2-3. The Case Manipulator

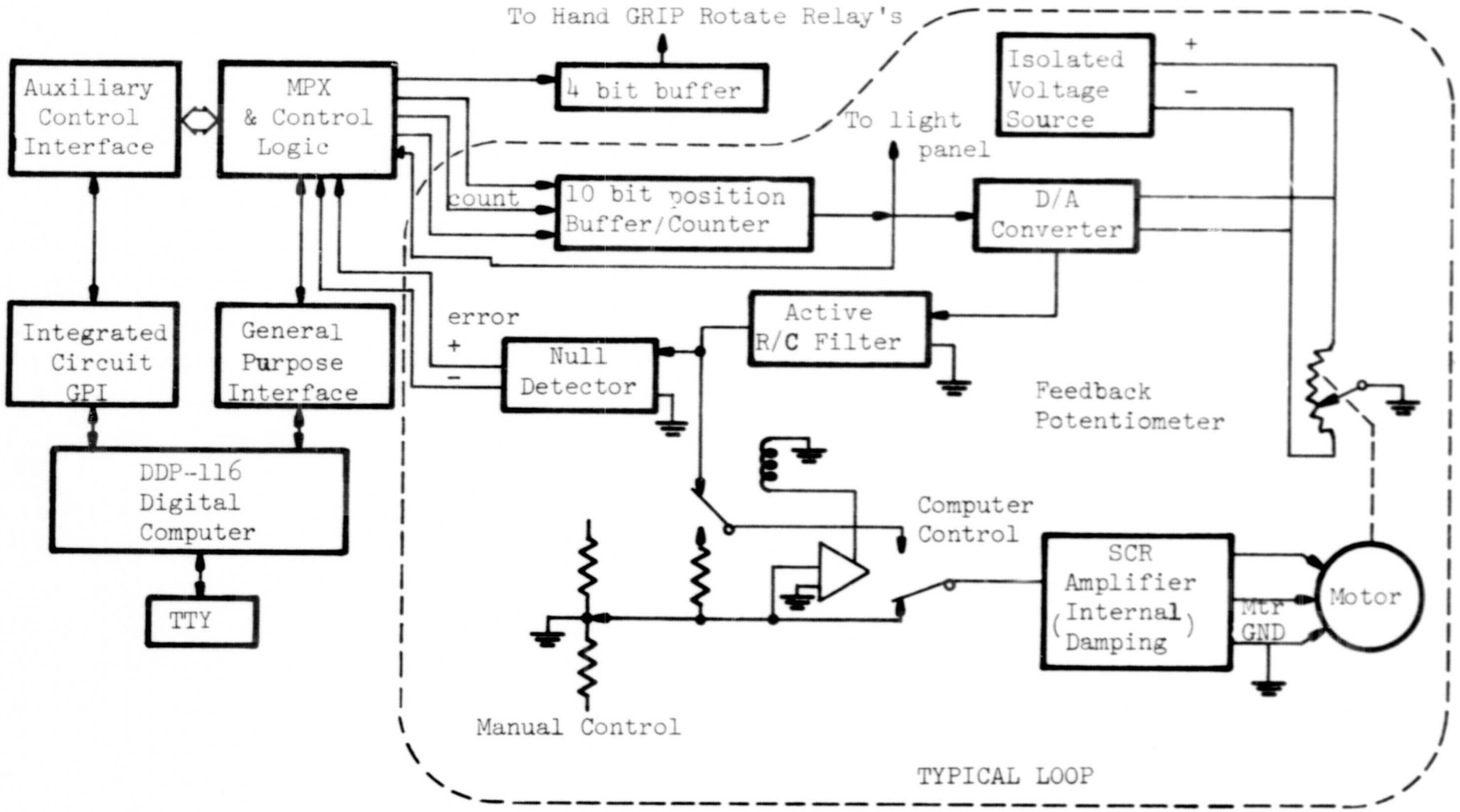


Figure 2-4 Experimental System Block Diagram

pair of tracking counters and buffers. This position is then utilized by the computer to determine the error relative to the next setpoint. This error is output to a second set of buffer counters and converted using seven bit D/A converters to provide an analog input to the same type of SCR drive as is utilized by the other seven axes. These counters are updated by digital quantizers which are geared to the actual drive motors.

2.2 Mode Definitions

Various "modes" referred to in this and other chapters can be defined as follows:

INPUT Mode (of GPI): The computer is permitted to input information from the GPI data buffer.

OUTPUT Mode (of GPI): The computer is permitted to output information to the GPI data buffer.

MANUAL Mode (of MCI): The manipulator can be controlled only by using the manual control panel.

SAD (software A/D): Clocks used to count manipulator buffers in tracking mode are inhibited to one or more of the manipulator axis buffers.

RATE-stimulated Mode: Rate control of the manipulator is stimulated by the computer. The moves for each axis are divided into the same number of equal increments and these are output to the manipulator buffers at a constant rate until the endpoint is reached.

FORCE Mode: Channel addressing for the manipulator logic is supplied from the computer through the ICGPI and AMCI, individual flip-flops control the SAD mode for each axis and certain signals through the GPI to the MCI are partially inhibited so as not to change the control mode.

CLOSED LOOP MODE (of AMCI): This forces the MCI to remain in automatic mode.

2.3 The Manipulator

To make the General Mills manipulator resemble the PaR Model 3000 as well as to permit computer control the following modifications have been made:

1. An upper arm and forearm have been added. These limbs are coupled to their drive motors by chains in such a manner that their absolute angles (measured relative to vertical) can be controlled independently. The position of the axis is transmitted back to the shoulder independent of the drive chains by utilizing PIC* belts and idlers so as to negate the possible effects of loads on this portion of the system.

2. The original relay controls for the X, Y, Z, SP, EP, WP, and SR motors were discarded and a servo control system utilizing an SCR power amplifier was designed to drive the motors.

3. The printed circuit switches utilized by the original relay control system were replaced by potentiometers which when coupled to the SCR drive system provide for proportional control of the seven major manipulator axes.

4. Potentiometers were mounted on the seven basic manipulator axes and are used to feed back the absolute position of these axes. In addition incremental optical quantizers were coupled to the Hand Rotation and Hand Grasp motors to provide position feedback for these two axes.

5. The repulsion motor which was used to provide the drive for the hoist axis was replaced by a series-wound universal motor to permit the use of identical drive systems for all axes.

6. A closed circuit TV camera was fitted at the shoulder of the manipulator structure and is controlled by two position printed circuit switches in the manual control box. This camera is intended to provide additional details for the operator.

---

*PIC Design Corporation, Subsidiary of Benrus Watch Company, Inc., East Rockaway, Long Island, New York.

The physical dimensions of the arm are shown in Figure 2-5 and a comparison between the PaR Model 3000 and the experimental manipulator is given in Table 2-1. Two limitations of the research manipulator should be mentioned at this point:

1. the design load limit at the end of the arm when extended is fifty pounds.

2. the design limit for a vertical lift when the arm is not extended is two hundred pounds.

2.4 Analog Control Section

2.4.1 Silicon Controlled Rectifier (SCR) Amplifier

The SCR amplifier was designed to drive a low-power series-wound universal motor, the speed and direction of which corresponds to the magnitude and sign of a DC signal. The signal controls the phase angle at which the SCR's fire. This in turn meters the amount of power supplied to the motor. In order to optimize the performance, the following features are incorporated in the amplifier:

1. Four SCR's are used in conjunction with a 220 V three wire AC source to drive the motors close to their rated value for maximum torque.

2. An adjustable "negative" dead-band has been added to compensate for the stiction of the motor. This is accomplished by adding a bias to the signal which produces enough power to almost overcome the stiction. The synchronization pulse (used to insure that all SCR's are turned off before the start of each cycle) is then extended to just delete the bias when the signal is zero. When the signal exceeds this small dead-band, the bias is added to the output signal.

3. The approximate EMF of the motor (produced by its rotation) is monitored at the amplifier and is used for first-derivative compensation to improve the response by reducing the over-shoot.

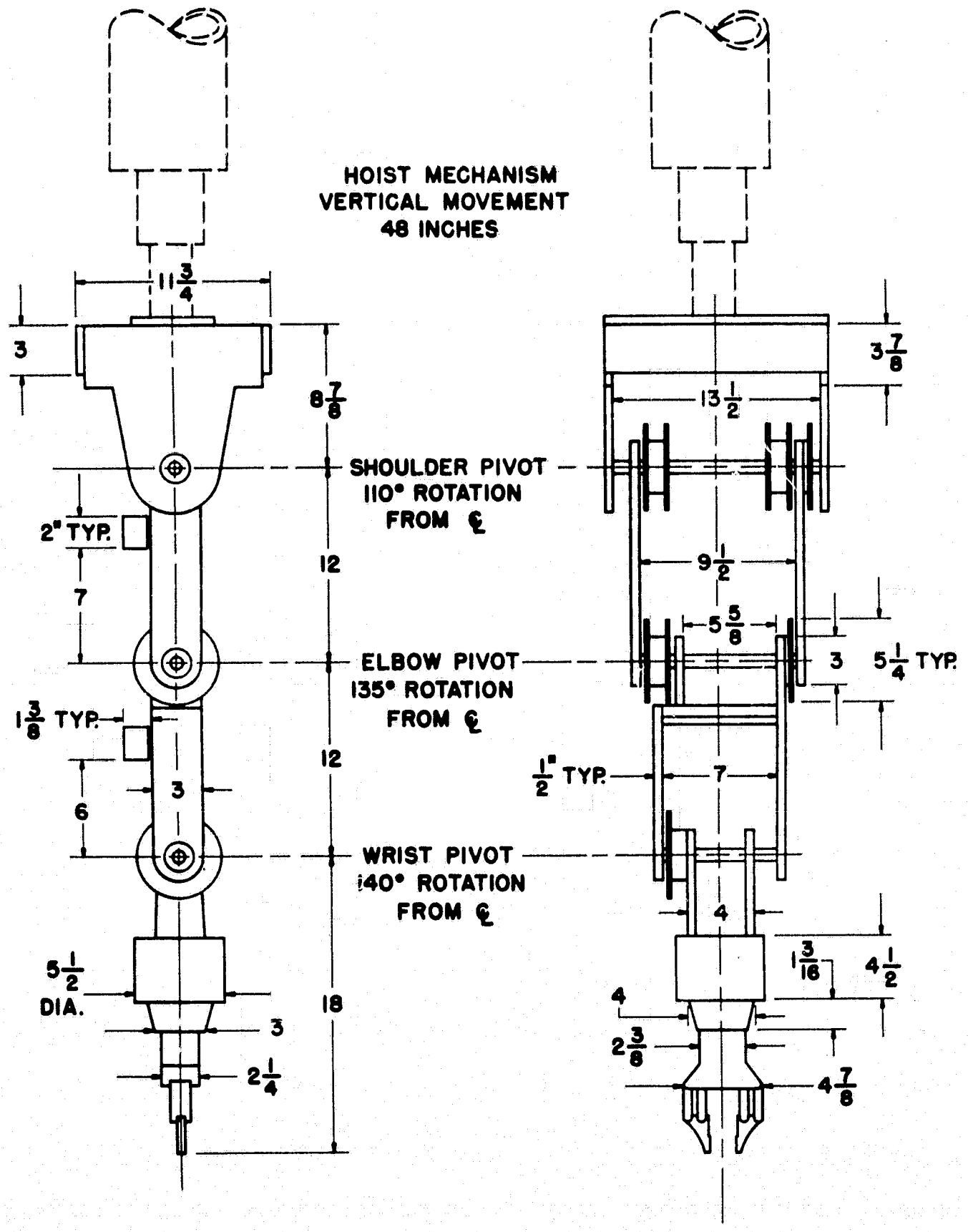


Figure 2-5. Manipulator Arm Dimensions

Table 2-1. Manipulator Specifications

Element	Case Manip.	PaR 3000
Bridge		
Travel	14.5 feet	These depend on the particular installation
Velocity	8 fpm	
Carriage		
Travel	12 feet	
Velocity	10 fpm	
Hoist		
Travel	4 feet	
Velocity	8 fpm	
Shoulder Rotation		
Travel	720°	Continuous
Velocity	5 rpm	1.5 rpm
Shoulder Pivot		
*Travel	±110°	±125°
Velocity	1.5 rpm	2.5 rpm
Elbow Pivot		
*Travel	±185°	±180°
Velocity	1.5 rpm	2.5 rpm
Wrist Pivot		
*Travel	±270°	±305°
Velocity	1.5 rpm	2.5 rpm
Hand Rotation		
Travel	Continuous	Continuous
Velocity	50 rpm	7 rpm
Grip		
Opening	4 in.	4 in.
Velocity	120 ipm	20 ipm

*Angles are measured from the center line.

4. By mounting two diodes near the motor, it is possible to run only two wires to each motor (plus a ground wire common to all motors). The diodes are connected so that the current through the field winding always flows in the same direction. The direction of current in the armature thus controls the direction of rotation of the motor.

5. Limit switches were added to the pivots and shoulder rotation. They were mounted in series with the motor wires such that one prevents rotation in one direction and the other prevents rotation in the opposite direction. It made it possible to prohibit motion beyond the limits while allowing motion out of the limits without running separate wires back to the control logic.

Note: A 15 amp circuit breaker is used to protect the SCR's; however, it is necessary to stagger the turn-off times of the motors because of the logic current surges which occur. In general, no more than one motor should be turned on or off each 17 m sec.[1]

2.4.2 D/A Converters

The seven axes controlled by the MCI each utilize a ten bit "voltage ladder" Digital to Analog converter. This converter is connected to the manipulator axis buffer register and is used in a bridge circuit with the position feedback potentiometer on each axis to determine the error between the position as indicated by the buffer and the actual position. This style of D/A is utilized with an isolated voltage source so as to provide a condition where no current flows at the null point since the voltage at the potentiometer wiper and the voltage at the output of the D/A are both ground (since the wiper is grounded). A simple diagram of this system is presented in Figure 2-7.

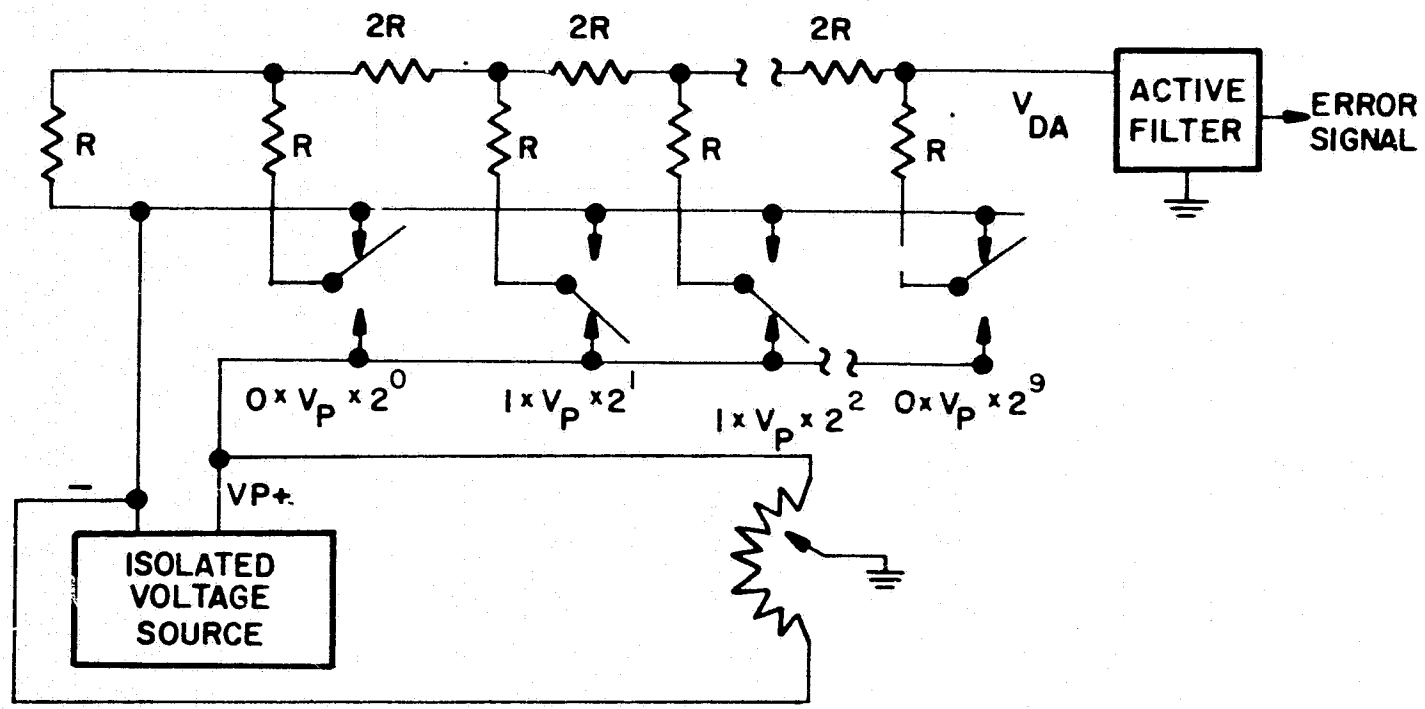


Figure 2-6. D/A Converter and Bridge Network [2]

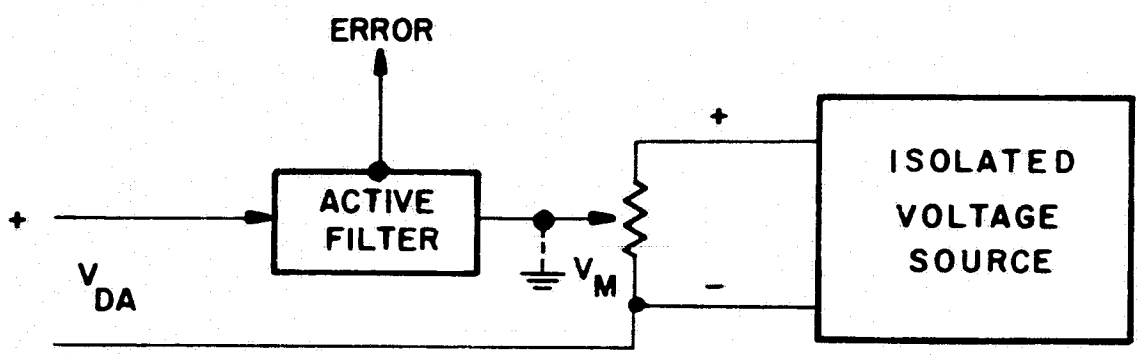


Figure 2-7. Simplified Bridge Network [2]

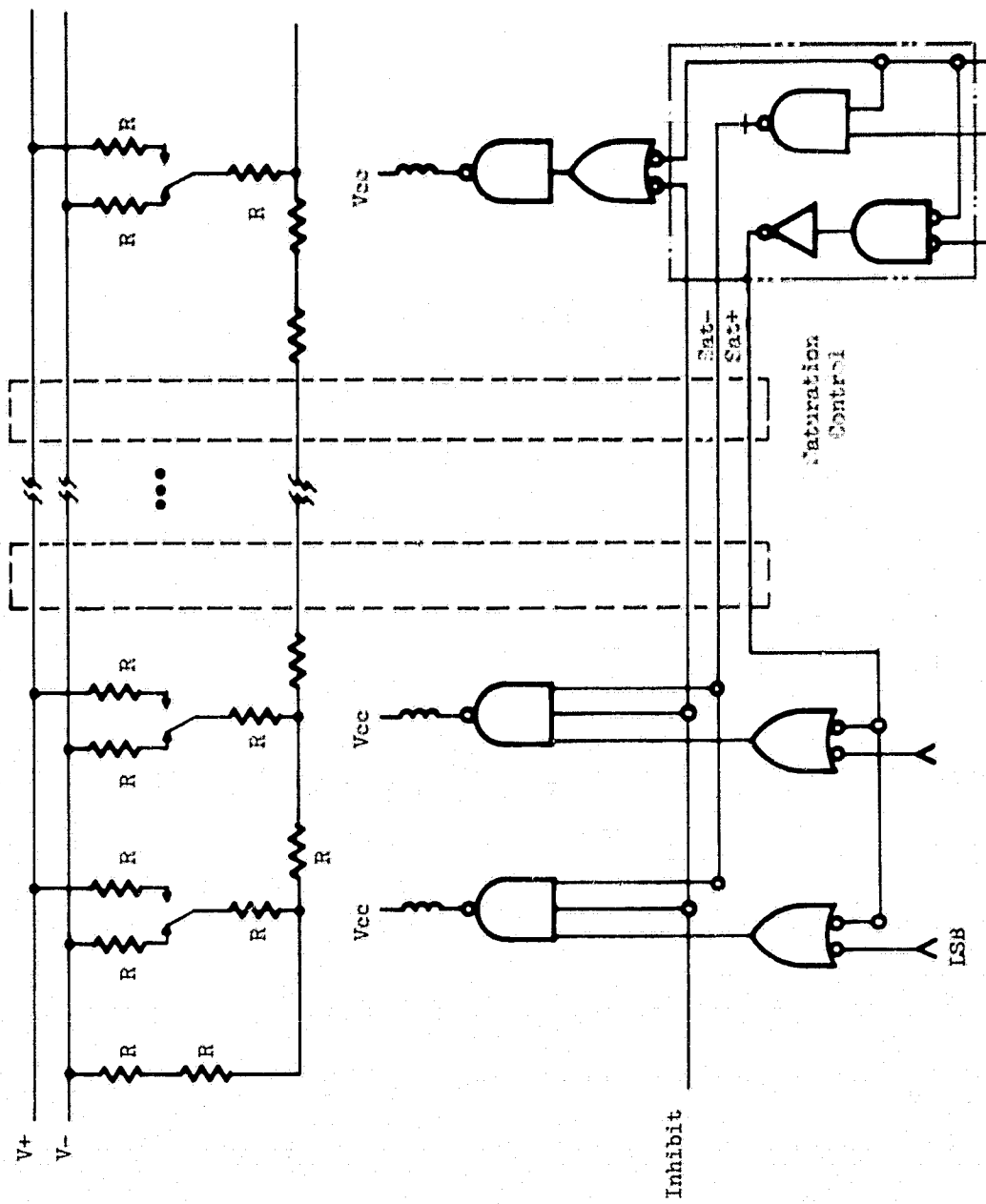


Figure 2-8 D/A System for HR & EI Axes

The two hand axes utilize a seven bit D/A to provide the analog error input to the SCR drive amplifiers. In this case the input to the D/A converters is the actual digital error in position of these axes. The output of the D/A converter is filtered and amplified to remove the high frequency components caused by bounce from the relay contacts and to increase the drive of the output without loading the D/A converter. Since this is actually error, the length of the D/A is determined by quantization considerations rather than range. This D/A converter is detailed in Figure 2-8

2.4.3 Amplifiers and Filters

To delete the synchronous noise which is generated by the SCR drives the amplifiers utilized for the seven major axes include an active RC filter with an upper limit of 20 hz. This amplifier also decreases the output impedance to provide more drive for the SCR drive circuits. For the two hand axes it is not possible to use the 20 hz cutoff since the system approaches its null point at about 300 pps rate. The 20 hz cutoff would have greatly reduced the frequency response of the system. Since this amplifier is closely coupled with its input signal, the output of the D/A converter, it is not seriously effected by the noise which the other axes see due to the long lines to their various signal sources.

2.4.4 Null Detector

The null detector compares the error for positive, negative, or zero. If the error is non zero, it grounds either the positive

or negative error line. The positive and negative error lines are brought back into the computer through the GPI and are used by the hardware A/D conversion scheme when the SAD mode is not enabled.

2.5 The Computer

To control the manipulator, a DDP-116 digital computer manufactured by Computer Control Division (3C) of Honeywell, was selected. This machine is equipped with hardware multiply/divide, high speed paper tape I/O, a Real Time Clock, and a teletype. All of which are used by the manipulator system. In addition a line printer, a card-reader, a disk memory, a magnetic tape transport and a plotter are interfaced to the computer although they are not used directly by the manipulator programs. The basic machine consists of 8,192 sixteen bit words of core memory having a basic cycle time of 1.72 μsec and typical instruction execution times of 3.4 μsec for addition, 9.5 μsec for multiplication, and 17 μsec for division. All of the above operations are fixed point.

2.6 The Discrete General Purpose Interface Unit (GPI)

The GPI provides a means of communication both data and commands between the computer and the MCI. It contains a sixteen bit input/output data buffer and decoding to provide five external control pulses (OCP) and six external sense lines (SKS) which include an interrupt line. The interface is equipped for use with Direct Multiplex Control (DMC) to permit high speed, cycle stealing data transfers and is connected to the priority interrupt structure of the

computer. This interface has two internal flip-flops which are used to control and to synchronize communication between the computer and an external device. These are the MODE flip-flop which indicates whether the external device is responsible for loading the buffer register (Input Mode) or the computer is responsible for loading the buffer (Output Mode). This information is brought to the interface connector so that the device can determine the direction of data transfer. The second control flip-flop is the READY flip-flop and it is used to synchronize data transfers. In Input mode the computer will wait for the Ready flip-flop to be set by the external device before attempting to transfer data. When the data has been transferred the computer will reset the Ready flip-flop. In Output mode the computer will transfer data to the buffer if the ready flip-flop is set and then it will reset the Ready flip-flop. When the external device has taken the data, and is ready to accept more data, it will set the Ready flip-flop to indicate more data is desired. A detailed list of instructions and the necessary connector information is contained in Appendix VII. This interface has been assigned channel address '40.*

2.7 Integrated Circuit General Purpose Interface (ICGPI)

The ICGPI is similar to the GPI except that it has fully decoded SKS and OCP functions giving it 12 external OCP lines and 13 external SKS lines. In addition, this device has the plus five and

*'will be used to indicate an octal number.

zero voltage levels which are readily connected to most logic today. This device has been assigned channel '37. A complete list of connections and instructions is contained in Appendix VIII.

2.8 Manipulator Control Logic

The manipulator control logic was constructed in two sections. The first section was constructed using discrete logic components and the second section was constructed using integrated circuit components. The reason for the change in construction techniques was that the price per unit of logical power for integrated circuits had greatly decreased in the two years since the first section was constructed and it had become much less expensive to build logic in integrated circuits and interface them to the discrete logic than to build the logic in discrete form.

2.8.1 Discrete Control Logic

The Discrete Control Logic consists of seven ten-bit buffers, one four-bit buffer, control logic for A/D conversion, and multiplexing logic to permit data transfers through the GPI data buffer.

2.8.1.1 Buffer Registers

Each of the seven major manipulator axes utilizes one ten-bit buffer register when under closed loop control. This buffer is used as the link between the digital and analog portions of the control system. The buffer contains the digital setpoint output by the computer in AUTOMATIC mode and is connected to one of the ten-bit D/A converters to obtain an analog setpoint. This analog

setpoint is then compared with the analog position from the feedback potentiometer and the resulting difference or error is used as the input to the drive system. This buffer in conjunction with the null detector (see Section 2.4.4) is used in the A/D conversion (see Section 2.8.1.2).

The four-bit buffer is used to provide control over the relay drives for the hand rotation and hand grasp motors. These drives are enabled only in the AUTOMATIC mode when the Hand SCR drives are not selected. They provide half power drive capability for these two axes. This drive has the disadvantage that it is open loop and must be timed.

2.8.1.2 Position Measurement (A/D Conversion)

The potentiometers connected to the seven major manipulator axes always reflect the absolute position of these axes and so they can be used to obtain a digital representation of this position. This position can be obtained in several ways. For this system two basic procedures are possible. The first utilizes the counting feature of the ten-bit buffers to modify the digital representation; the second utilizes the computer.

The first method, Hardware Analog to Digital Conversion (HAD), uses the output of the null detection circuitry to gate the clock inputs to the counters to count them in a direction in order to reduce the magnitude of the error. This technique is restricted by the bandpass of the amplifier and comparitor circuitry to a rate of about 15 hz which can provide for a worst case conversion time of up to 140

seconds. However, this feature does work fairly well for tracking manual moves since the rates of motion are not much higher than this and can be followed fairly well. This method is only applicable:

1. when the manipulator control interface is in MANUAL mode and the SAD mode is reset;

2. when the FORCE mode is set, and the SAD mode flip-flop for a particular axis is reset, and the manual control for that axis is actuated.

The second method utilized the computer program to obtain the position. This operation is referred to as Software Analog to Digital Conversion (SAD). It utilizes the output of the null detector to determine the sign of the error and thus to select the next approximation. The algorithm used is described in Section 3.5. In general, the computer outputs an approximation to the selected axis and then waits approximately seventy milliseconds for the output to stabilize. The error is then tested. Based on the sign of this error, a new approximation can be made until the position is determined. This operation is performed using a binary search technique which requires only ten approximations and thus gives a fixed conversion time, limited only by the settling time of the amplifiers, or approximately 0.7 seconds independent of the magnitude of the error. The technique requires that the SAD mode be enabled for the axes which are being converted. A single SAD mode flip-flop is provided in the MCI and is applicable when FORCE mode is not set. When FORCE mode is set, an individual flip-flop is available for each of the seven major manipulator axes.

2.8.1.3 Multiplexing Technique

The data transmitted between the computer and the seven ten-bit manipulator buffers is multiplexed through the single GPI data channel. Addressing is provided by initiating the appropriate sequence and incrementing the channel address after each transfer. This technique provides for the sequential transferring of data but does not lend itself to single, nonsequential data transfers. For example: The buffers are addressed SP, EP, WP, SR... If it were necessary to access the WP buffer, it would be necessary to first position through the SP and EP buffers before reaching the WP buffer. This addressing problem can be solved by providing addressable channel selection as described in Section 2.8.2.2.2. With the sequential technique, only the latter buffers can be ignored when making transfers, the earlier ones in the sequence must be transferred in order to reach a given position.

The general technique utilized for sequential transfers calls for the program to first output an OCP '0240 to clear the AXIS counter and to initialize the multiplex logic; then an OCP '0140 is output to place the GPI in OUTPUT mode; and an OCP '0440 is output to set the AXIS counter to "1". The first word output by the computer will then be automatically gated to buffer "1" and the AXIS counter will be incremented. This process will continue until word "7" is output. At this time the AXIS counter will be reset and the multiplex logic will be set to ignore all transfers until the next OCP '0440 is given.

The states "1" through "7" of the AXIS counter are used to select the seven major manipulator axes in the order SP, EP, WP, SR, X, Y, and Z as shown in Figure A-IX-2. The "0" state is used to input the error status from the null detection circuts and to output from the computer to the four-bit hand control register.[3] In addition, when FORCE mode is enabled, the AXIS register must be in state "0" to permit the Discrete Channel Selection Register to select the desired channel or channels (see Section 2.8.2.2.2). The general sequence of operations required for data transfers under the discrete selection format is shown in Figures 2-9 and 2-10.

2.8.2 Additional Control Logic

The first section of manipulator control logic was completed and the algorithms by Beckett(1) were tested. It was determined that additional refinements and features were necessary to permit additional modes of operator control, including:

1. individual axis addressing for the manipulator buffers
2. device input/output without changing control states
3. closed loop control of the Hand axes
4. sensing of additional machine conditions.

Because it is more economical to provide the additional control capabilities using integrated circuits and interfacing to the existing discrete logic than to build the new logic utilizing discrete logic technology, a second GPI was constructed rather than expanded the initial unit.

The second section of the manipulator control thus

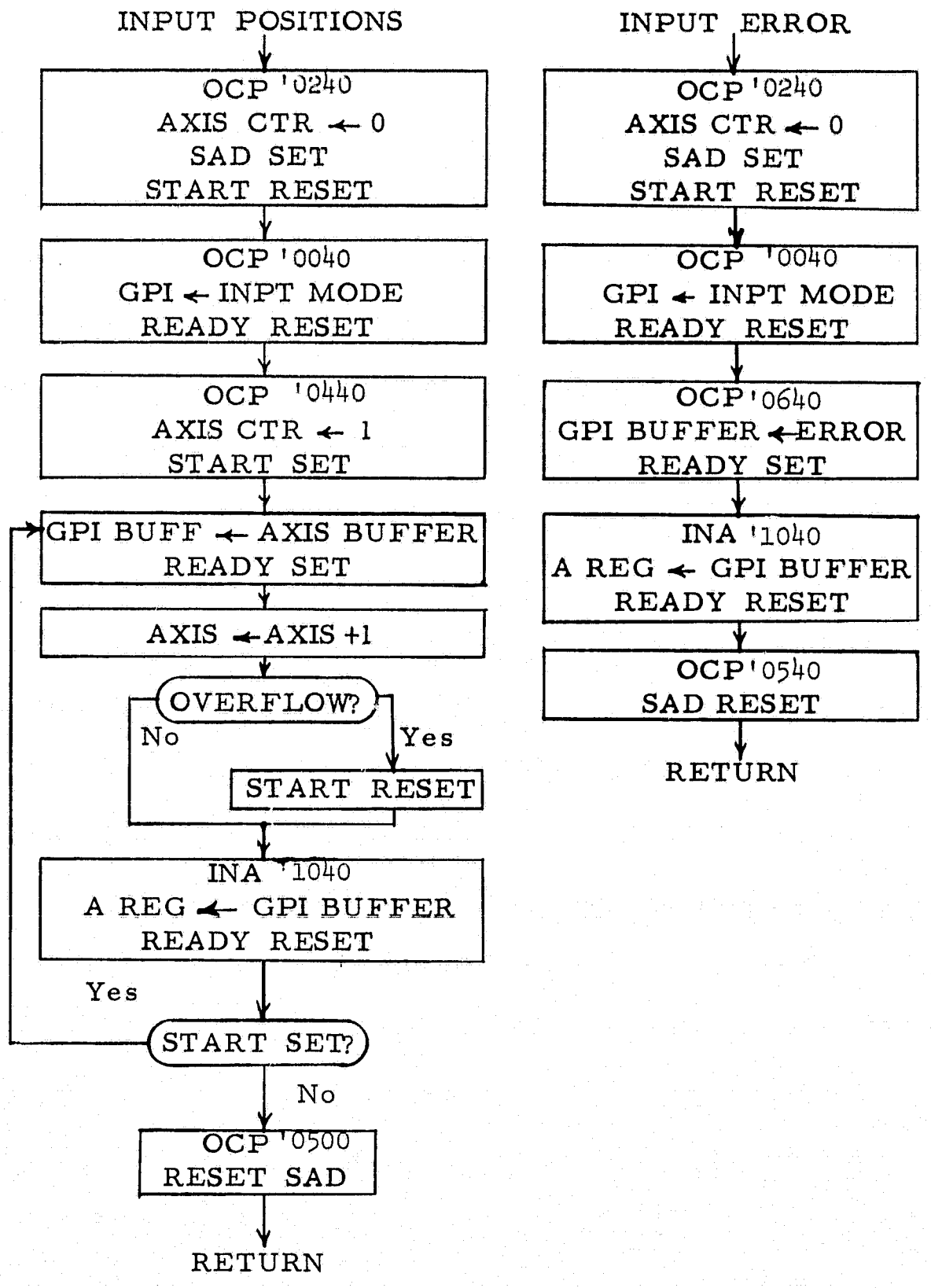


Figure 2-9. Typical Information Flow -- Manipulator to Computer[4]

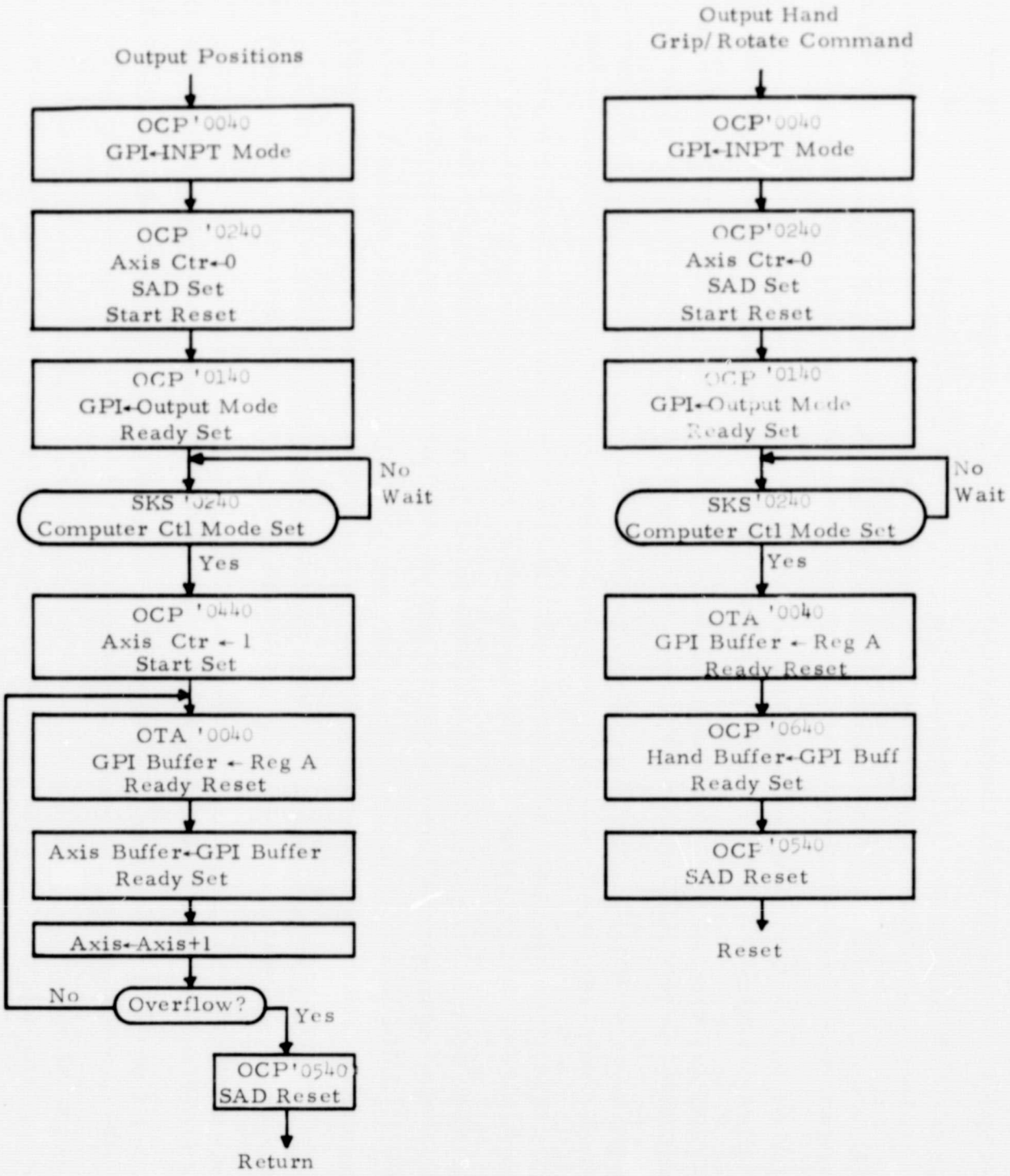


Figure 2-10. Typical Flow of Information From Computer to Manipulator Logic

communicates with the computer through the ICGPI and helps to control the manipulator both directly, as in the case of Hand Rotation and Hand Grasp, and indirectly, as in the case of the FORCE mode of channel selection for the MCI. This unit consists of eight twelve-bit maskable buffer registers, two channel select registers, four counter buffer registers, and various other counters and registers for control sequencing. A block diagram of this control section is shown in Figure 2-11.

2.8.2.1 Twelve-bit Maskable Buffer Registers and Detectors

These registers are constructed with two flip-flops per bit position to permit the selective monitoring of various conditions on the manipulator structure and in the control system. One flip-flop is used to store the current status of the condition and is updated at a rate of approximately 100 hz; the second flip-flop is used to control the connection of the status to a sense line which is eventually connected to the ICGPI external interrupt input. This permits the selection, by the program, of those conditions which are to be monitored by the interrupt line while still permitting the program to input these statuses at any time. An example of the structure employed is shown in Figure 2-12. This circuit is contained on a printed circuit card (DSL011) along with a detection circuit which permits the selection of individual axes for manual control while the MCI is in AUTOMATIC mode. This circuit detects the presence of an error on the manual control input to a given axis and switches the input of the drive amplifiers over to the manual control source.

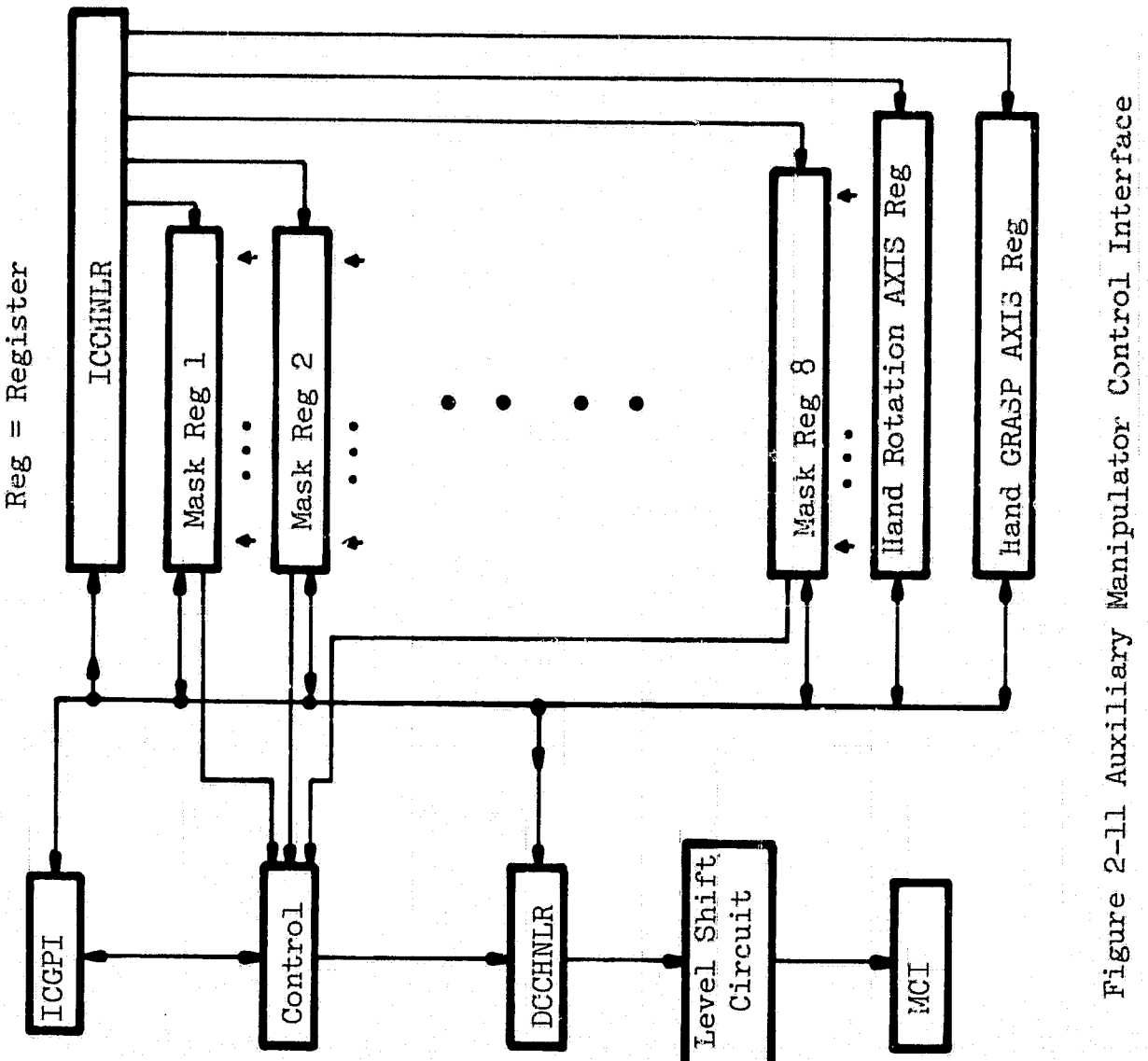


Figure 2-11 Auxiliary Manipulator Control Interface

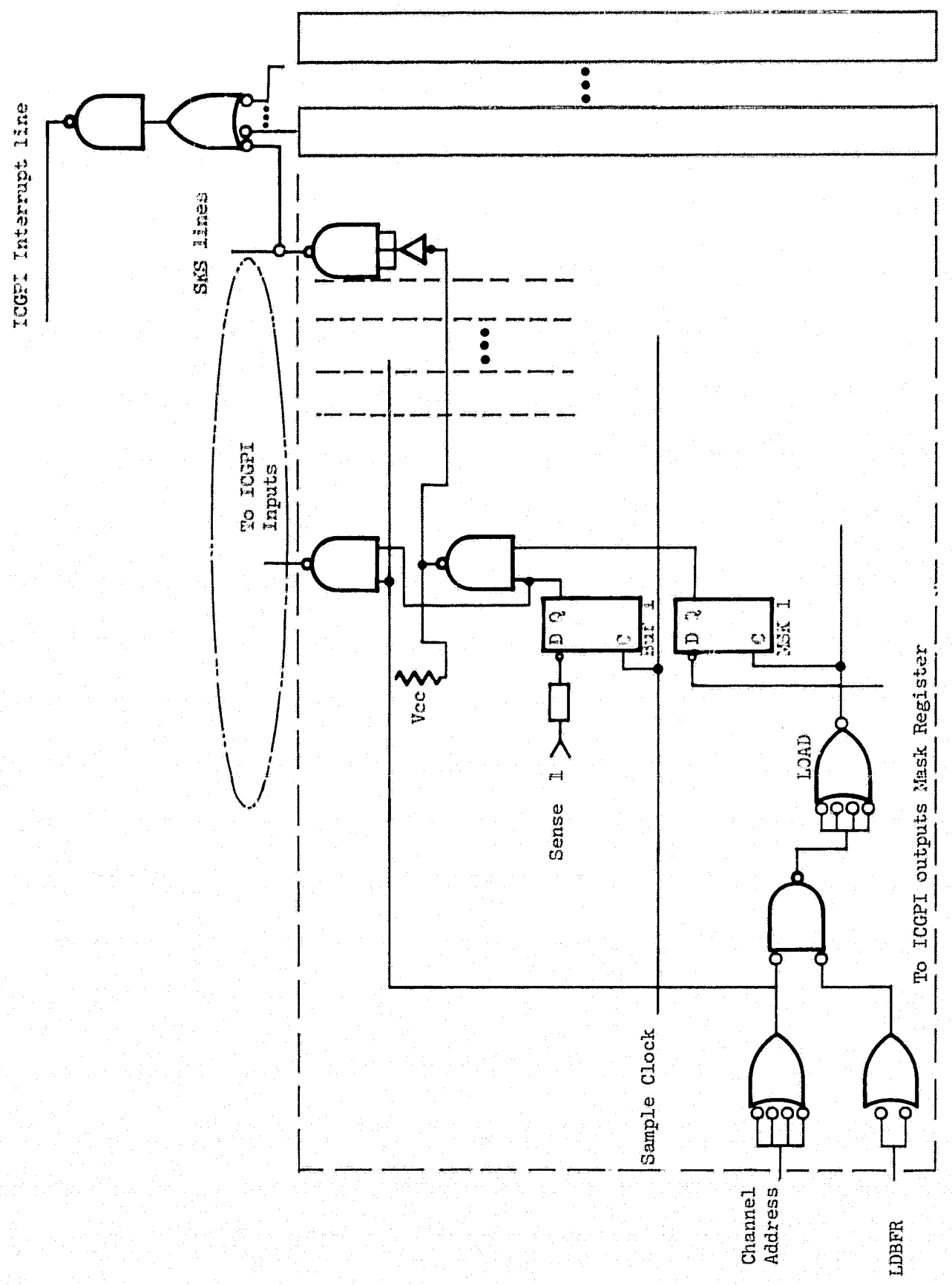


FIGURE 2-12 Maskable Error Registers and Interrupt Structure

This condition is also provided as a status for input to the computer through the Maskable Buffer Registers. The detector circuit is detailed in Figure 2-13. Its application is shown in Figure 2-4.

2.8.2.2.2 Channel Select Registers

The formats of these registers are contained in Appendix IX. However, these two registers are used basically for channel selection and for status storage.

2.8.2.2.1 Integrated Circuit Channel Register (ICCHNLR)

This register is sixteen bits in length and has both single-bit shift and parallel load capability. In addition the length of the shift may be externally wired to permit the selection of only part of the range. The short region must start with channel "1". This register is used to select the appropriate Maskable Buffer Register as defined by a unique bit for each of channels 1 through 8, and is used to select the Hand Rotation and Hand Grasp through channels 9 and 10. Whenever a transfer is completed, the register is shifted one bit toward the next higher channel number. This register is always initialized to "100000 (channel "1") on powerup and by an OCP '0237 (device clear). In addition, the register is set to this value following a transfer to/from the last channel as selected by the external connections.

This register can be loaded from the computer by the following sequence of commands:

1. OCP '0137 ICGPI in OUTPUT mode
2. OCP '1237 Set ICCHNL

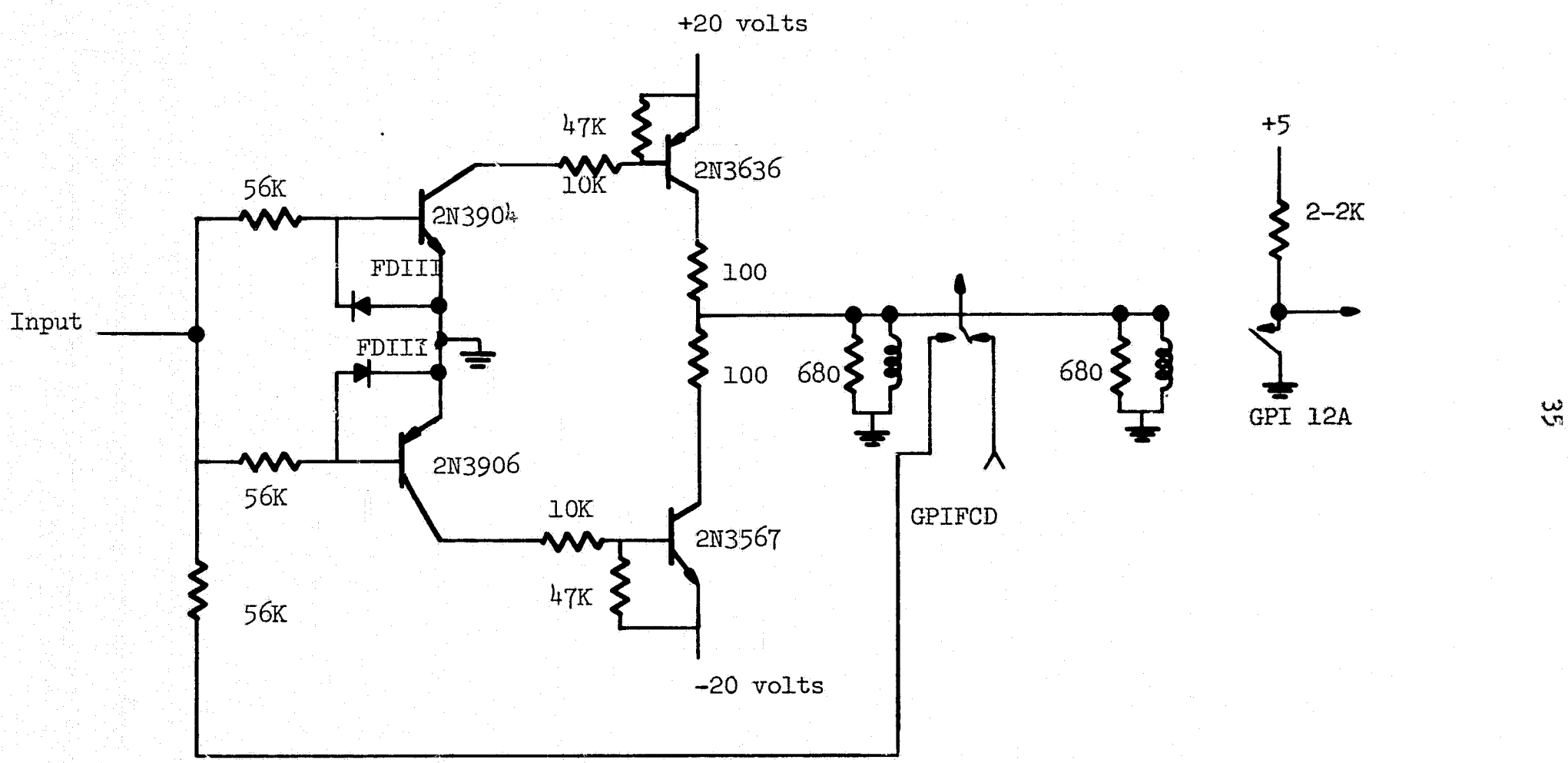


Figure 2-13 Manual ERROR Sense Amplifier

3. LDA (Channel) Desired channel number

4. OTA '0037 Output to ICGPI

5. JMP *-1

The next data transfer will be directed to/from the channel (or channels) selected by this sequence. Since each axis is uniquely selected by one bit of this register, it is possible to transfer to/from more than one register at a given time. If this is the case, the program should reset the register after the transfer has been made.

2.8.2.2.2 Discrete Logic Override Register (DCCHNLR)

The Discrete Logic Override Register is used in conjunction with the MCI through level shifting circuits to select individual axes for single axis transfers to the manipulator buffer registers. In addition, it contains several flip-flops which are used as individual SAD mode flip-flops when FORCE mode is enabled, and a single flip-flop which is used to force the MCI to stay in AUTOMATIC mode. To use this register for channel selection on the MCI, the MCI AXIS counter must first be cleared, with an OCP '0240. Then the contents of bits 8 through 16 of the DCCHNLR may be used to select an axis or several axes by enabling FORCE mode with an OCP '1037.

This register may be loaded through the ICGPI by using the following sequence of instructions:

1. OCP '1037 ICGPI in OUTPUT mode

2. OCP '1137 Set DCCHNL

3. LDA DATA New contents for DCCHNLR

4. OTA '0037 Output data

5. JMP *-1

DCCHNL is reset automatically after the data transfer. This method may be utilized if transfer to only one of the manipulator axis buffers is desired. After the address has been output to the buffer and the FORCE mode has been set up, the data transfer is initiated as with axis counter state zero. The sequence of instructions and operations is shown in Figure 2-14.

2.8.2.3 Counter/Buffer Registers

These four registers are utilized in the closed loop control of the HR and HG axes. Basically there are two registers assigned to each axis. The first register is utilized as a tracking counter and is connected to ICGPI input to permit interrogation by the computer of current position of the associated axis. In addition these counters are connected to display lights to visually indicate the position to the operator. This register is initialized either by a manual reset button or by a fixed origin point to provide a relative reference system utilizing an incremental quantizer.

The two registers used for this purpose are somewhat different in counting configuration although the elements utilized in the construction are similar. The HR register is sixteen-bits in length with the lower six bits counting in binary sequence on a scale of forty-five, and the upper ten bits counting in straight binary. When coupled with a quantizer having a resolution of one pulse per degree, a resolution of one turn per increment at bit ten results. This

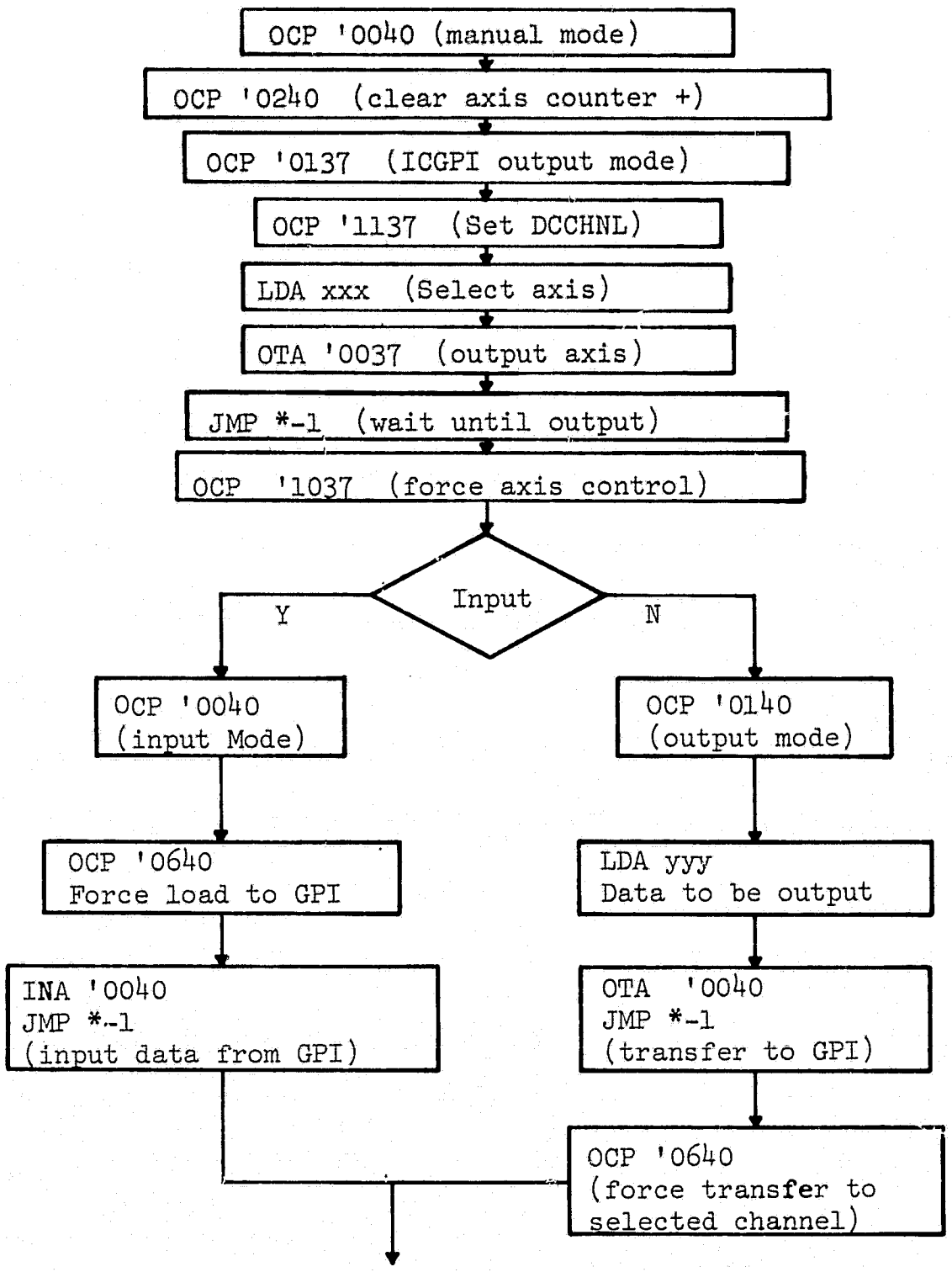


Figure 2-14

Flow Chart for data transfers in FORCE mode

counter simplifies the operator's task should it become necessary to visually interrogate this axis. The counter utilized for the HG is straight binary and thirteen bits in length. This length, coupled with a resolution of one thousand parts per inch, gives a range of approximately eight inches which greatly exceeds the range of the hook or hand. When no tool is fitted, the actuating rod can be used as a pointer with a working range of approximately five inches.

The second set of registers is utilized directly in the process of closing the loop between digital and analog portions of the HR and HG controls. These registers are referred to as error counters since a nonzero content will result in power being supplied to the motors so as to reduce the error. The error can be output from the computer through the ICGPI and the zero condition can be sensed by using the status register on channel "1" of the AMCI. The contents of these registers is converted to an analog potential by utilizing the D/A converter shown in Figure 2-8. Both error counters use a binary sequence, but the lengths are not the same. The HR error counter is sixteen bits long, and the HG error counter if fourteen bits long. In both cases the highest order bit is taken to be the sign bit. A block diagram of the system utilizing these registers is shown in Figure 2-15.

2.8.2.4 Interrupt Structure

To permit the detection of operator intervention or changes in status indicators, an interrupt structure was constructed in the AMCI utilizing the eight Maskable Buffer Registers. As indicated

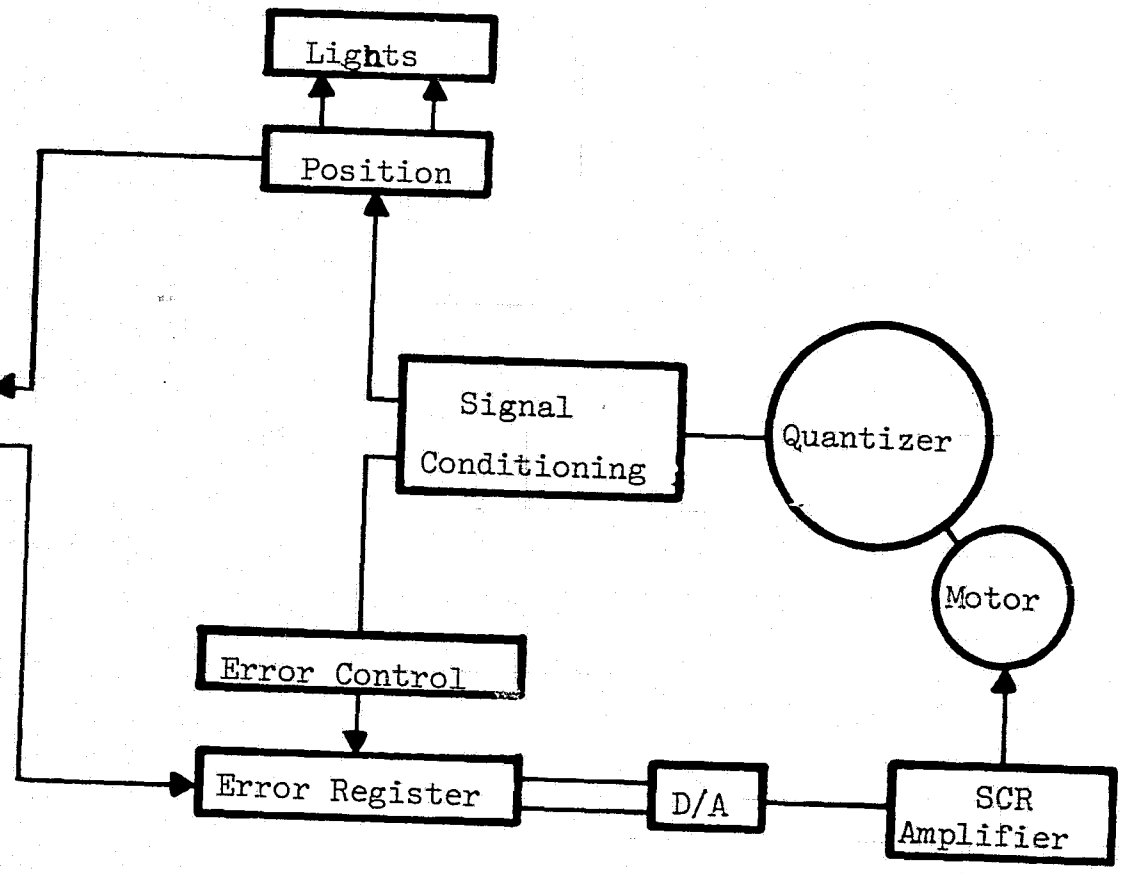


Figure 2-15 HG & HR Axis Block Diagram

in Section 2.8.2.1, the status of any individual bit may be checked at three different levels:

1. any status bits set
2. any status bits set for an axis
3. a particular bit set

The first two conditions can be masked off to avoid generating an interrupt. The third condition is obtained by interrogating the status register and is always the current status of the register as of the last clock time. The clock used to update this status can be inhibited by giving an OCP '0437 and can be reenabled by giving an OCP '0537 or by any Clear condition.

Since the interface is designed to transfer data at a high rate, and since the Ready flip-flop is one of the interrupt conditions, it is necessary to leave the Ready flip-flop in a reset state when external interrupts are desired from the ICGPI. To implement this, an interlock has been provided which will permit one data transfer and then disable the ready response of the AMCI. This interlock is set with an OCP '1337 and can be reset by several conditions, in particular OCP '0637, OCP '1137, OCP '1237, and OCP '0237, as well as, any manual Clear condition.

2.8.2.5 Data Transfer Rate

The transfer rate for the auxiliary Manipulator Control Interface is dependent on the machine transfer for meaningful sequential data transfers. This occurs since the time required to complete a data transfer is between 4.5 and 5.4 μsec, from the time the Ready flip-flop is reset. The transfer is initiated on the next pulse of

a 1.1 megahz clock following the resetting of Ready. The data is transferred one clock time later, and an additional three clock times are used to handle various control functions including the shifting of the channel select register for normal transfers and the resetting of various status flip-flops. The fifth clock pulse is used to reset the transfer cycle sequence.

2.8.2.6 Changes in MCI

Several changes were required in the MCI to permit the Force mode of control. These include: the addition of inputs to permit selecting an axis buffer from outside th MCI; inhibiting several control signals when in FORCE Mode; the addition of gating to permit forced single axis transfers; and the modification of the HAD mode hardware to permit individual flip-flops to control the count signals for tracking. In order to permit the operator to override the motion of the manipulator while in computer control, it was necessary to rewire some of the circuitry in the analog control section to permit the insertion of the circuit mentioned in Section 2.8.2.1.

2.9 Programming Notes

Due to timing considerations and the automatic sequencing inherent in some of the data transfers, there are some possible programming traps which must be avoided when programming this equipment. These conditions will be indicated in this section. Additional programming aids, such as lists of instructions and their functions are contained in Appendices VII-IX.

2.9.1 GPI and MCI

The SKS and OCP instructions for this equipment are indicated in Appendices VII and IX. Due to the limited number of OCP commands available through the GPI it was necessary to perform several operations with the OCP '0240. As a result it may be necessary to give several OCP's in sequence so as not to set undesired modes. For example, to clear the axis counter, it may be necessary to execute an OCP '0040 first; then the OCP '0240; and finally an OCP '0540. The first instruction is used to insure that the computer control mode will not be forced; the second instruction clears the axis counter and sets the SAD mode; the last instruction resets the SAD mode.

Because of the multiplexing technique used, several side affects have arisen and care must be exercised to prevent damage to the system as well as to insure that the proper data has been transferred between the computer and the MCI. When operating in the sequential transfer mode through the GPI the following rules should be observed:

1. Following an INA or OTA instruction allow at least ten μsec before giving an OCP instruction.

2. Before changing modes in the GPI give an OCP '0240 to clear the axis counter.

3. After executing an OCP '0140 (GPI output mode) wait until the Computer Control Mode is established (test with an SKS '0240) before executing an OCP '0440 to initialize the counter. The Computer Control Mode is set using a 20 hz. clock.

By considering the multiplexing technique, one can predict the side effects which will occur. A "START" flip-flop in the multiplexing logic is set when the AXIS counter is initialized by an OCP '0440 and is reset when the counter overflows or by an OCP '0240. If the GPI READY flip-flop is reset and START is set, then the following operations will occur on the next pulse of a 200 khz clock:

1. Leading edge of clock pulse

   a. Data is transferred to (from) manipulator buffer (as selected by current value of AXIS counter) from (to) GPI data buffer.

   b. Ready is set.

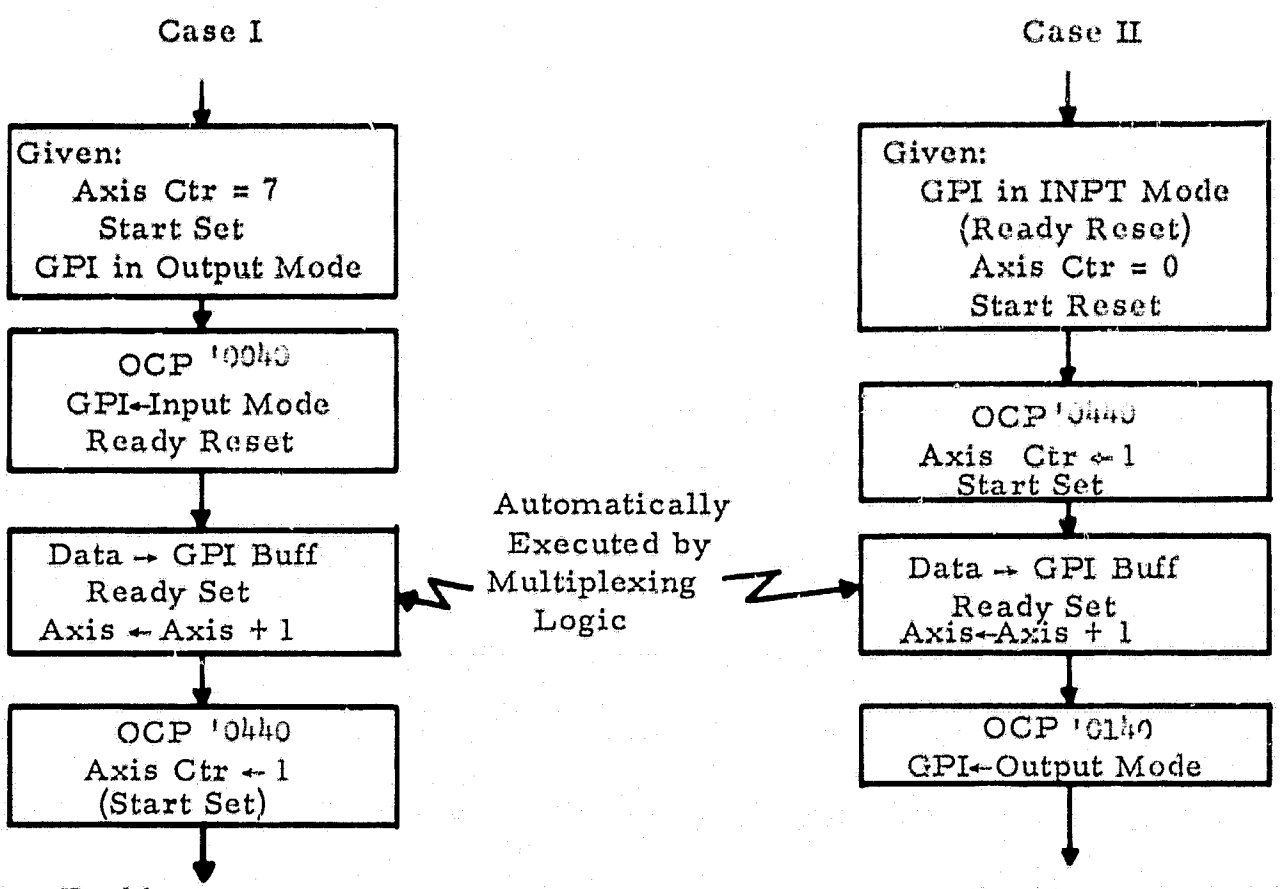
2. After a slight delay, the AXIS counter is incremented by one (if the last value was seven then the counter is cleared and START is reset).

The above operations require up to 10 μsec and the sequence occurs after all INA and OTA instructions. It is important that the of this sequence be permitted before executing any instructions which can change the contents of the AXIS counter (OCP '0040, OCP '0140, OCP '0240, and OCP '0440).

Two examples of the type of problem which can occur if this is not observed are shown if Figure 2-16.[5]

2.9.2 ICGPI and AMCI

Due to the time required for the completion of data transfers (as mentioned in Section 2.8.2.5) several OCP commands should not be given as the next instruction following the completion of an INA or OTA. These instructions are:



Problem:

SP←?
EP←Axis 1
WP←Axis 2
SR←Axis 3
X←Axis 4
Y←Axis 5
Z←Axis 6

Problem:

SP→Axis 2
EP→Axis 3
WP→Axis 4
SR→Axis 5
X→Axis 6
Y→Axis 7
Z→Lost

Figure 2-17. Two Programming Hazards

```
1.  OCP  '0237          Device clear
2.  OCP  '1137          Set Dcchnl
3.  OCP  '1237          Set ICCHNL
4.  OCP  '1337          Disable Ready response
```

When this precaution is observed, the I/O timing chain is given time to complete.

The two OCP commands required for register loading (OCP '1137 and OCP '1237) are recognized only when the ICGPI is in Output mode. They will be ignored (except for resetting Ready response inhibit) when the ICGPI is in Input mode. An example of this is shown in Figure 2-17.

```
1.  OCP  '0137          ICGPI output mode set

2.  OTA  '0037          Output to ICGPI

3.  JMP  *-1            Data transfer initiated
                        when ready is reset  by
                        computer.

4.  OCP  '1137          DCCHNL flip-flop set, re-
                        set at end of data transfer
                        above.

5.  LDA  addr           Address for following trans-
                        fer

6.  OTA  '0037          Output data
                        (this will transfer data to
                        one of the ICCHNL Registers
                        instead of to the DCCHNL
                        register as expected.
```

This problem can be avoided by inserting any non I/O instruction between instructions 3 and 4.

Figure 2-17

Description of the hardware utilized in the Teleoperator system is now completed. Specific hardware construction details are available in references 4, 6, 11, & 14.

FOOTNOTES

CHAPTER II

1. Jon T. Beckett, A Computer Aided Control Technique for a Remote Manipulator, (Report 1-67-44, Digital System Laboratory, Case Institute of Technology), pp. 13-16.

2. Ibid. p. 18.

3. Ibid. pp. 24-28.

4. Ibid. p. 29.

5. Ibid. p. 30.

6. Ibid. pp. 31-33.

7. Ibid. p. 34.

CHAPTER III

TELEOPERATOR CONTROL ALGORITHMS

3.1 INTRODUCTION

Several position and path control algorithms have been developed during the term of the MMCP. These algorithms were implemented in the experimental manipulator system and are used with several modifications to control the Computer Aided Teleoperator System. These algorithms have demonstrated the ability of the computer to act as an intermediary between the manipulator and the operator. The computer has made it possible to greatly reduce the amount of information required from an individual operator in order to perform a specific task.

A number of special programs have been written to provide intermediate information and to facilitate the generation of AUTOMATIC mode programs to be run by the AUTOEXC. These programs include:

1. a family of printout routines
2. two routines to perform software analog to digital conversion
3. specific mode control operations
4. status control routines
5. trace routines

An executive routine permits the operator to select any of the individual routines or any special functions at random by using the teletype console. In addition, the computer console and the operator's control panel are used for various functions to permit a degree of operator intervention over specific routines during their execution.

A second executive routine (PUNCH) permits the generation of paper tapes containing specific operations recorded in either an absolute or a relative format. This paper tape is then interpreted by a third executive routine (AUTOEX) which processes the paper tape prepared by PUNCH and, with the operator's aid, routes the data and control information to the specified routines.

The coordinate system used for the teleoperator system is defined in Appendix I along with the quanta size for each of the nine degrees of freedom.

A list of the available subroutines is given in Appendix III along with a brief description. Flow diagrams and more detailed descriptions are available in separate documents.[2][5]

3.2. Executive Control Programs

The "Teletype Executive Control Program" (TTY Exec) provides the principle means of communication between the operator and the computer. When all routines have been loaded into the computer memory, the operator starts execution of these routines through the TTY Exec. The TTY Exec performs several operations to initialize the system and then requests that the operator indicate the current tool fitted. Following the operator's keyin of the current tool, the TTY Exec requests a new option and waits for the operator to select it.

When a new option has been selected the TTY Exec transfers control to the selected routine at which point additional information will be requested if necessary. Then the control will be transferred to the execution routine associated with the requested option. Most of the routines associated with manipulator motion type operations contain an

input routine to handle the requesting and formatting of desired control parameters and an execution routine to process the parameters for actual manipulator motions. This permits the use of either or both of these routines depending on the current executive mode. Both routines are used for the TTY Exec. The input routine is used for the Punch control program and the operation routine is utilized by the AUTO Exec. These will be described in Section 3.2.2.

Two examples will indicate the selection technique:

Example I: Instruct the computer to type the current size of the grip (unscaled).

1. The operator first types a "T"--the TTY Exec transfers control to the type subroutine.

2. The operator then types a "G"--which instructs the type routine to type the position of the Hand Grip.

3. The type routine then reads the current Hand Grip position from the AMCI and then types out "Grip NNNN" where NNNN is the current value of grip size.

Example II: Instruct the manipulator to move 5 inches along a vector as defined by the hand orientation.

1. The operator first types a "V"--the TTY Exec then calls the routine "IDIST."

2. IDIST types out DIST: and waits for the operator to specify the distance in inches.

3. The operator then types in the distance, in this case 5 followed by a carriage return.

4. The IDIST routine then stores this in the memory buffer DIST.

5. Control is returned to the TTY Exec which immediately transfers control to the VECTOR subroutine which outputs a series of increments to drive the manipulator the specified distance.

Figure 3-1 indicates the major options available through the TTY Exec. Appendix II contains a flow diagram for the TTY Exec and a complete list of the possible teletype commands and the additional information required.

An important feature of the TTY Exec is that it permits the operator, the Real Time Clock, or the external device to interrupt the current routine as necessary to permit operator intervention and automatic system status checking. In the first case, the TTY Exec will check to determine the source of the interrupt and, if it is the teletype, will return control to the TTY Exec and terminate the current operation. If the interrupt is from the operator's control panel the TTY Exec will transfer control to a status checking routine to determine the desired response and will then return control to the interrupted program. The third case is for the Real Time Clock (RTC). This is used to update a pseudo RTC and to check the status before returning control to the interrupted program. The teletype interrupt can be simulated in several ways when teletype input is anticipated: 1. the operator can set sense switch 4 on the computer console; and 2. the operator can type an ALT MODE on the teletype.

When an error interrupt is sensed by the TTY Exec, the computer outputs the necessary commands to stop all manipulator motion. This feature is included to permit the operator to make a "panic" stop if necessary.[1]

3.2.2 The Automatic Mode Executive Routine and Punch Routine

The Automatic Mode for the Case Teleoperator Control system

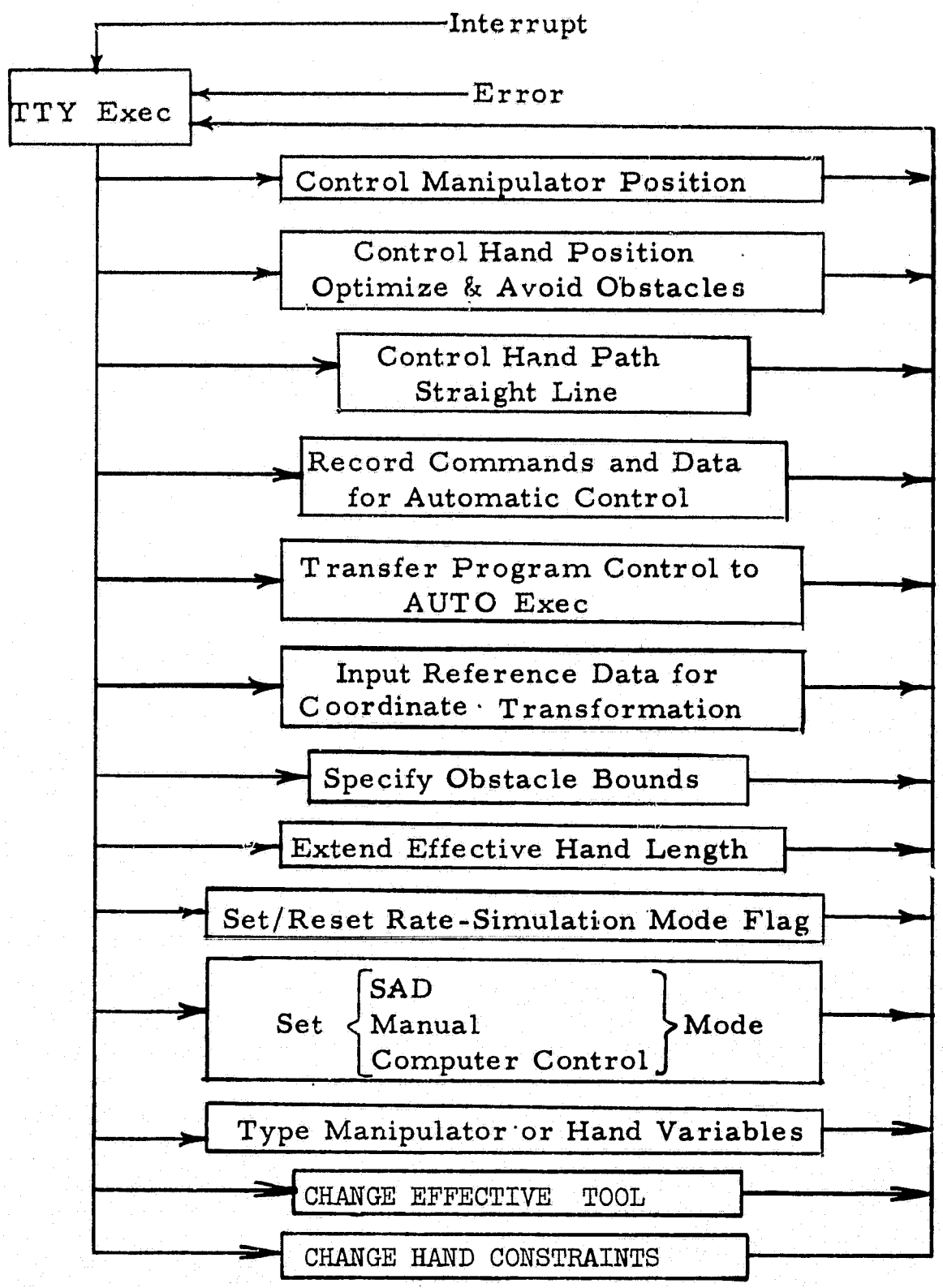


Figure 3-1. TTY Exec Options[2]

is based on prerecorded commands in either an absolute or a relative format. In this case the commands are recorded on paper tape, but they could be recorded on any other media such as disk or drum or magnetic tape. The paper tape was selected as it required the smallest amount of program overhead to perform the desired task.

3.2.2.1 Punch Routine

When the operator desires to prepare a program for use by the AUTOEX, it is indicated to the TTY Exec with a "P" keyin. This is followed by a second keyin to indicate which of the routines is desired. Currently fourteen different types of prerecorded blocks of information may be prepared. Some of these blocks, however, may be prepared in several ways. In general the punch routine will first transfer control to the input routine for the desired option and then after the data has been obtained it is formatted and punched out in an encoded block. Once the data has been punched, control is returned to the TTY Exec. The next operation can be punched by once again calling the punch routine with a "P" keyin.

3.2.2.2 Automatic Mode Executive Program

When the operator desires the use of a prerecorded operation sequence, he first selects the desired sequence and loads the paper tape in the high speed tape reader. After loading the tape the operator calls the Automatic Mode Executive routine with a "R" keyin to the TTY Exec. Control is then transferred to the AUTOEX routine.

The AUTOEX routine will read a block of paper tape from the previously prepared program, and after checking for a valid program

block, the AUTOEX will transfer the supplied data to the proper buffers and call the execution routine for the selected program. The execution routine will return control to the AUTOEX when execution has completed.

There are six conditions under which the AUTOEX will not retain control of the Teleoperator System:

1. The "TTY Return" Code is encountered.
2. An "ORIGIN" Code is encountered (See Section 3.4.3.4).
3. A program interrupt is forced by the operator.
4. An error condition is encountered by one of the operating routines.
5. Sense Switch three on the computer console is set.
6. A tool change is requested.

Figure 3-2 indicates the major options available through the PUNCH and AUTOEX routines.[3]

3.3 Control Algorithms

3.3.1 Manipulator Position Control

The most basic manipulator control algorithm allows the operator to directly specify each of the seven basic manipulator axes. This can be done in either an absolute form or an incremental form, but these positions must be specified for each of the axes that require change. Axes that are not specified are assumed to be unchanged. In general an input routine will obtain the final values desired and store them (in scaled form) in the FMV (final manipualtor variables) buffer. The routine FBUF then converts these values to the ten-bit manipulator position values and stores them in an output buffer (OBUF). The routine OMANIP is then called to output the contents of OBUF to

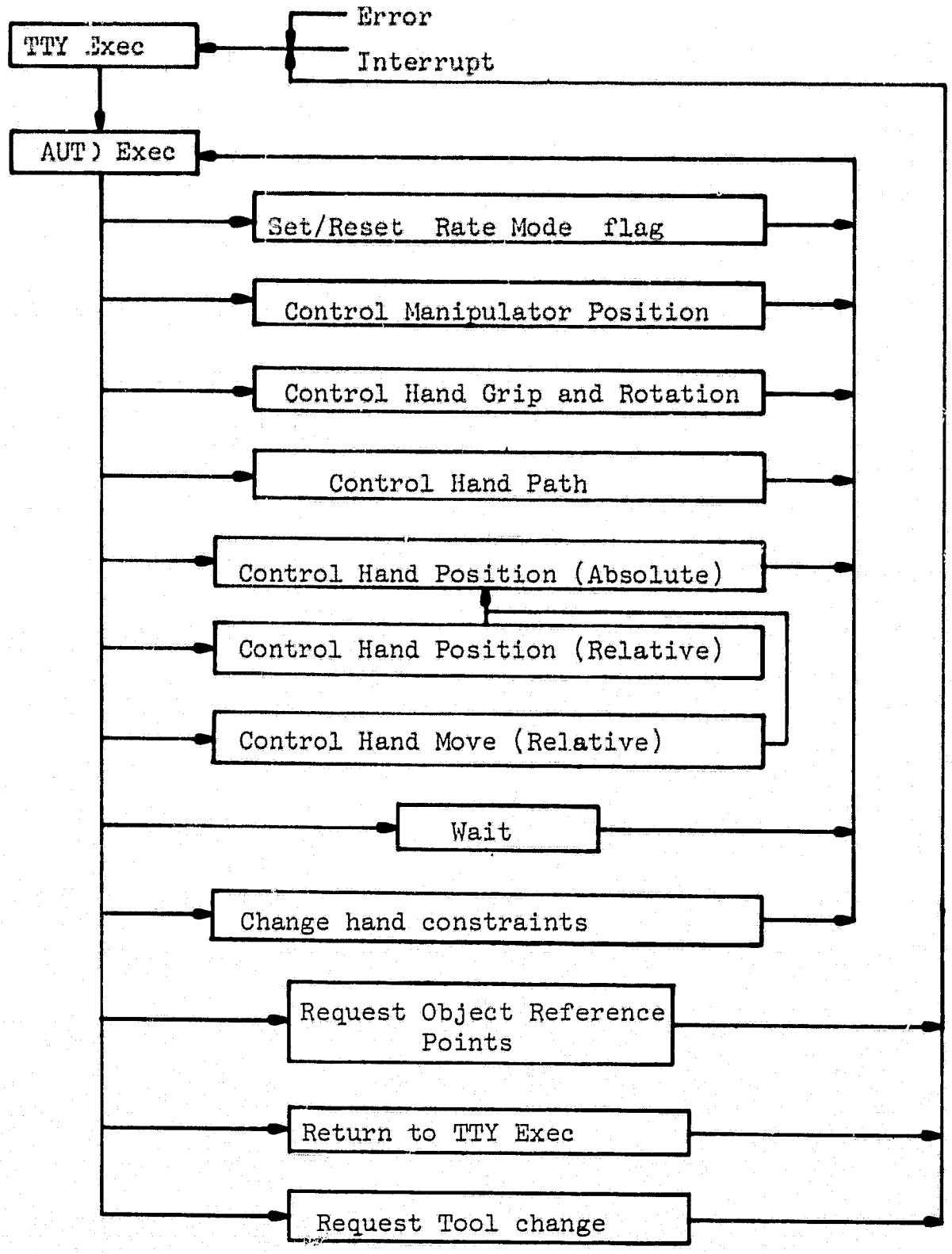


Figure 3-2. AUTO Exec Options[4]

the manipulator buffers. A wait of approximately 17 msec is made between outputs to prevent simultaneously turning on two motors. The subroutine ETST is then used to test for possible terminating conditions and to check for sensor indicated position changes.

When this mode is used through the AUTOEX, the FMV buffer is filled from the paper tape. Then the three subroutines specified above are called. When the operation is complete, control is returned to the AUTOEX.

For control through the TTY Exec, the routine ITMNIP is called to obtain the commands of the operator with regard to the desired motion by the manipulator. The ITMNIP routine will update the FMV buffer before calling the three subroutines named above.

These subroutines are also used by the VECTOR and Hand Position Control subroutines described in the following sections.

A flow diagram for ITMNIP appears in Figure 3-3. The operator can select any one of three modes of command. The first is indicated by a C keyin and is used to center the manipulator structure in its coordinate system. This is intended primarily for use when the operator is trying to obtain a fix on the coordinate system. The second option is indicated by an "A" keyin and permits the operator to specify the absolute position of any one or all of the manipulator axes. The third option is specified by an "I" keyin and indicates that the following inputs are to be added to the present manipulator variables to obtain the new FMV buffer contents. As indicated above, in the "A" and "I" options any axes not referenced will retain their current value.

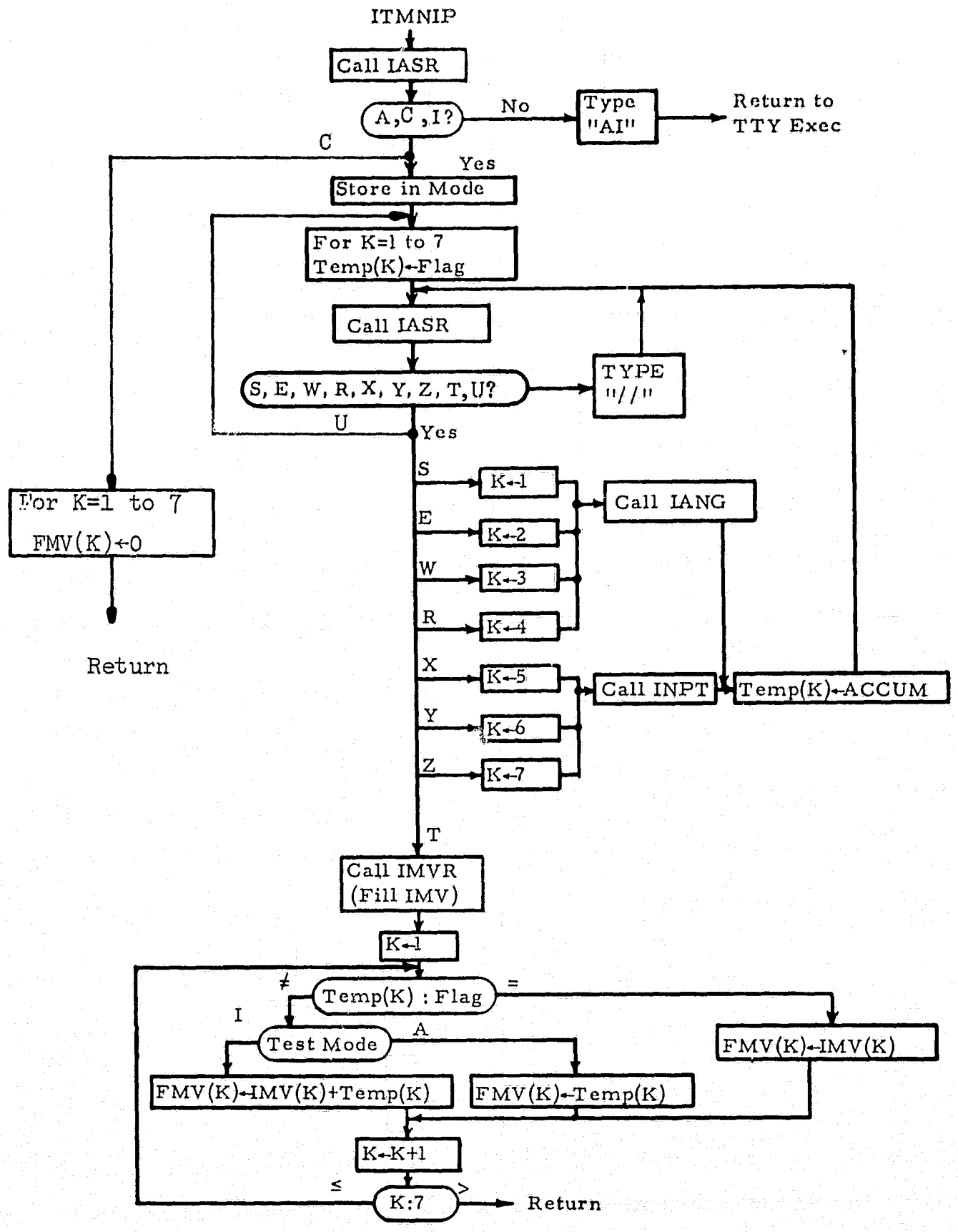


Figure 3.3 ITMNIP Flow Diagram[5]

This algorithm is of limited value for most operator generated moves through the TTY Exec. The reference system is not clear and the operator is hard pressed to determine the results of a move requiring the motion of more than one or two axes. This option is useful in the Automatic mode since it permits the manipulator to be moved into particular configurations when necessary.

3.3.2 Hand Grip and Rotate Control

The Hand Grip and Rotate axes are a special case in that they have two types of control available. The first type is open loop and all operations performed using this mode utilize a timed motion. The second control system utilizes a closed loop control system with incremental feedback. This closed loop system is a relative positioning system and thus the exact position of one of these axes is dependent on where the origin was set. It is possible for the operator to set either of these axes directly, or the Hand Rotation will be set on the first positive move through the zero index. With the closed loop system it is possible to control the exact position of the axes relative to the preset origin. Thus it is not necessary to time the moves to provide a specified motion. The flow diagrams for the "IGRIP" input routine and the "GRIP" operating routine are shown in Figures 3-4 and 3-5 respectively. In addition to controlling these axes, the IGRIP and GRIP routines also provide a routine with the capability of providing incremental Z motion coordinated with Hand Rotation for th purpose of removing and replacing nuts from studs in the disassembly/assembly operations. With this routine it is possible to indicate the pitch of the thread and

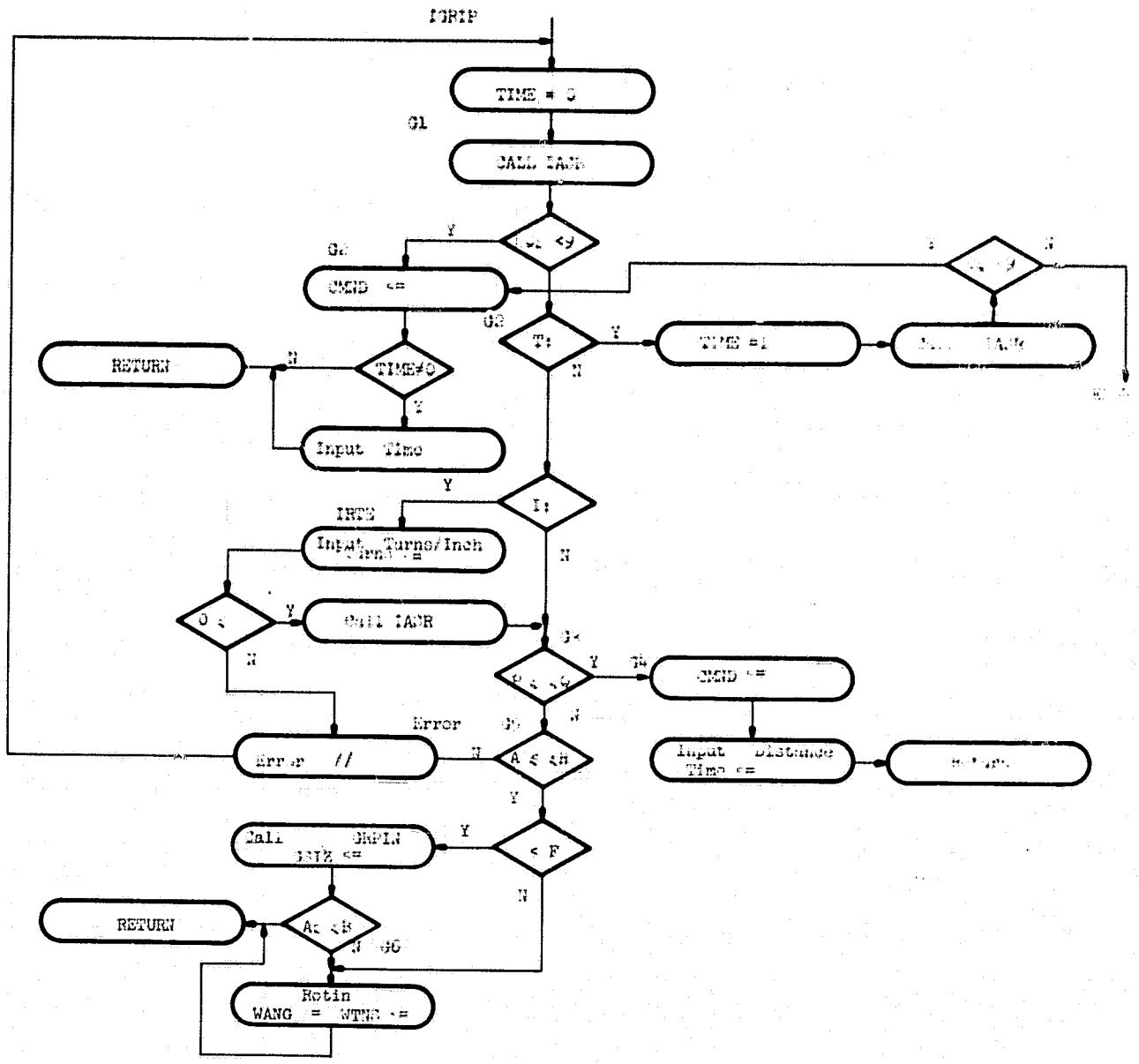


Figure 3-4 IGRIP Flow Diagram

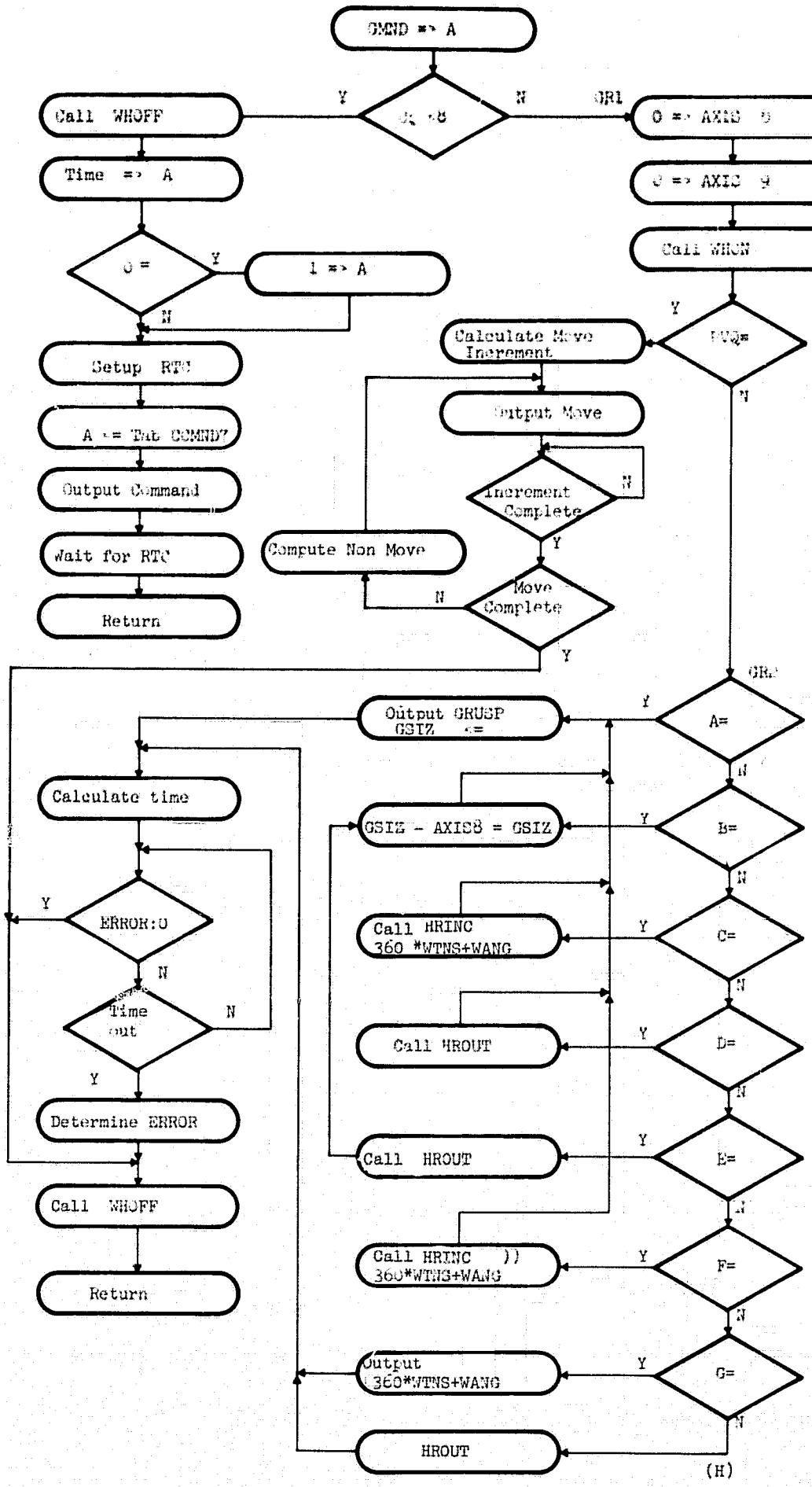


Figure 3-5 GRIP Flow Diagram

to specify the distance and direction. The routine will break up the move into Z moves of .1 inches so as not to slide off the nut should the move be completed before the turn is finished. This feature utilizes the fixture constructed to hold extensions and sockets. This fixture is a replacement for the hook or the hand.

3.3.3 Path Control - - Straight Line

The path control algorithm has demonstrated the relative ease with which the computer can control path type operations. This is compared with the extreme difficulty encountered by the operator in following a specific path when the path is not aligned with the motion of one of the axes.

The algorithm implemented permits straight line motion along a line defined by the extension of the centerline of the hand. This routine requires that the operator only specify the distance to be moved. The direction is computed based on the current values for the SR and WP angles. The TTY Exec requests the distance through the routine "IDIST" and then calls VECTOR to make the requested move. Punch calls IDIST to obtain the distance which is executed by AUTOEX when it calls VECTOR.

Figure 3-6 gives the relationships between the various axes and the orientation which are used to obtain:

$$\Delta X = D \sin WP \cos SR$$

$$\Delta Y = D \sin WP \sin SR$$

$$\Delta Z = -D \cos WP \qquad (1)$$

A straight line path is formed by dividing ΔX, ΔY, and ΔZ into an equal number of relatively fine increments. As indicated

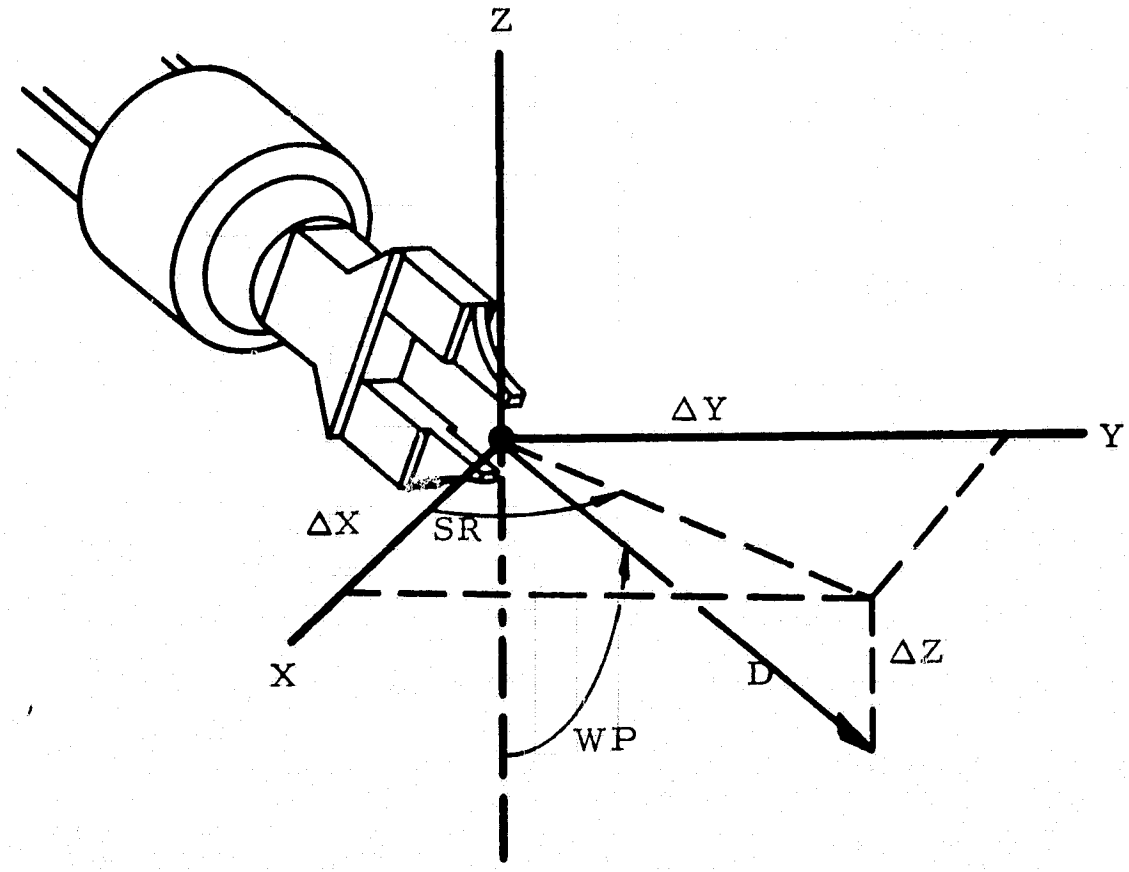


Figure 3-6. Vector and Hand Orientation Relationships [6]

by the flow diagram in Figure 3-8 these increments are then added to the current manipulator position at a constant rate. The resultant path will be a good approximation of a straight line if the increments are sufficiently small. In addition the rate of motion is limited only by the maximum speed of the motors. Figure 3-7 shows the flow for IDIST.

To manually generate a straight line, the rates RX,RY, and RZ must be adjusted in such a way that the equation

$$\frac{\Delta X}{RX} = \frac{\Delta Y}{RY} = \frac{\Delta Z}{RZ} \qquad (2)$$

is satisfied. Thus the operator must control the rates of these axes simultaneously. This is almost impossible to do with any degree of accuracy, especially if the distance involved is of any consequence.

A master-slave manipulator is better for the generation of paths. However the quality of the path produced would be a function of the dexterity of the operator. In addition this type of operation has a very high fatigue factor and the operator's efficiency would degrade rather rapidly.[7]

3.3.4 Hand Position Control

The Hand Position Control algorithm most effectively demonstrates the ability of the computer to perform a complex operation with only one command and a minimum of accompanying data. The operator must specify only five variables which indicate the desired hand position (FHX,FHY, and FHZ) and orientation (Azimuth, Elevation).

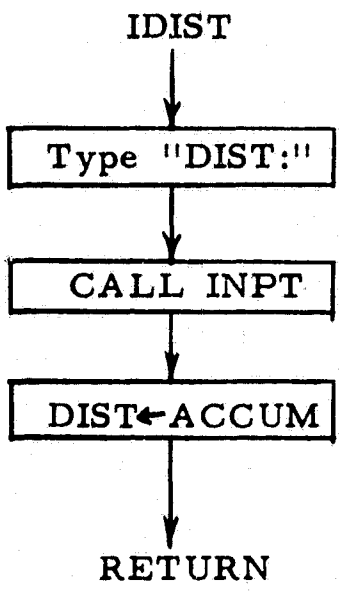


Figure 3-7. IDIST Flow Diagram[8]

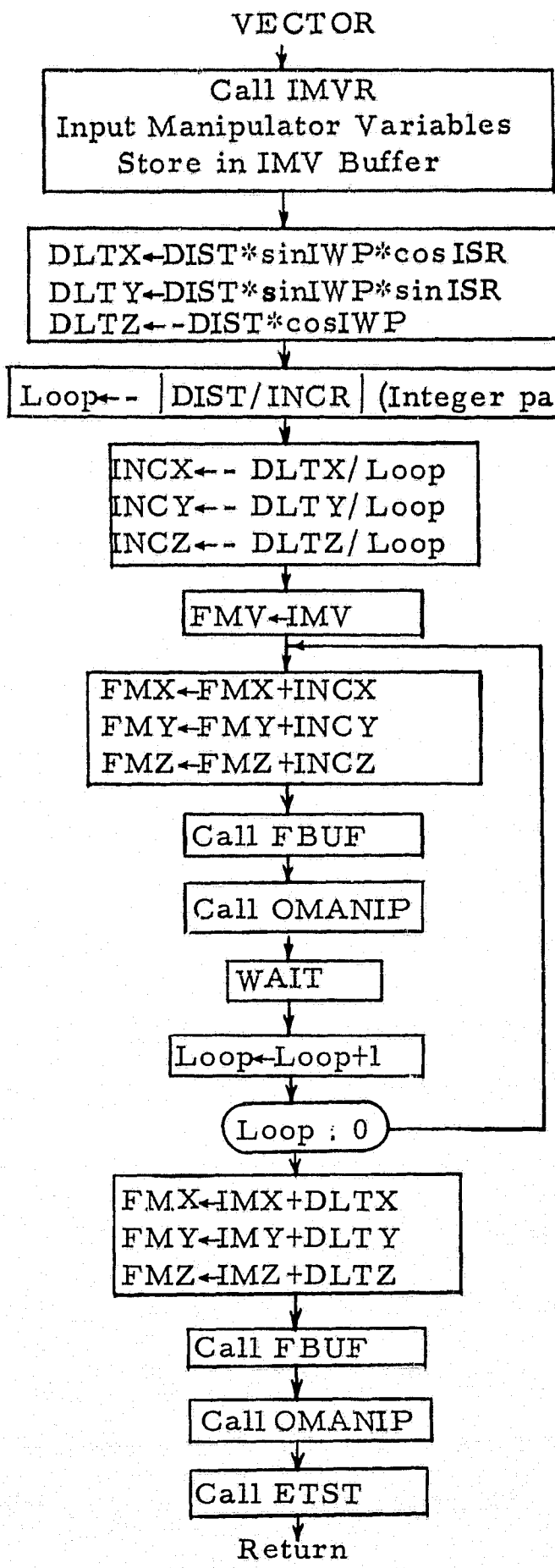


Figure 3-8. VECTOR Flow Diagram

A terminal manipulator position is computed which satisfied the five specified variables. Generally, it is the one requiring the least transit time; however, the projected path is tested by aspecial obstacle avoidance algorithm to insure that the manipulator will not collide with any predefined obstacles. If a potential collision is detected, a new configuration is specified or, if necessary, a set of intermediate positions are automatically selected to avert the collision.

The flow diagram of the "HOPOUT" (Hand Optimization, Obstacle Avoidance, and Output Routine) is shown in Figure 3-9.

3.3.4.1 Buffer Initialization

Three buffers must be initialized before calling the "HOPOUT" subroutine. These indicate the present manipulator variables (IMV), the present hand variable (IHV), and the desired hand variables (FHV).

The AUTO Exec reads the desired hand variables from tape stores them in the FHV buffer, then calls "CIHV" to fill the IMV and IHV buffers.

The TTY Exec allows two input modes. The "RTHAND" subroutine, Figure 3-10, requests all variables, accepts the terminal positions, and stores them in the FHV buffer. "CIHV" is called to fill the IMV and IHV buffers. The "ITHAND" subroutine is nearly identical to "ITMNIP" and is shown in Figure 3-11. This allows the operator to specify a partial set of variables in either an absolute or an incremental mode and fills the three buffers.

The "RELHND" subroutine described in Section 3.4.3 also calls the "HOPOUT" subroutine, but fills the buffers before calling it.

3.3.4.2 Optimization Algorithm

During the execution of many tasks, the operator is primarily concerned about the position and orientation of the hand while the exact manipulator configuration is relatively unimportant. As a result, the following algorithm requires only the five hand variables (HX, HY, HZ, Azimuth, and Evaluation).* From Figure 3-12 these

---

*Note that the rotation is not under closed loop control [for these routines].

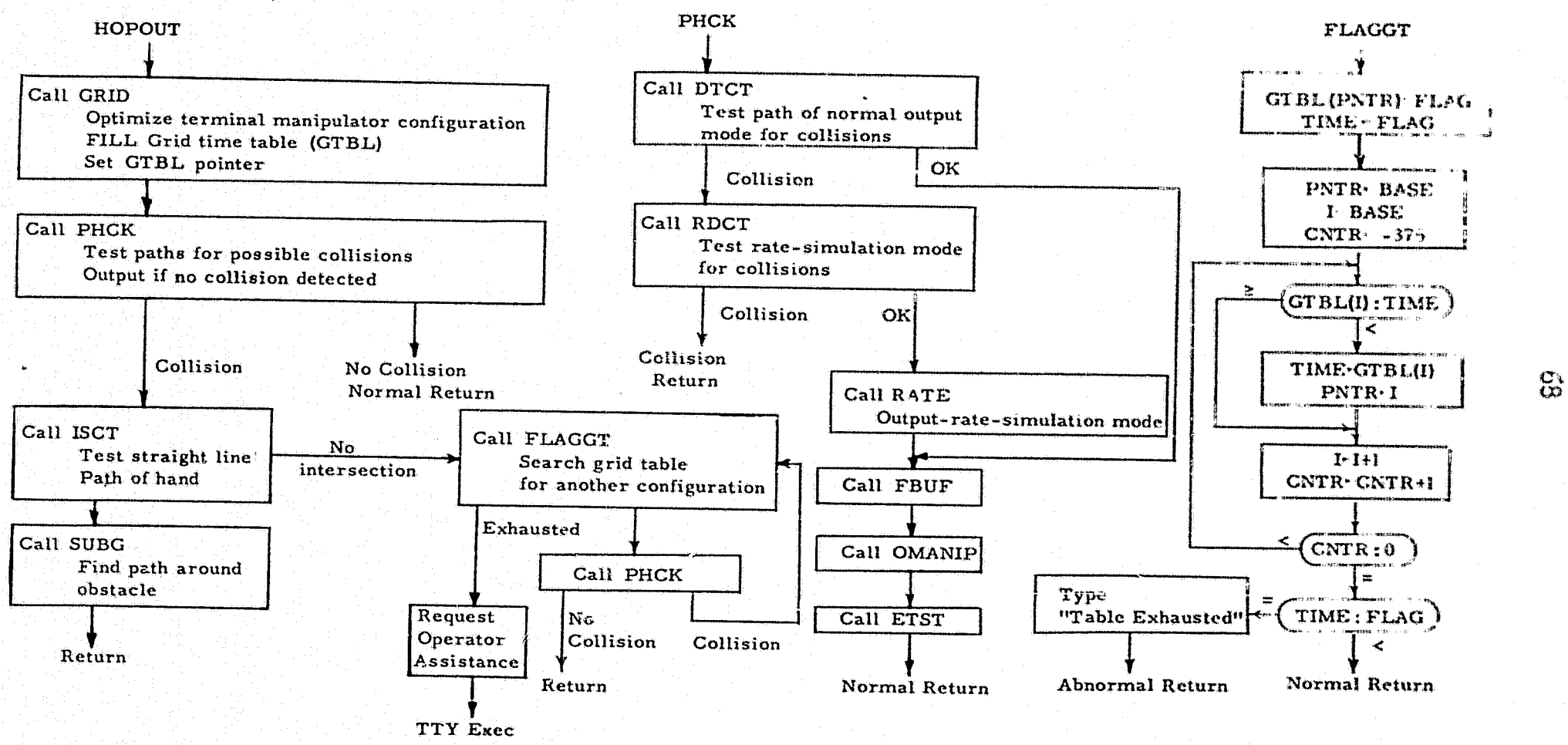


Figure 3-9. HOPOUT, PHCK, and FLAGGT Flow Diagrams

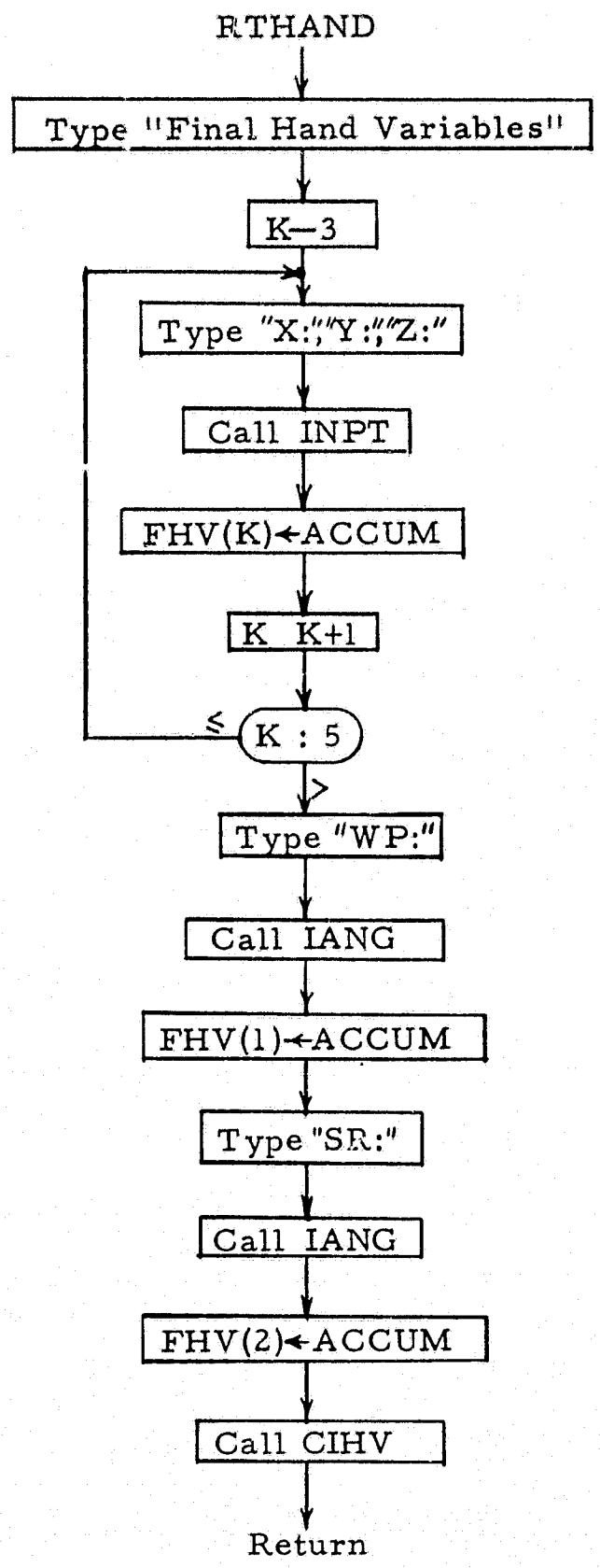


Figure 3-10. RTHAND Flow Diagram

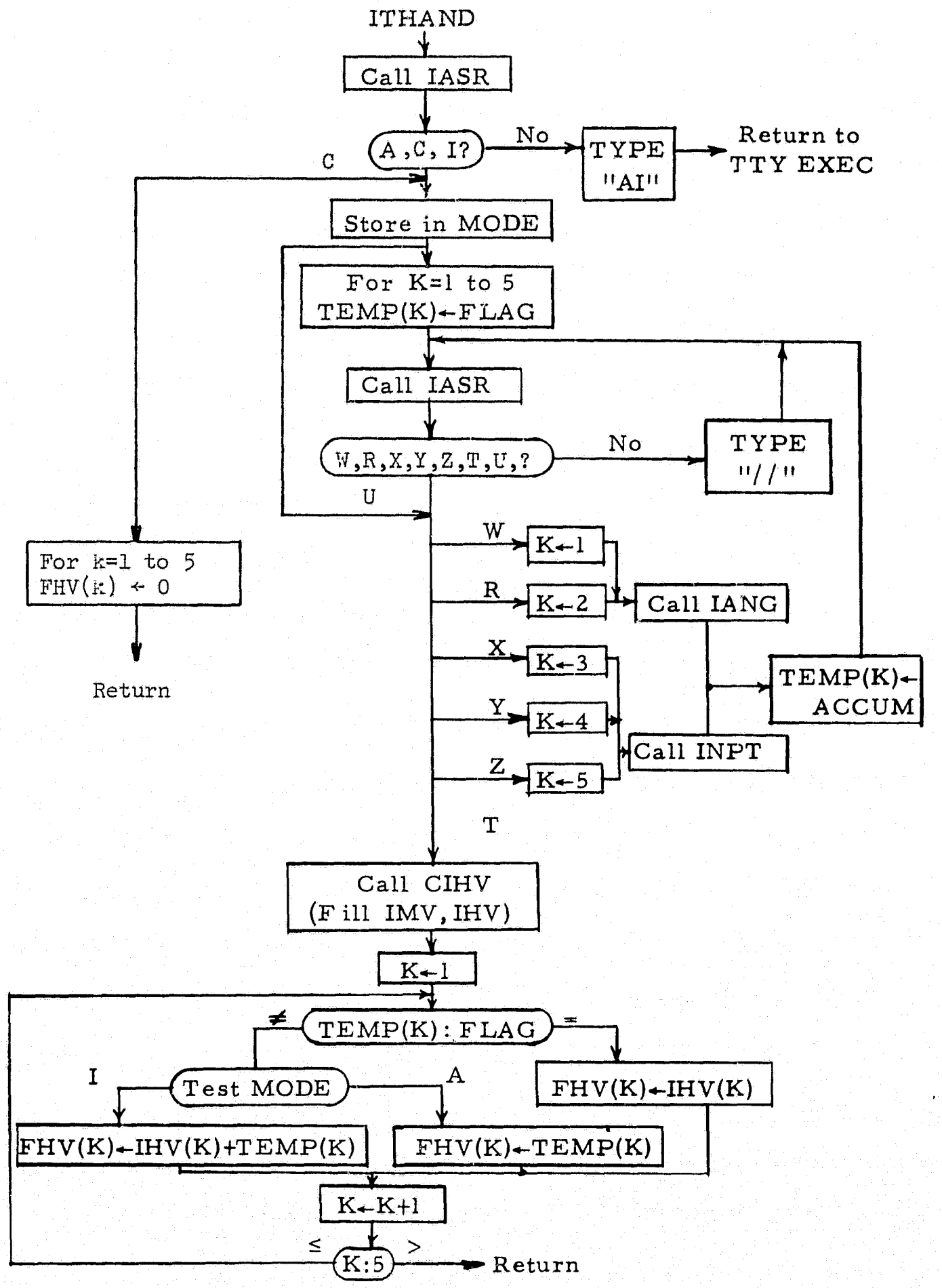


Figure 3-11. ITHAND Flow Diagram

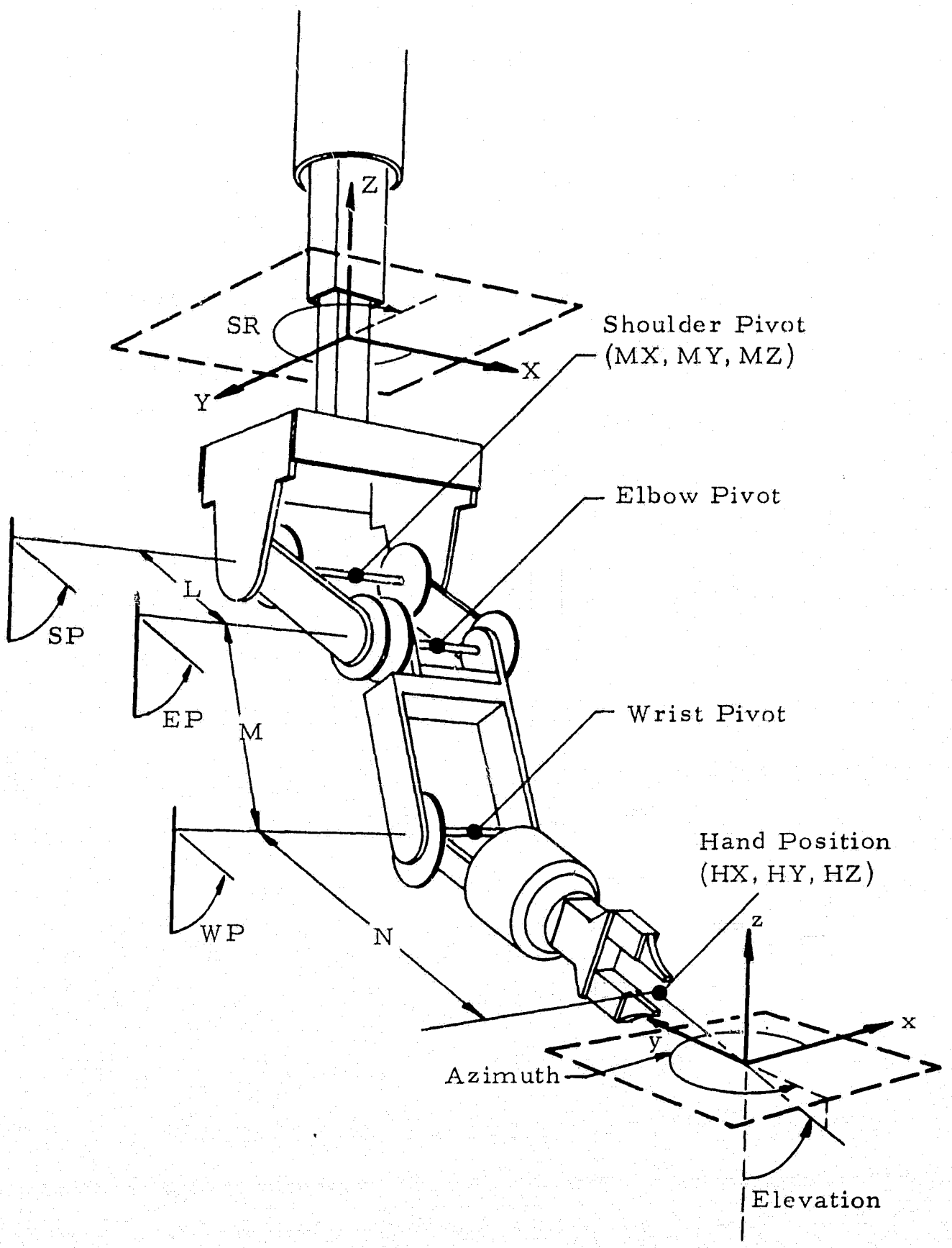


Figure 3-12. Manipulator and Hand Variable Relationships

are related to the seven manipulator variables by

$$MX = HX-(L \sin SP + M \sin EP + N \sin WP) \cos SR$$

$$MY = HY-(L \sin SP + M \sin EP + N \sin WP) \sin SR$$

$$MZ = HZ + L \cos SP + M \cos EP + N \cos WP \qquad (3)$$

and

$$WP = \text{Elevation}$$

$$SR = \text{Azimuth} \qquad (4)$$

where the prefixes M and H refer to the manipulator and the hand variables respectively and where L,M, and N are the lengths of the upper arm, forearm, and hand respectively.

The task of the algorithm is to compute a set of terminal manipulator variables which satisfy Equation (3). The wrist pivot and shoulder rotation angles are exclusively related to the Elevation and Azimuth respectively. This, in effect, reduces the problem to the specification of two independent variables belonging to the set MX, MY, MZ, EP, and SP and then using Equation (3) to compute the remaining three variables.

In order to enhance the performance of the system, it is possible to couple the specification of the two independent variables to a set of optimizing criteria (e.g., minimum time, minimum momentum, minimum energy, etc.). The algorithm implemented in the experimental system is based upon minimum time considerations. Figure 3-13 shows a few of the manipulator configurations which can be assumed for a specified set of hand variables and the time required to reach each configuration from the initial position. Note that (c) is the minimum time solution computed by the algorithm. In effect, the principal concern of the algorithm is to find values for SP and EP such that the resulting ΔMX, ΔMY, or ΔMZ variable which requires the longest transit time is reduced to the point that it is equal to or exceeded by the transit time of another axis.

Since it is not imperative to compute the absolute minimum time solution, two basic simplifying assumptions have been made. The first one is that the manipulator motion is so slow that the dynamics of the system can be neglected. The second is that the motors drive the axes at a constant rate. Of the two, the second will produce the greatest error. This is due to the fact that series- universal motors

	ΔX	ΔZ	ΔEP	ΔSP	Time
a	0	0	75	165	20
b	0	10	105	135	16
*c	15	7	60	75	10
d	25	0	15	45	15
e	30	-10	0	0	19

*Minimum time solution

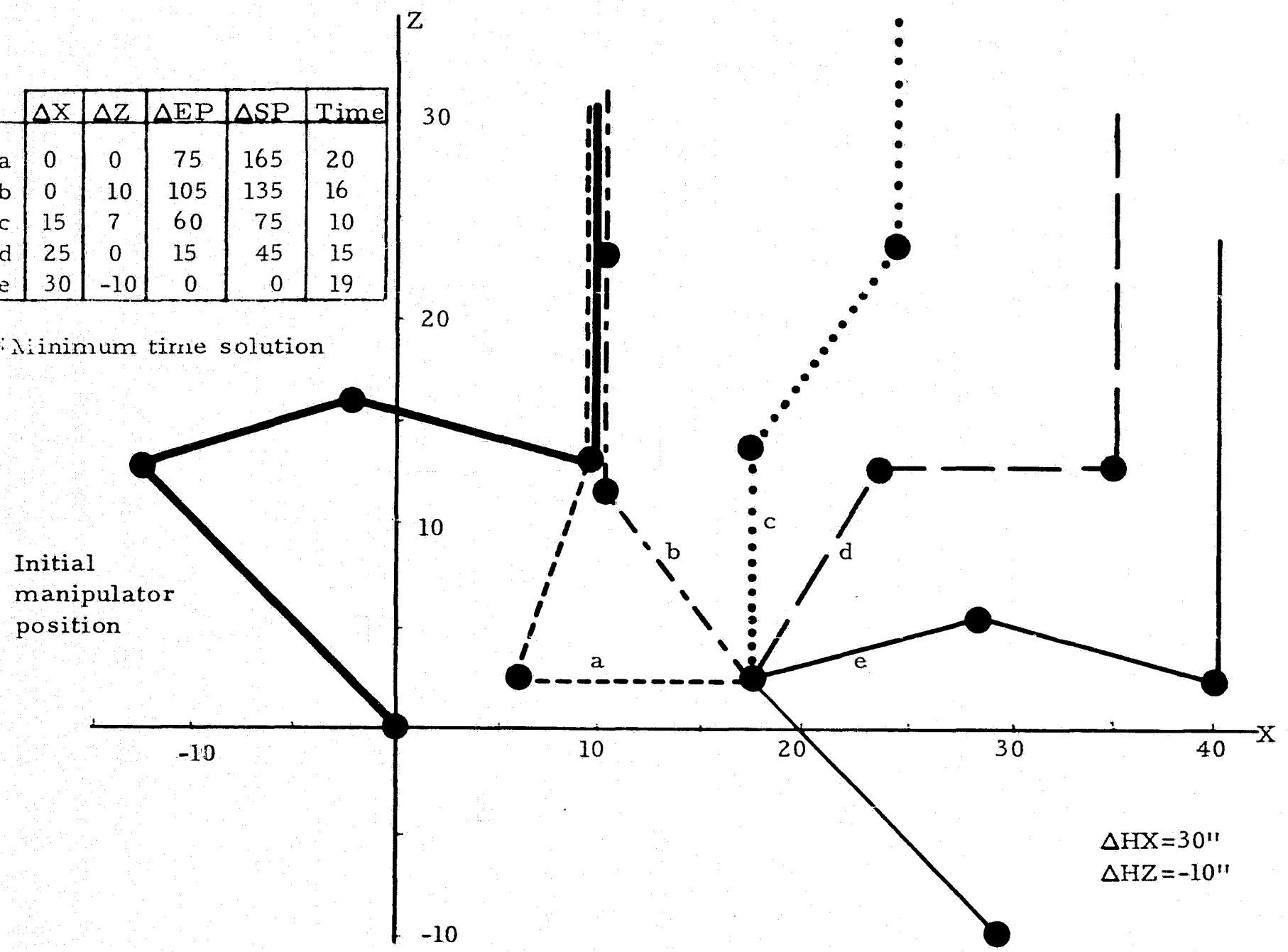


Figure 3-13. Possible manipulator configurations for a specified change in the hand position

are used to drive the manipulator and that their speed is strongly dependent upon the load. Experimentally, this assumption has produced no adverse effects; however, if necessary, it would be possible to modify the algorithm to select one of several rates for each axis dependent upon the load, the initial manipulator position.

Based upon the two assumptions, the time required by each axis of the manipulator is given by

$$\begin{aligned} TMX &\quad |(FMX-IMX)/RX| \\ TMY &\quad |(FMY-IMY)/RY| \\ TMZ &\quad |(FMZ-IMZ)/RZ| \end{aligned} \tag{5}$$

and

$$\begin{aligned} TSP &\quad |(FSP-ISP)/RSP| \\ TEP &\quad |(FEP-IEP)/REP| \end{aligned} \tag{6}$$

where the prefixes T, R, F, and I denote time, rate, final values, and initial values respectively. The WP and SR times are not considered since the terminal values of these angles are unique.

From Equation (5), Equation (3) can be rewritten as

$$\begin{aligned} TMX &\quad | K_0-K_3(L \sin FSP + M \sin FEP)| \\ TMY &\quad | K_1-K_4(L \sin FSP + M \sin FEP)| \\ TMZ &\quad | K_2+K_5(L \cos FSP + M \cos FEP)| \end{aligned} \tag{7}$$

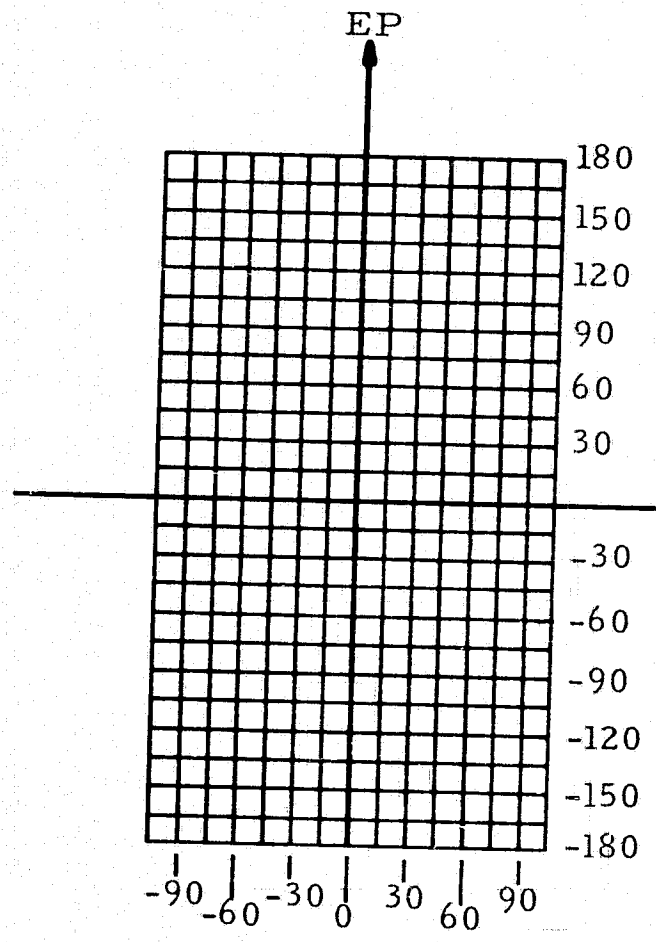
where

$$\begin{aligned} K_0 &= (FHX - IMX - N \sin FWP \cos FSR)/RX \\ K_1 &= (FHY - IMY - N \sin FWP \sin FSR)/RY \\ K_2 &= (FHZ - IMZ + N \cos FWP)/RZ \\ K_3 &= (\cos FSR)/RX \\ K_4 &= (\sin FSR)/RY \\ K_5 &= 1/RZ \end{aligned} \tag{8}$$

The optimization algorithm utilizes a grid technique to find the minimum time solution. This method was chosen because

it avoids problems with local minimum values, is easily implemented, allows pivot and boundary constraint tests, and is compatible with the obstacle avoidance routines described in the next section.

The EP and SP angles were selected as the two independent variables since they already appear as functions of MX, MY, and MZ in Equation (3). A grid is formed by restricting the values of EP and SP to an integral multiple of 15 degrees throughout their entire range. The grid for the experimental system is shown in Figure 3-14. For each intersection on the grid, the times required by the X, Y, Z, EP, and SP axes to travel from their initial positions to those which satisfy Equation (3) for the EP, SP values are computed. The greatest of these times represents the time required to reach the terminal point (excluding TWP and TSR). As each of the intersections is considered, the transit time is stored in a "grid" table for future use by the obstacle avoidance routine. The time is also compared with the smallest time previously encountered. If equal or larger, it is ignored; otherwise, its value and the grid intersection pointer are preserved. The combination which remains after all intersections have been tested is then chosen as the "minimum time" solution.[11]



SP, EP Grid Intersections

SP	EP
-105	-180
	-165
	⋮
	+165
	+180
-90	-180
	⋮
	0
	⋮
	+180
-75	-180
	⋮
	0
	⋮
	+180

Grid Time Table

Figure 3-14. Example of SP, EP Grid and the Grid Time Table [12]

During the computation and selection sequence, three special constraint tests are performed. The first test checks the associated wrist pivot constraint table word for an operator specified constraint (see Section 3.3.5.1). This constraint can be applied via the teletype or from paper tape when in the Automatic Mode. Next the Wrist Pivot constraint table is examined to determine whether the position is physically within the range of the manipulator. That is, will the manipulator be hitting itself if this move were to be made. The third test is made to determine whether the endpoint can be reached for a given arm configuration.

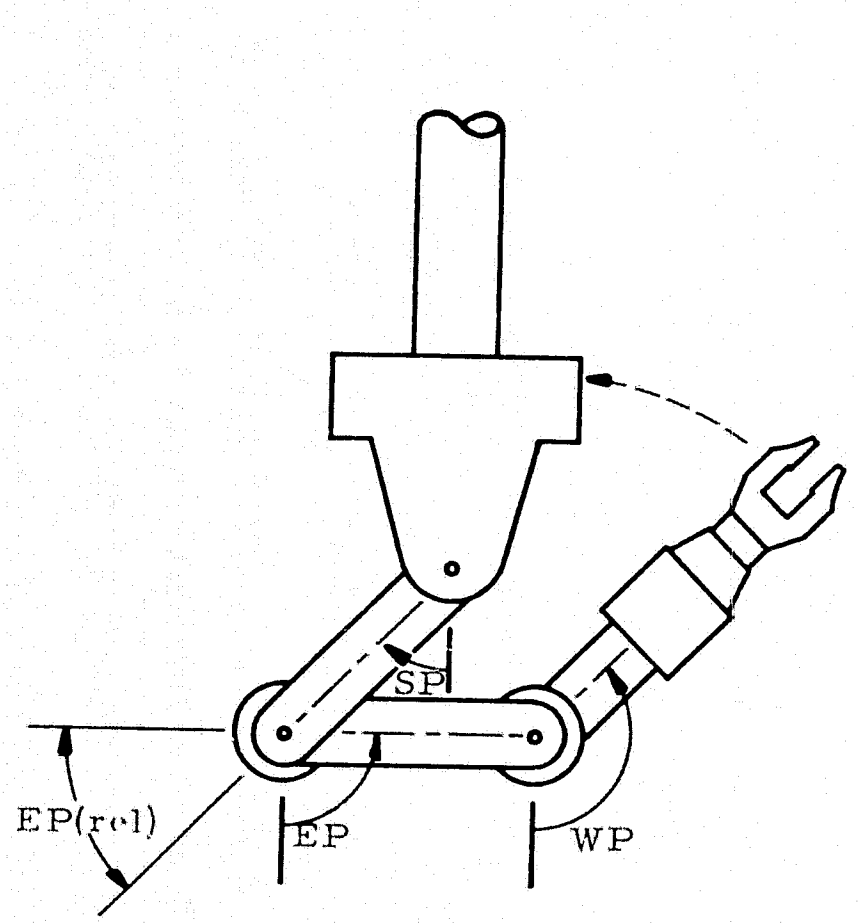
Figure 3-15 gives two examples of the types of constraints which must be applied in the second case. The table used for these constraints is prepared in advance and is loaded along with the main manipulator control tables.[13]

Figures 3-16 and 3-18 demonstrate several terminal manipulator configurations computed by the optimization routine for various increments in the X and Z hand positions respectively. In addition the transit times required by the manipulator are listed and are compared with the times which would be required if EP and SP were not incremented.

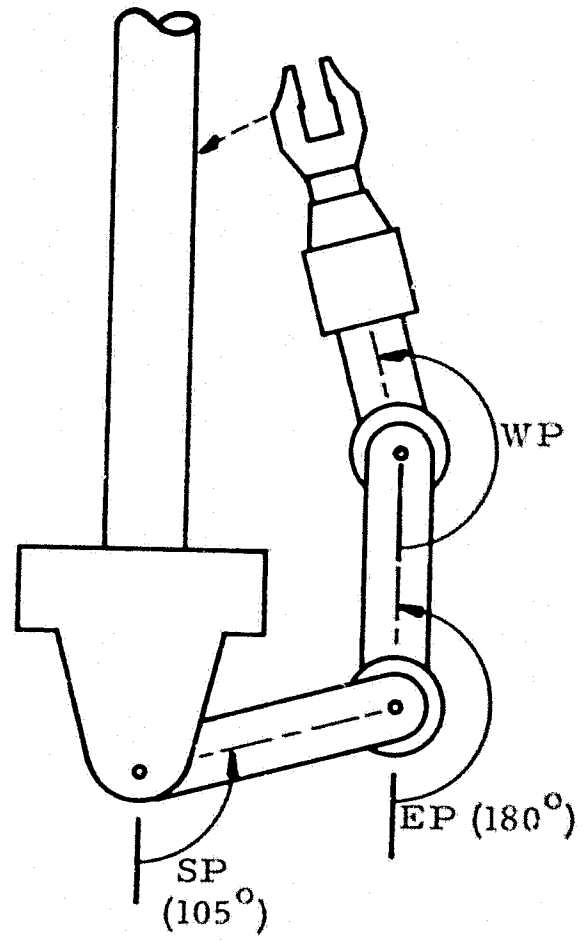
Figures 3-17 and 3-19 show the paths followed by each pivot and the hand position for the examples shown in Figures 3-16 and 3-18. Note that there is an abrupt change in the paths as each axis reaches the terminal positions.

Before calling the optimization routine, it is necessary to fill the IMV, IHV, and FHV buffers. In addition, the following buffers are loaded into computer's memory with the optimization routine:

1. The constraint table.
2. The X,Y,Z manipulator boundary table.



a. EP (rel) at limit
WP must be constrained to prevent hand from hitting motor



b. SP and EP at absolute limits
WP must be constrained to prevent hand from hitting tube

Figure 3-15. Examples of SP , EP , and WP constraints

	ΔX	TΔHX	TΔMX
a	-30	7	19
b	-40	10	25
c	-50	16	31

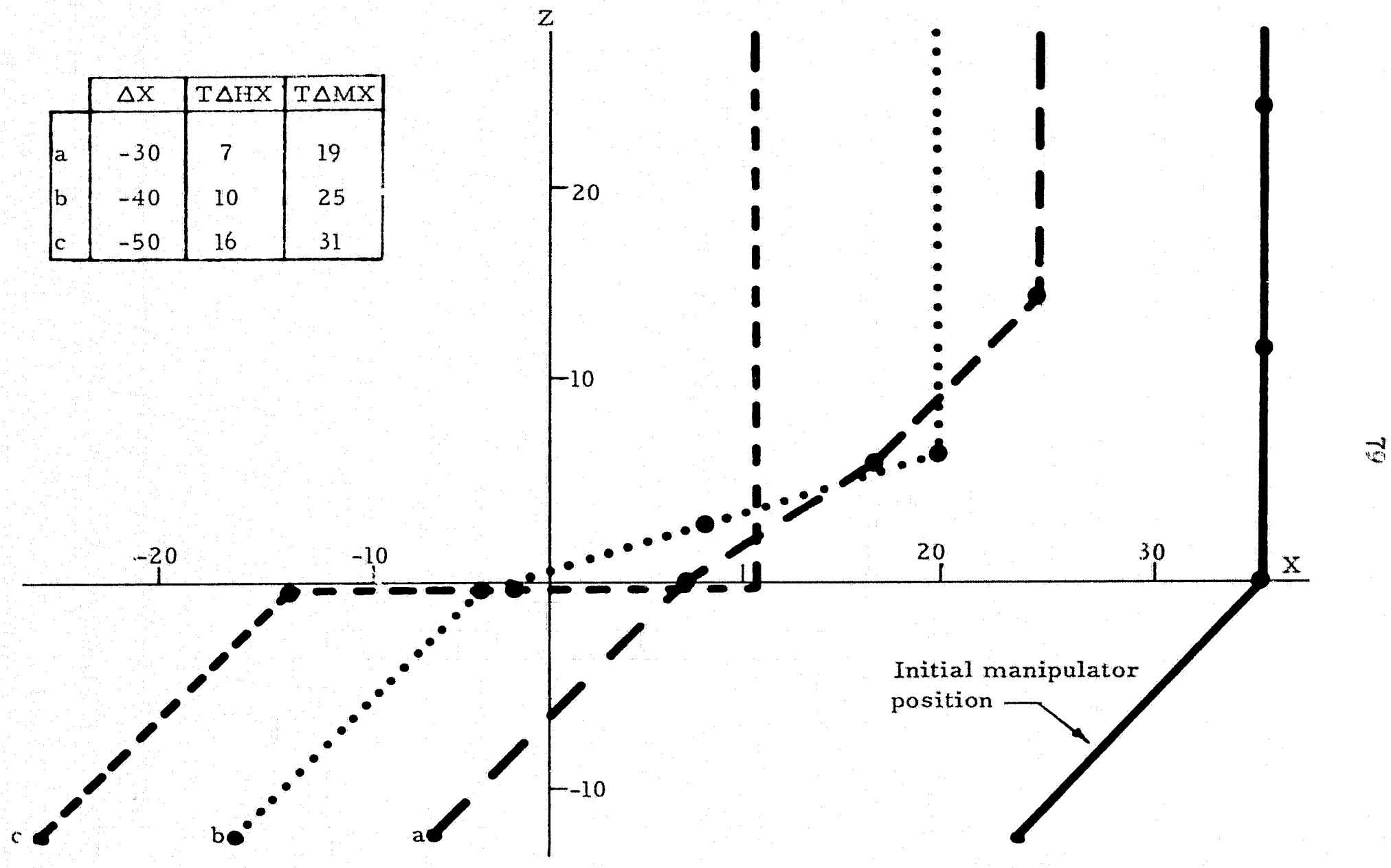
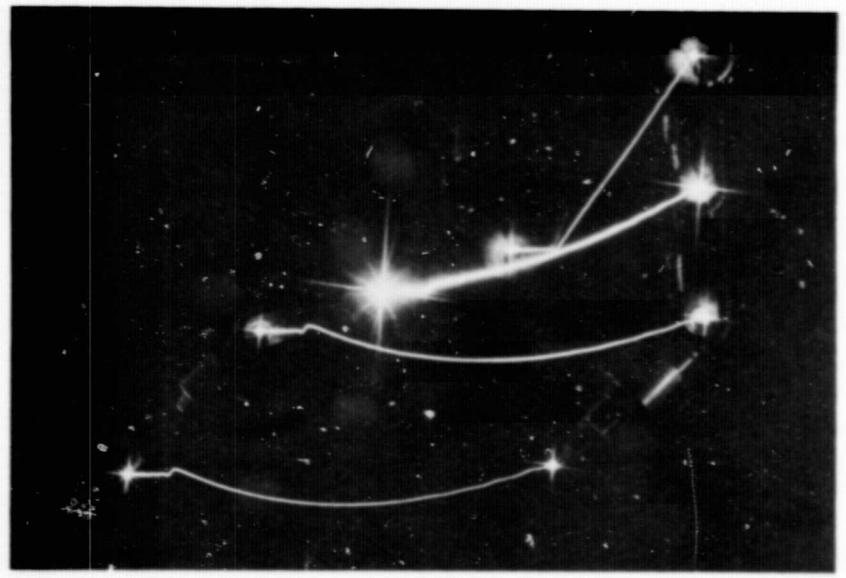


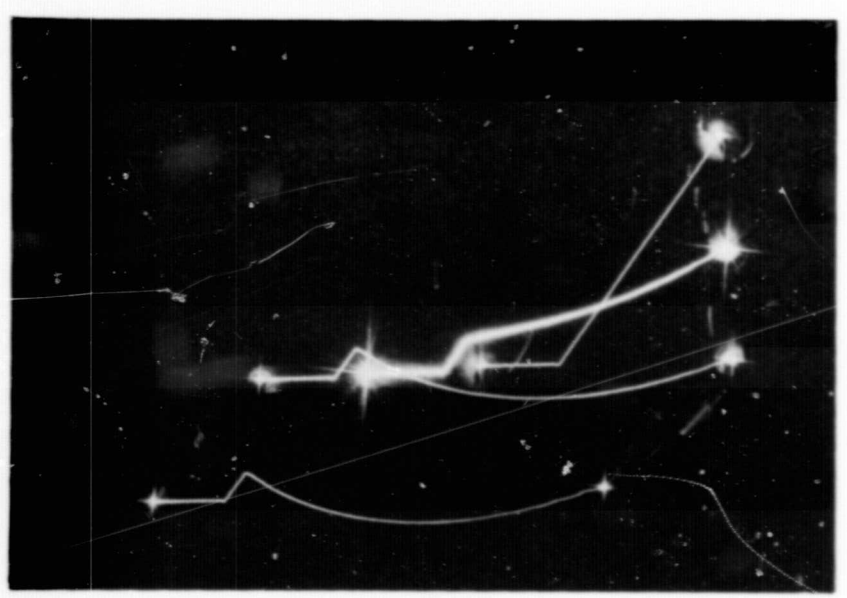
Figure 3-16. Optimized manipulator configurations for HX increments



a. ΔHX=−30

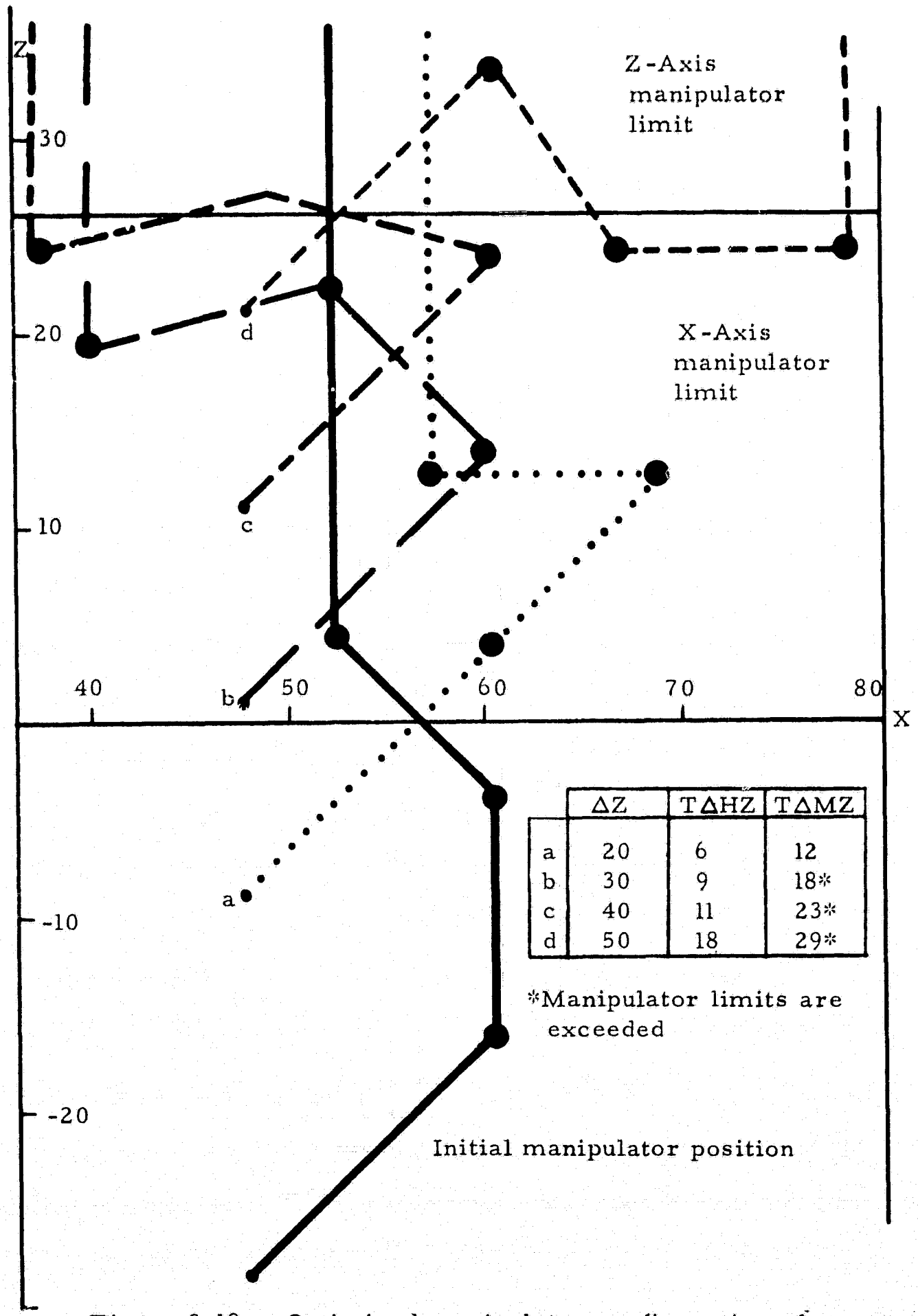


b. ΔHX=−40



c. ΔHX=−50

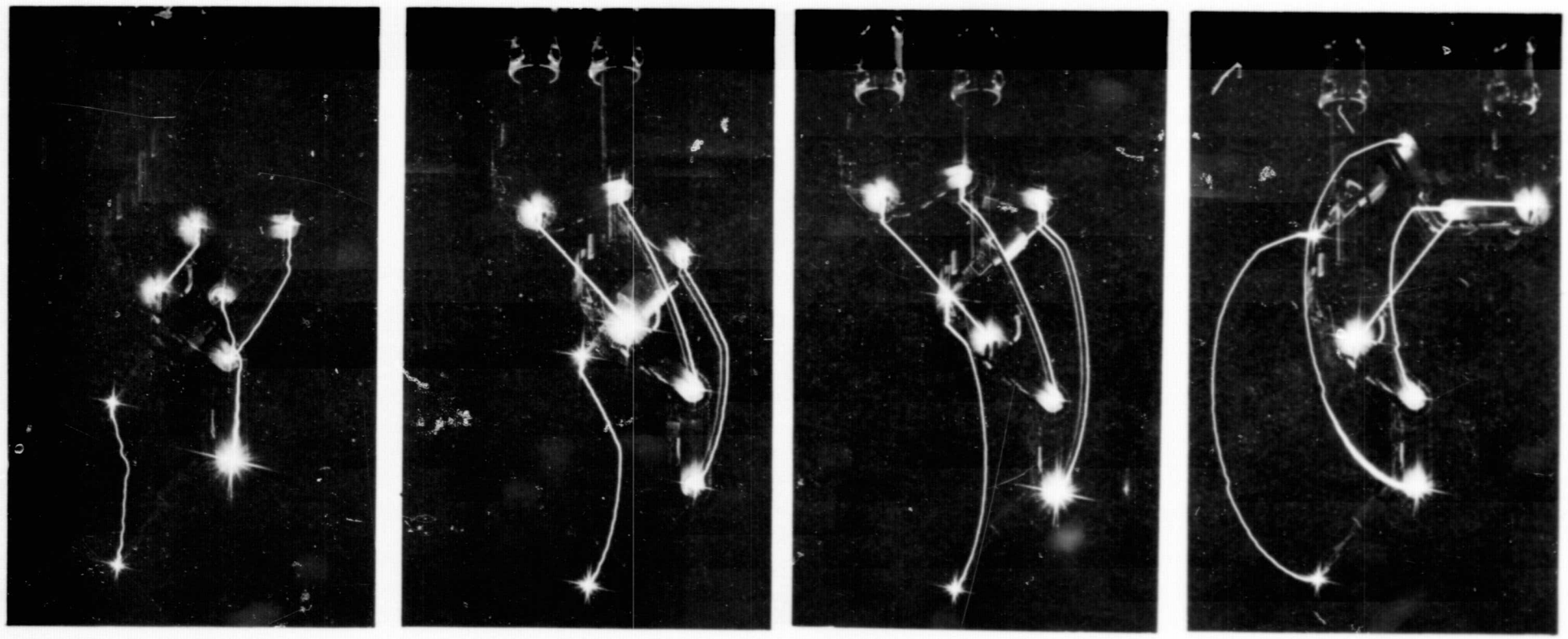
FIGURE 3-17. PATHS OF MANIPULATOR FOR EXAMPLES IN FIGURE 3-16



	ΔZ	TΔHZ	TΔMZ
a	20	6	12
b	30	9	18*
c	40	11	23*
d	50	18	29*

*Manipulator limits are exceeded

Figure 3-18. Optimized manipulator configurations for HZ increments



a. ΔHZ=20 b. ΔHZ=30 c. ΔHZ=40 d. ΔHZ=50

FIGURE 3-19. PATHS OF MANIPULATOR FOR EXAMPLES IN FIGURE 3-18

3, Tables containing the values of L sin FSP, L cos FSP, M sin FEP, M cos FEP (for allowed values of FSP, FEP).

The sine and cosine terms were determined in advance to reduce the computation time required by the algorithm (approximately 80 msec).

The algorithm fills the grid time table, the grid intersection pointer, the FMV buffer, and the transit time, TMAX.

The flow diagram of the optimization algorithm is shown in Figure 3-30 and the individual subroutines appear in Figure 3-21.

3.3.4.3 Obstacle Avoidance

One of the major tasks which was deferred until the completion of the initial phase of the experimental system was the inclusion of a feedback path to inform the computer when the manipulator was within close proximity to an obstacle. Without this feedback, the operator was required to carefully observe the manipulator's motions when under computer control to insure that no part of the manipulator would collide with any obstacles. Unfortunately, this task was considerably more difficult than when using the manual control mode for two reasons. First, with computer control, the seven independent servosystems drive all motors simultaneously and at their maximum rate until the desired coordinates are reached -- manually, this is rarely done. Second, the operator did not always know the terminal configuration of the manipulator and even when it was known, the exact path of the manipulator could not always be accurately visualized in advance; hence, a number of collisions occurred before the operator was aware of the danger. For example, Figure 3-22a demonstrates the path of the manipulator when the hand position optimization algorithm was directed to move the hand to a location 40 inches to the left of the initial position. Note that the table would have been hit if it had been located directly under the hand even though both the initial and terminal manipulator configurations would be acceptable.

The collision detection subroutines were formulated to recognize potential hazards such as that described above. In addition, a subroutine was formulated which tries to automatically select an acceptable path over or around the obstacle. Figure 3-22b demonstrates the path generated to prevent the manipulator from striking the table.

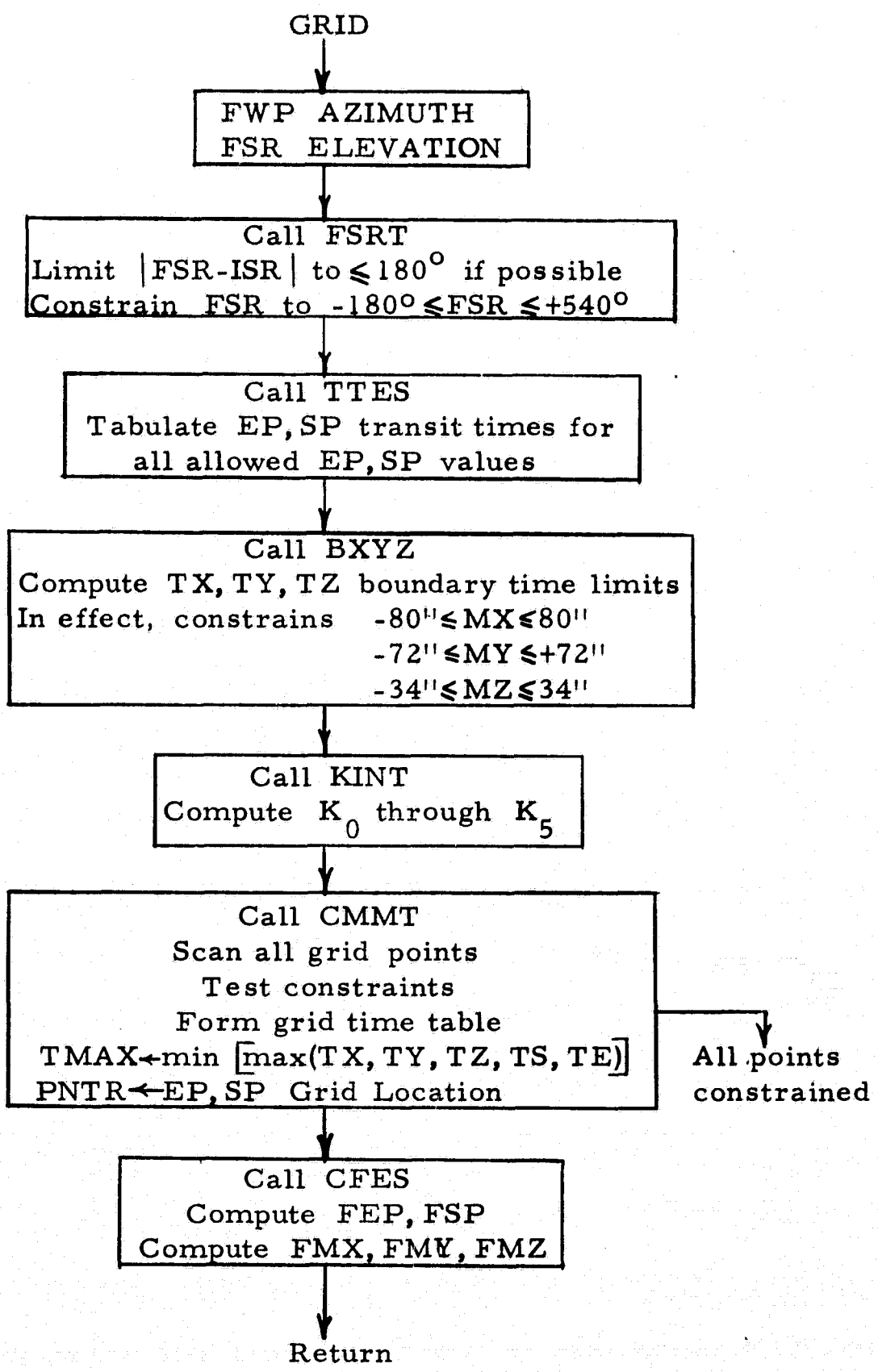


Figure 3-20. GRID Flow Diagram

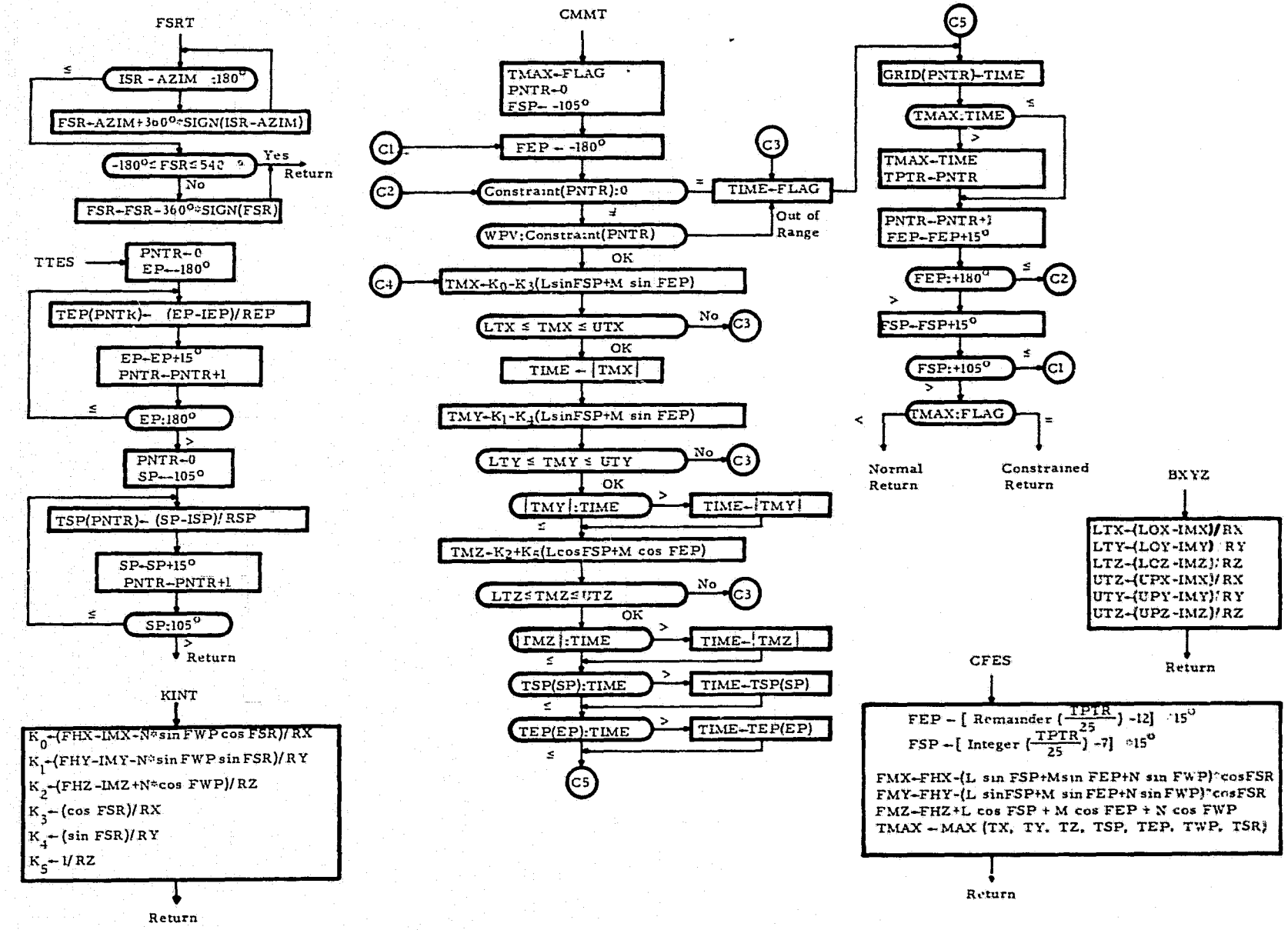
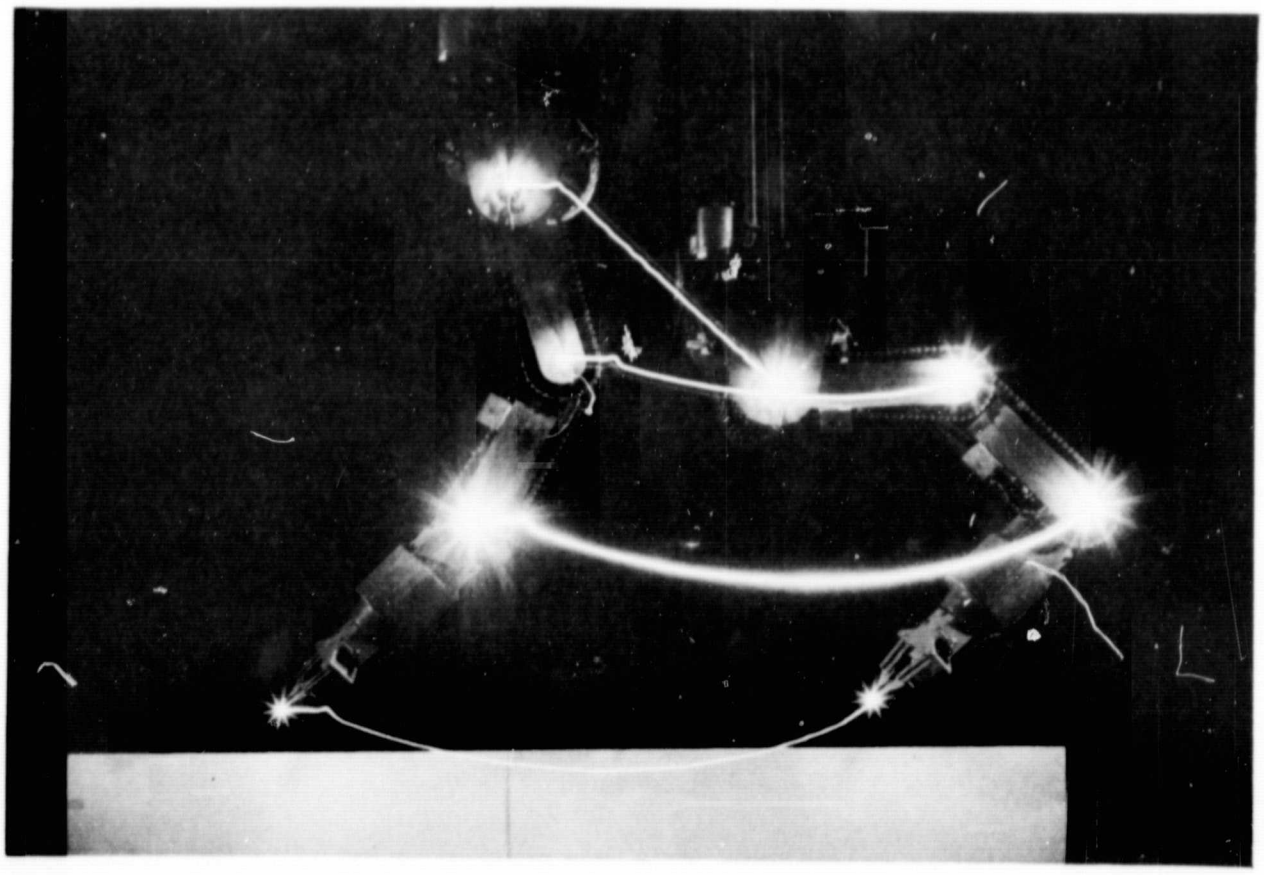
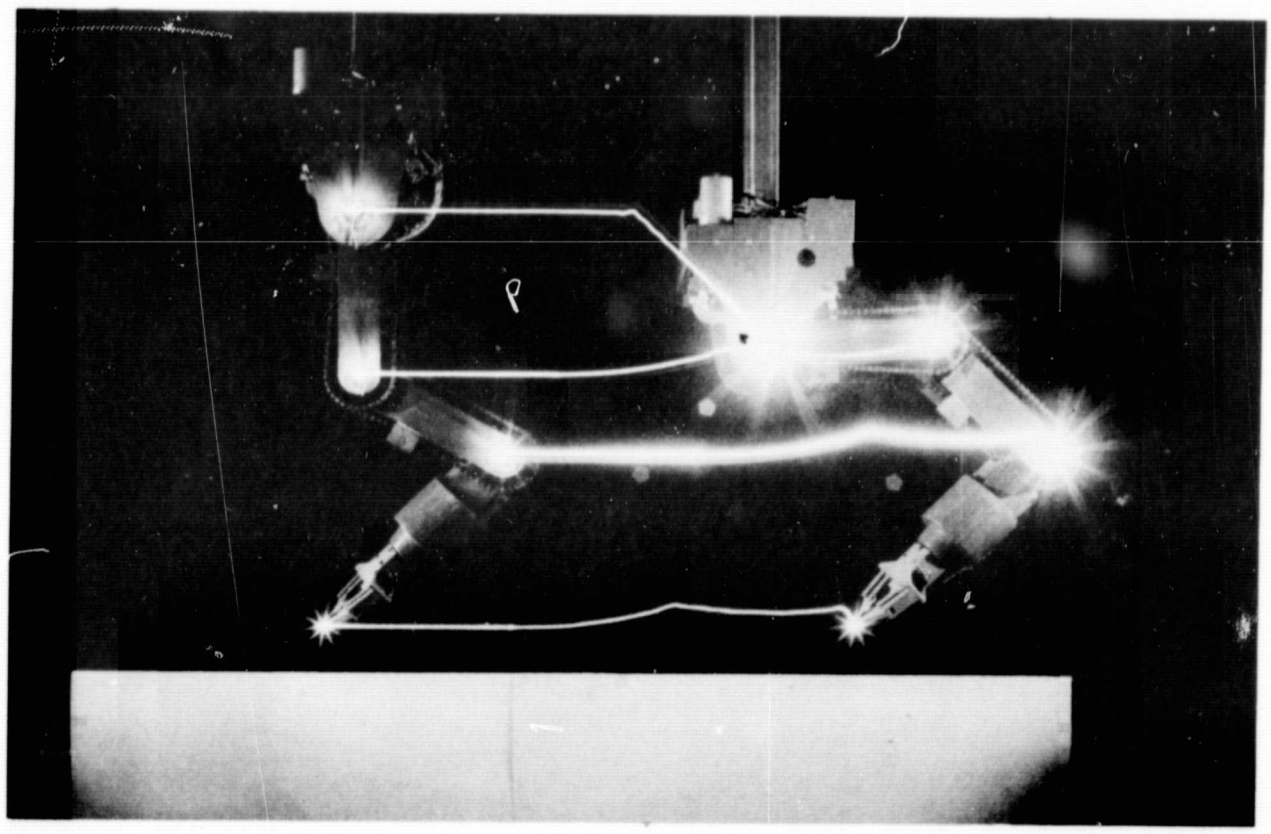


Figure 3-21. Flow Diagrams of Subroutines Called by GRID



a. First path generated by Optimization routine



b. Path allowed by Obstacle Avoidance routine

Figure 3-22. Example of a possible collision between the manipulator and an obstacle

In order to facilitate the development of the subroutines, the following assumptions were made:

1. All obstacles have the same shape and orientation (see next section).
2. All obstacles remain stationary (due to lack of sensory feedback).
3. The table accommodates only four obstacles (but can easily be enlarged).
4. All obstacle dimensions specified by the operator are enlarged (see next section).
5. Additional operator assistance is required if more than one obstacle is encountered en route to the specified destination (see Section 3.3.4.3.3)
6. The routines are called only by the Hand Position Control algorithm (see below).

In spite of the restrictions, the subroutines have increased the value of the computer control technique by a significant amount. The collision detection routine has drastically reduced the number of collisions encountered by the experimental manipulator and the obstacle evasion capability dramatically illustrates how the system performance can be enhanced. In the semi-automatic mode, it reduces the amount of information which the operator must specify to direct the manipulator around an object. In the automatic mode, it allows obstacles to be placed into or removed from the manipulator task area without modifying any pre-recorded data.

Since the obstacle bounds are enlarged, the routines are primarily used to supervise the gross motions of the manipulator and they are called only by the Hand Position Control Algorithm. The remaining path and position control algorithms must be used (cautiously) to work within the obstacle bounds under computer control.

In the future, the experimental system will be modified to provide a feedback path which will inform the computer when the manipulator is in close proximity to an obstacle. A set of routines can then be formulated which allow the manipulator to work within the gross obstacle bounds but which not require detailed operator supervision to avert a collision. These can be coupled with the obstacle avoidance routines presented below to form a comprehensive obstacle detection and avoidance scheme.[14]

The installation of the Maskable Buffer Registers (see Section 2.8.2.1) provides a means by which sensor data can be selectively made

available to the computer. These sensors could be tactile in nature - such devices as limit switches and cats wiskers-or they could be proximity devices. The fitting of these devices has been left for a later effort.

3.3.4.3.1 Specifications of Obstacle Bounds

Due to the lack of sensory feedback from the manipualtor to the computer, it is necessary for the operator to specify all obstacle bounds via the TTY console. The bounds must conform to the following restrictions:

1. The shape must be reduced to a rectangular parallelopiped.
2. The sides extend to the floor.
3. The sides are parallel to the X and Y axes.
4. The effective bounds will be enlarged five to ten inches beyond those specified by the operator.

Figure 3-23 demonstrates how several arbitrary geometrical shapes can be bounded to satisfy the first three restrictions. The second item was included for programming ease but is justified by the fact that the manipulator travels on overhead rails. The third item was included to reduce the programming complexity during the formulation and evaluation of the obstacle avoidance routines.

The last restriction most severely limits the utility of the algorithm but is necessary because of two problems. First, the collision detection scheme considers the projected path of the manipulator. This path is only an approximation and the load-speed characteristics of the motors can cause the actual path to deviate from the simulated one. It would be possible to eliminate this problem by transmitting a series of small path increments to the manipulator in a manner similar to the path control algorithm. Secondly, it is both difficult and time consuming to insure that all parts of the manipulator will not hit the obstacle. In order to minimize the computation time and reduce the complexity of the algorithm, only the positions of the shoulder pivot, elbow pivot, wrist pivot, and hand are compared with a set of "effective obstacle bounds." The effective bounds are the bounds specified by the operator plus five to ten inches to conservatively accommodate the area occupied by the motor platform and the limbs of the arm.

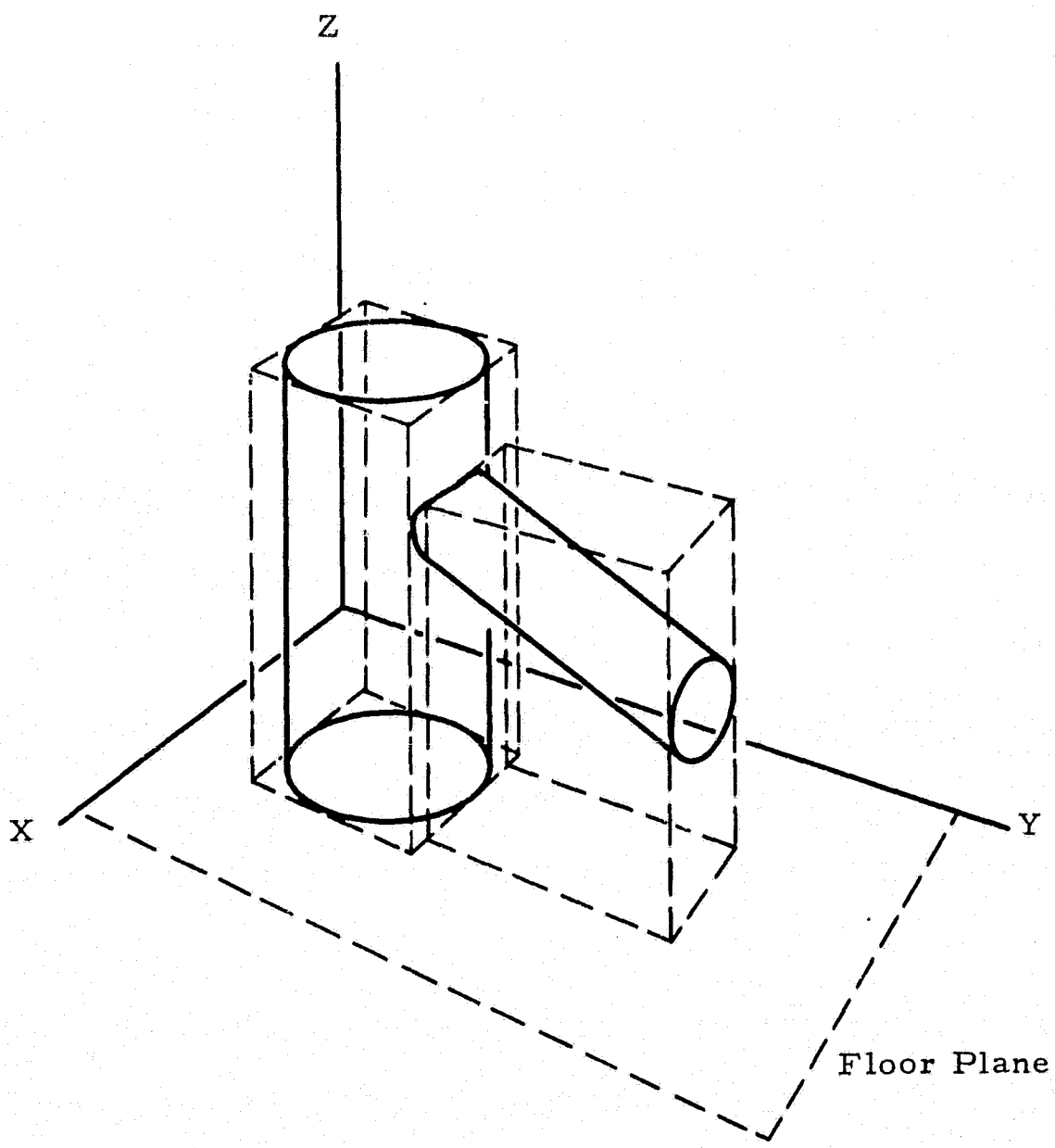


Figure 3-23. Typical obstacle bounds

The "IBOUND" subroutine is called by the TTY Exec and allows the operator to specify the obstacle bounds. As shown in the flow diagram, Figure 3-24, it requests and accepts the X and Y bounds and the top (Z) of the obstacle.

3.3.4.3.2 Path Simulation and Collision Detection

The collision detection subroutines accept the initial and final manipulator configurations and project the path of the three pivot points and the hand. The paths are divided into increments of up to one inch and, at each increment, the positions of the hand and the pivots are compared with the effective obstacle bounds. If any point is within the constrained area, a special "collision" exit is taken to indicate that the FMV values are not acceptable. The manipulator can be driven in both the normal and rate-simulated mode, and since they frequently produce quite different paths, the two collision detection subroutines shown in Figures 3-25 and 3-26 have been written to test each mode.

When an obstacle is encountered, it is possible for these routines to be executed many times for a single command. Thus, in order to minimize the effective computation time when an obstacle is likely to be encountered, the testing of the path is staggered such that the routine first checks the mid-point, next the 1/4 and 3/4 points, then the 1/8, 3/8, 5/8, and 7/8 points etc. until an obstacle is encountered or until the path has been sufficiently segmented to insure that no obstacles will be hit.

3.3.4.3.3 Obstacle Evasion

If a collision is detected, several efforts are made to evade the obstacle without requesting assistance from the operator. First, the straight line connecting the initial and terminal hand positions is tested for an intersection with the X and Y effective bounds of all obstacles. If no intersection is detected, "FLAGGT" is called to search the Grid Time Table for other possible terminal manipulator configurations, then "PHCK" is used to test the corresponding paths for obstacles. The table search continues until the table is exhausted or until a configuration is found which does not result in a collision. If an acceptable set of variables are found, they are used to actuate the manipulator; otherwise, the operator's assistance is requested.

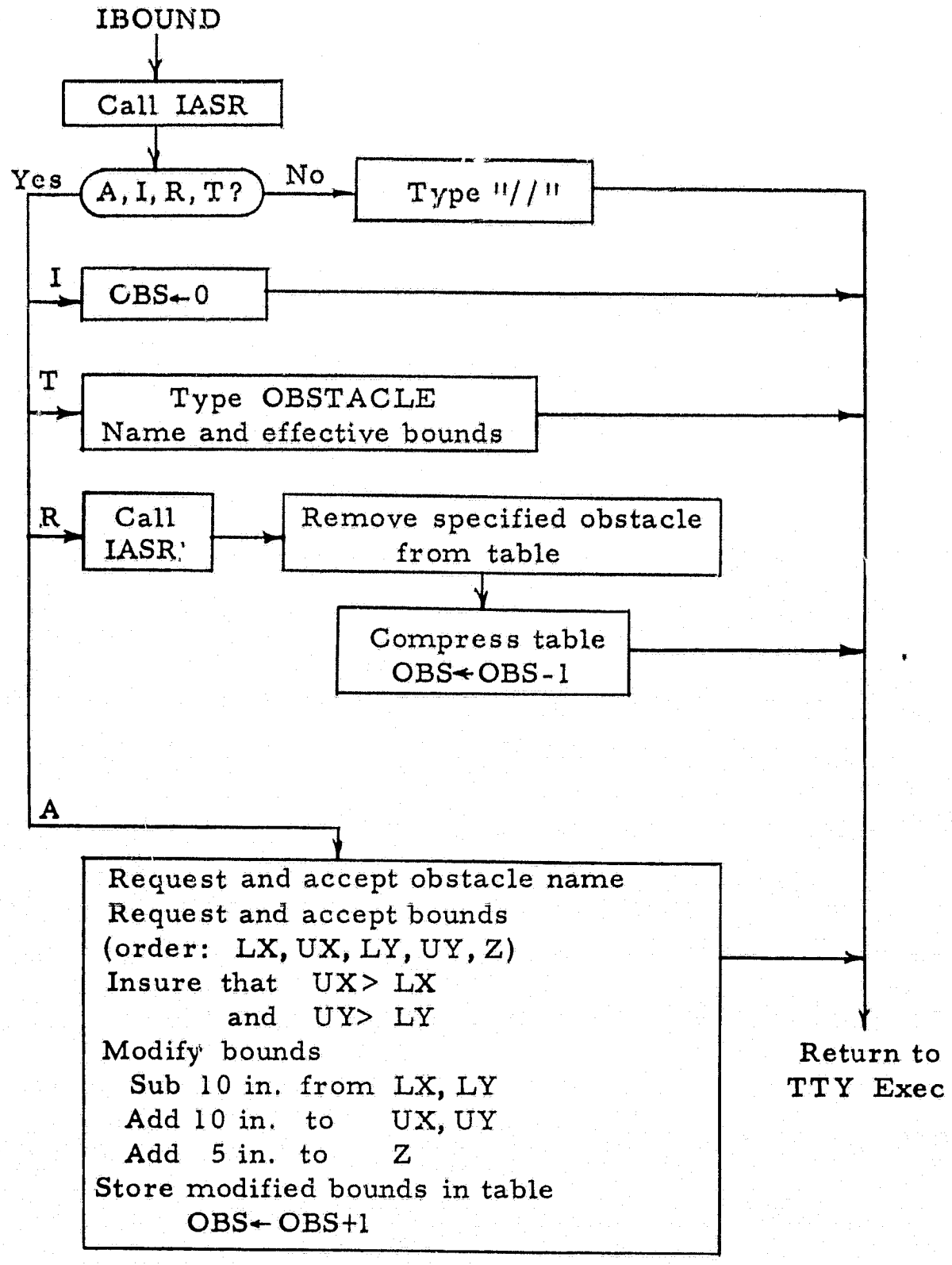


Figure 3-24. IBOUND Flow Diagram

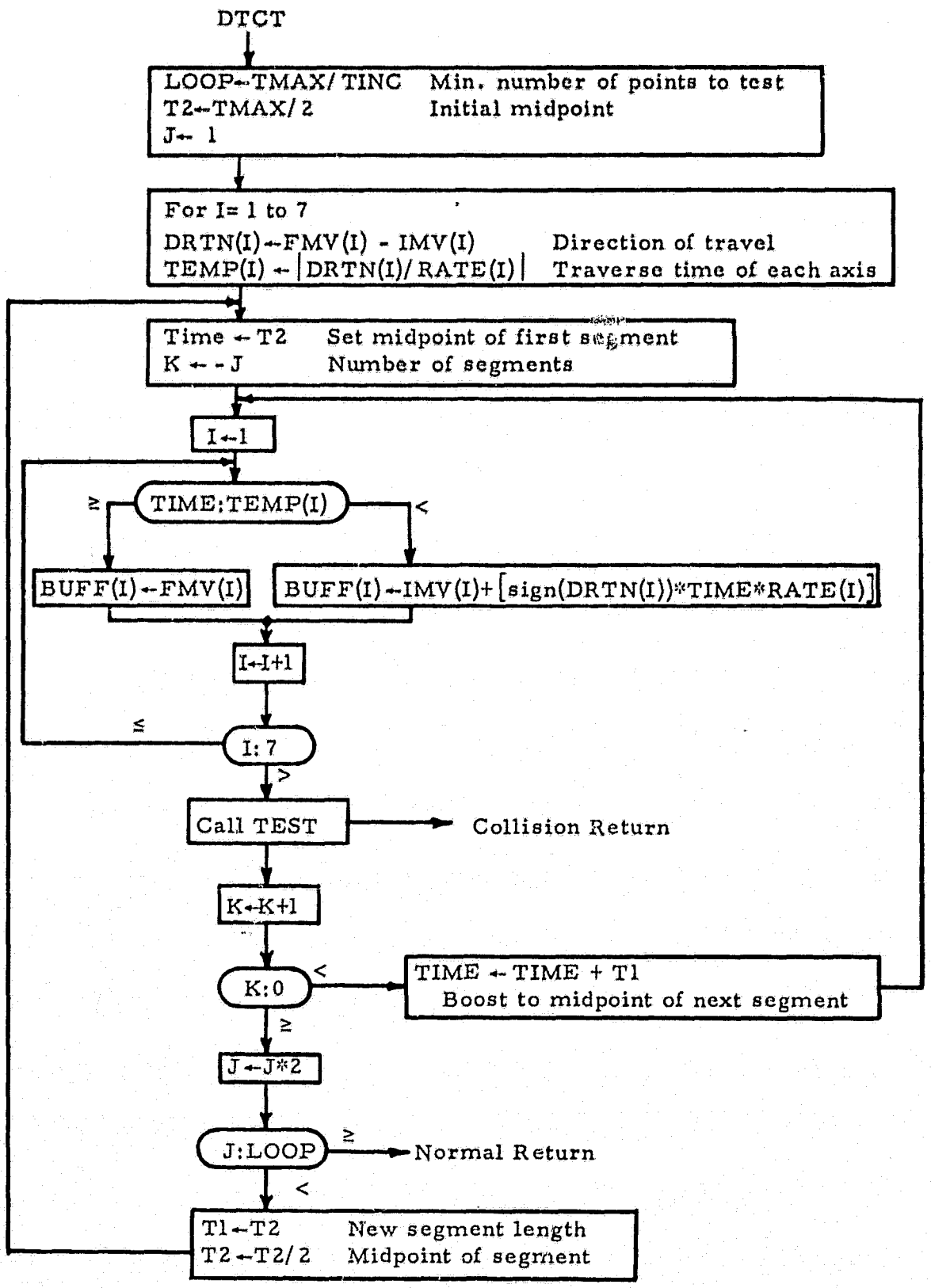


Figure 3-25. DTCT Flow Diagram

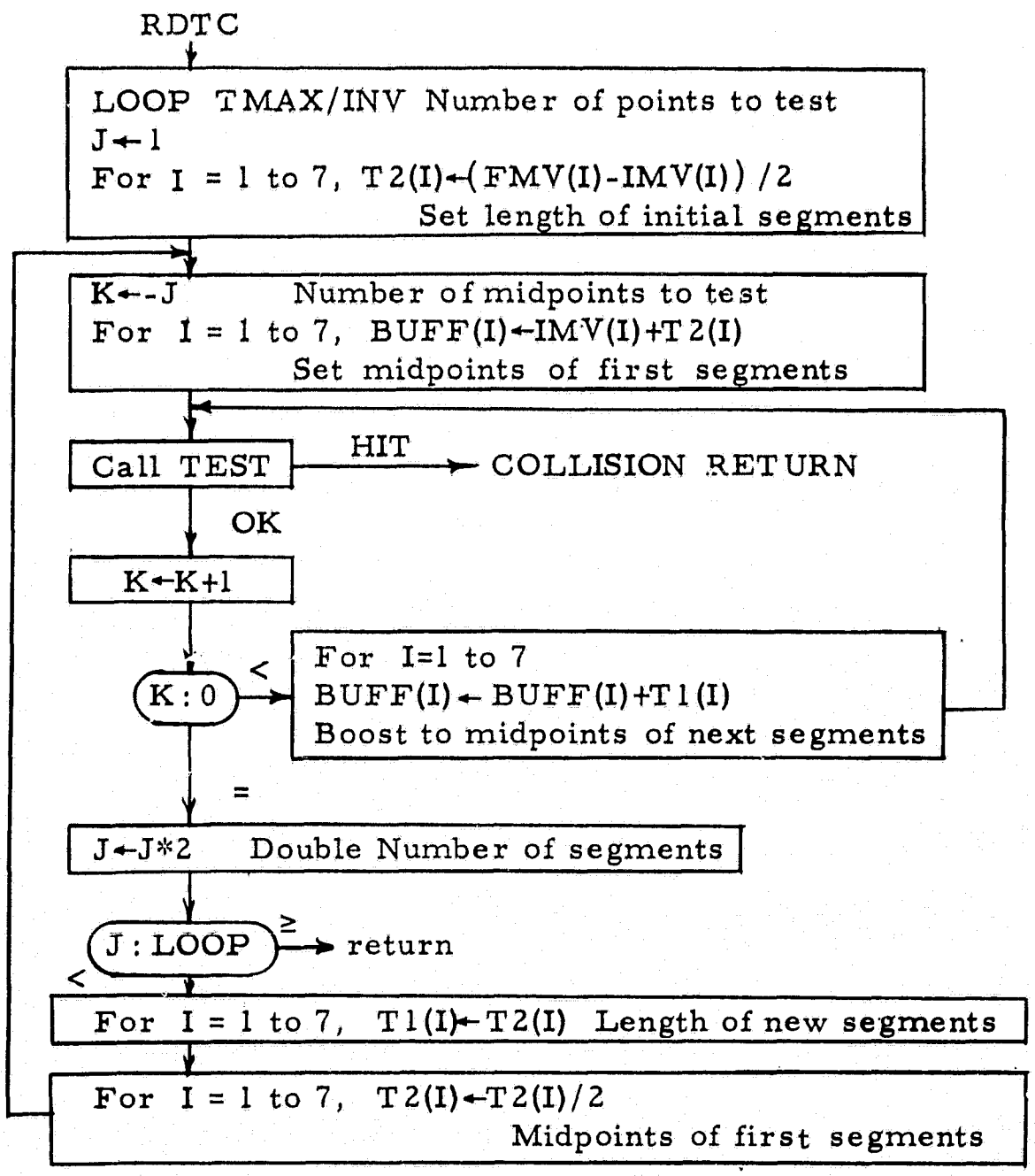


Figure 3-26. RDCT Flow Diagram

If the straight line connecting the initial and terminal hand positions intersected with a set of obstacle bounds, the "SUBG" routine is called.

The SUBG routine can direct the manipulator around only one obstacle at the present time and requests assistance if more than one obstacle was intersected. Further extensions of the routines were deferred until a sensory feedback loop is added to the system and a more comprehensive obstacle evasion technique can be developed.

When only one obstacle is encountered, the SUBG routine determines a set of intermediate hand variables which direct the manipulator over or around the obstacle, computes the approximate transit time required by each path, and selects the most promising one to output to the manipulator. A sketch of the three possible paths is shown in Figure 3-27. The hand variables at each of the intermediate points are selected as follows:

1. Over the obstacle
   a. First intermediate goal
   HX,HY = X,Y values of first intersection point
   WP = IWP + (FWP-IWP)/3
   SR = ISR + (FSR-ISR)/3
   b. Second intermediate goal
   HX,HY = X, Y values of second intersection point
   HZ = top of obstacle plus one inch
   WP = IWP + 2 (FWP - IWP)/3
   SR = ISR + 2 (FSR - ISR)/3

2. Around the obstacle
   Paths are tested in both a clockwise and a counterclockwise direction around the obstacle. The manipulator is directed to each corner in turn until the path to the desired hand position is clear. If N intermediate goals must be specified, the values of WP and Z at the M^{th} point will be

   $$Z = IHZ + M\,(FHZ - IHZ)\,/N$$
   $$WP = IWP + M\,(FWP - IWP)/N$$

   The values of HZ, HY and HZ are dependent upon the corner and are as shown in Figure 3-28.

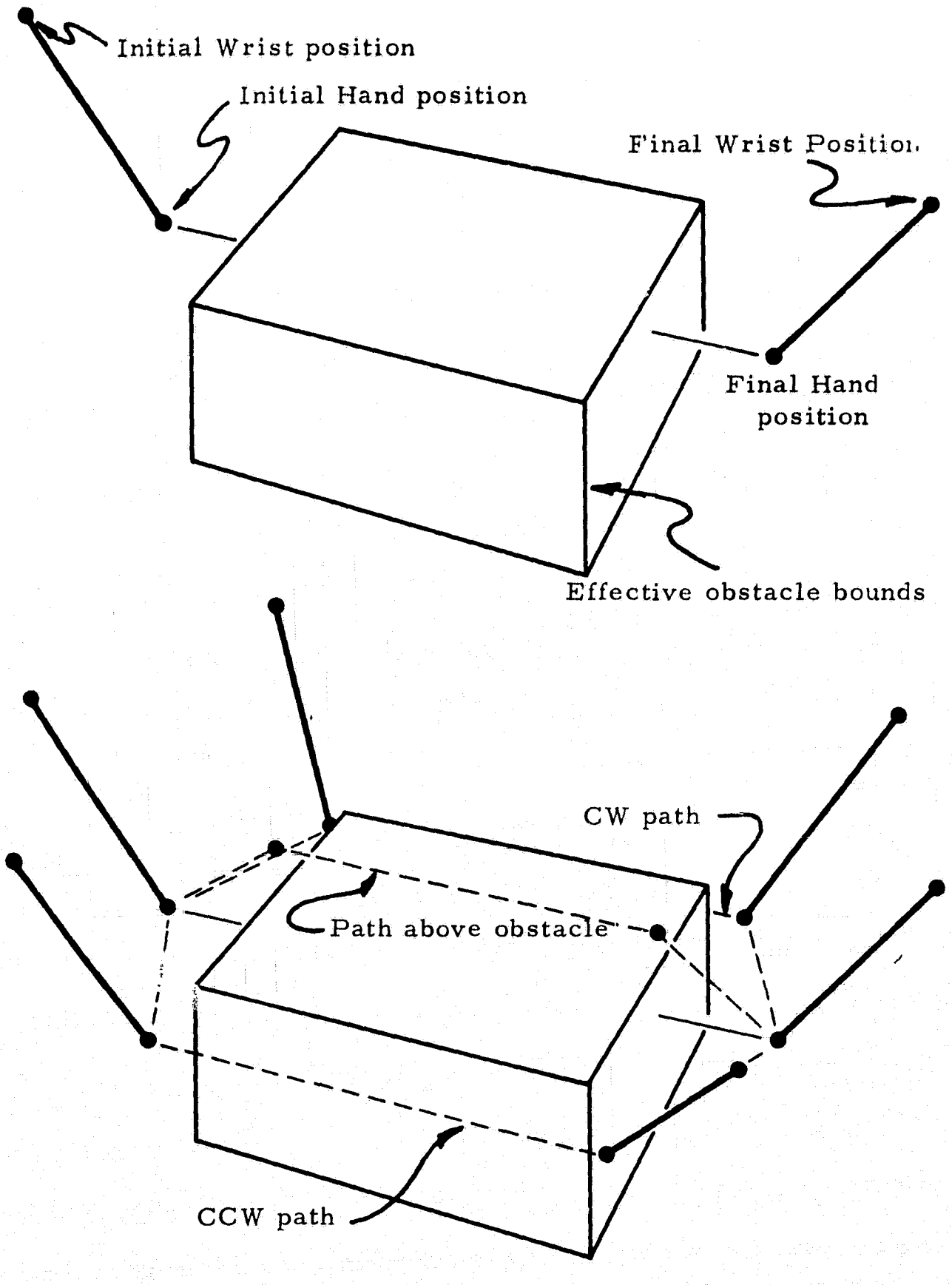


Figure 3-27. Obstacle evasion paths tested by SUBG

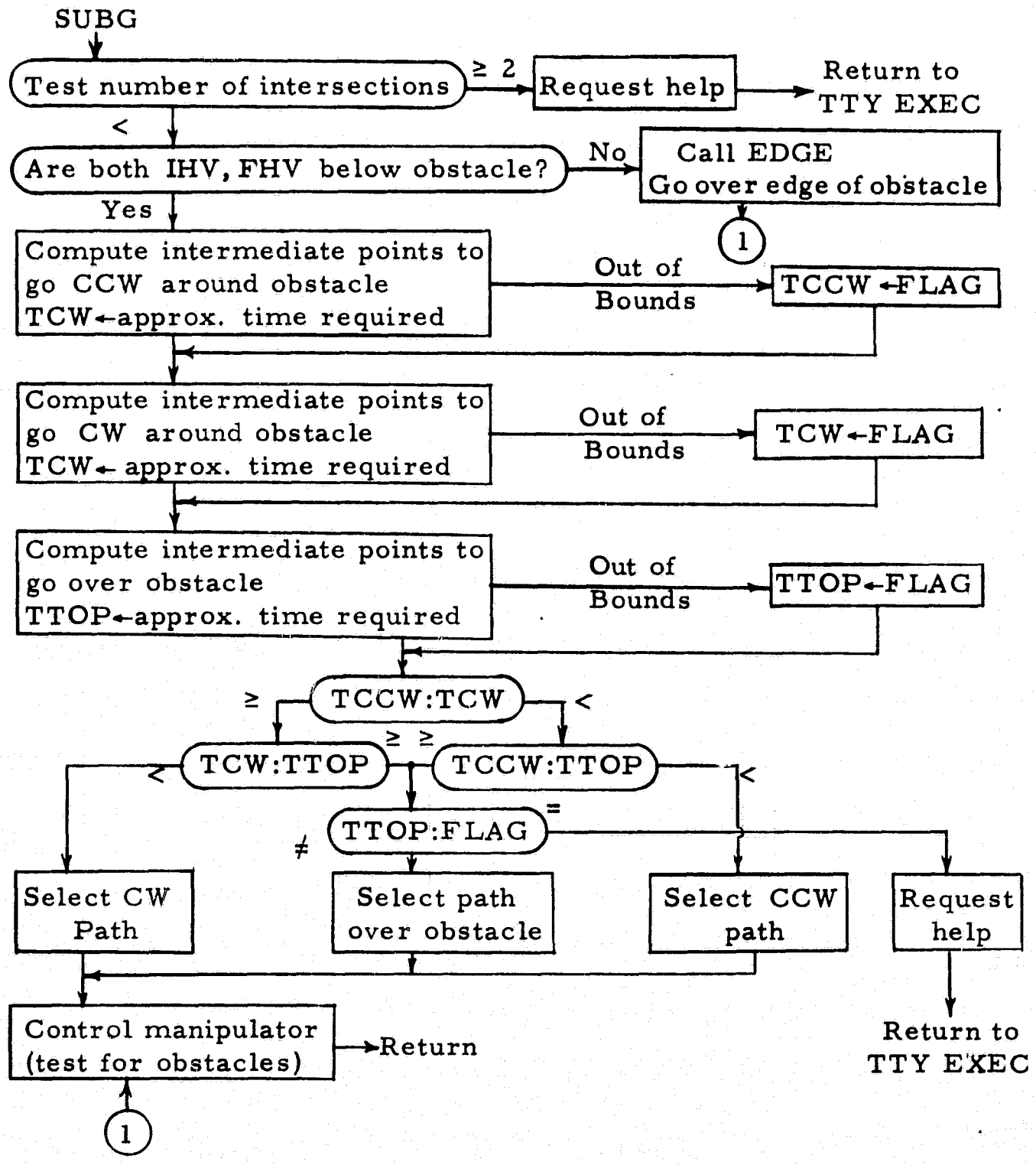
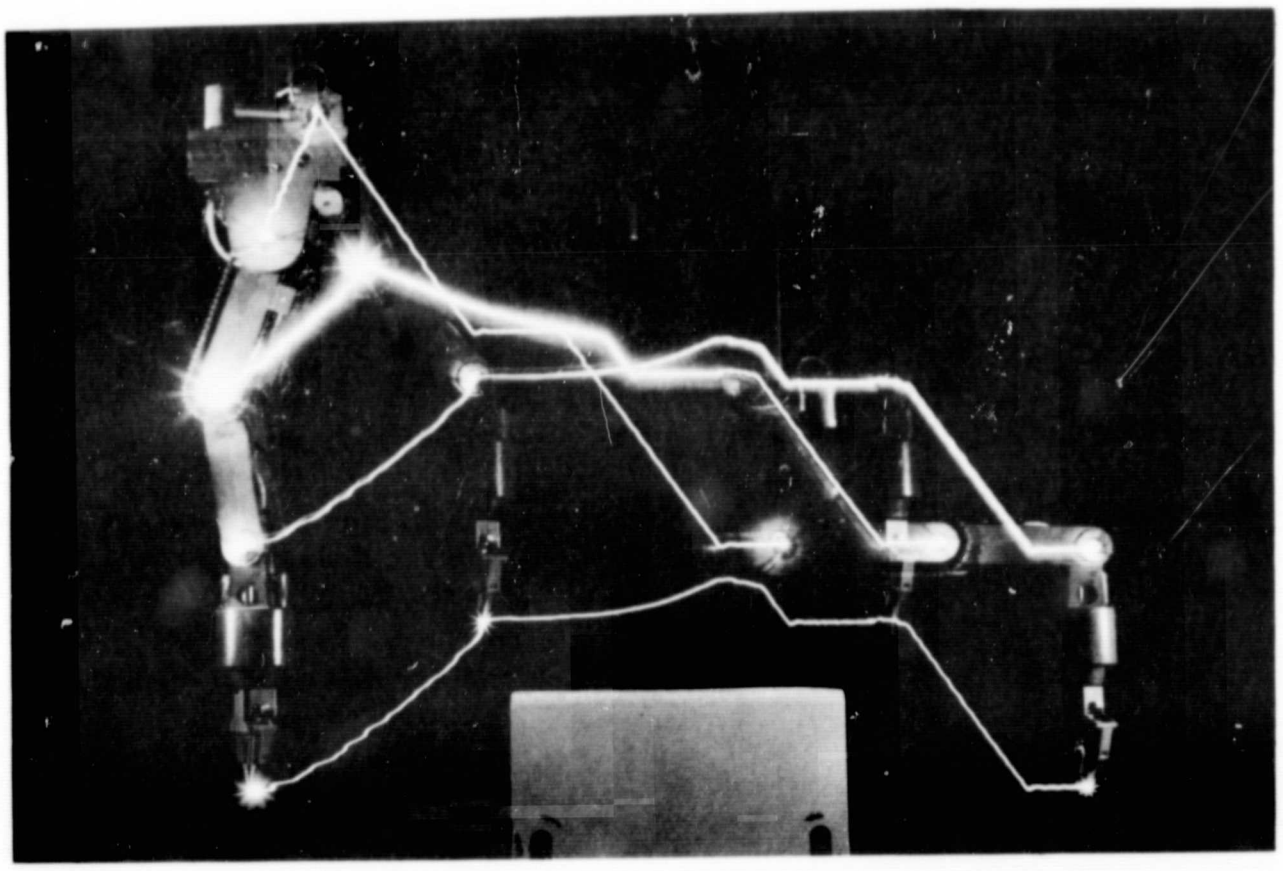
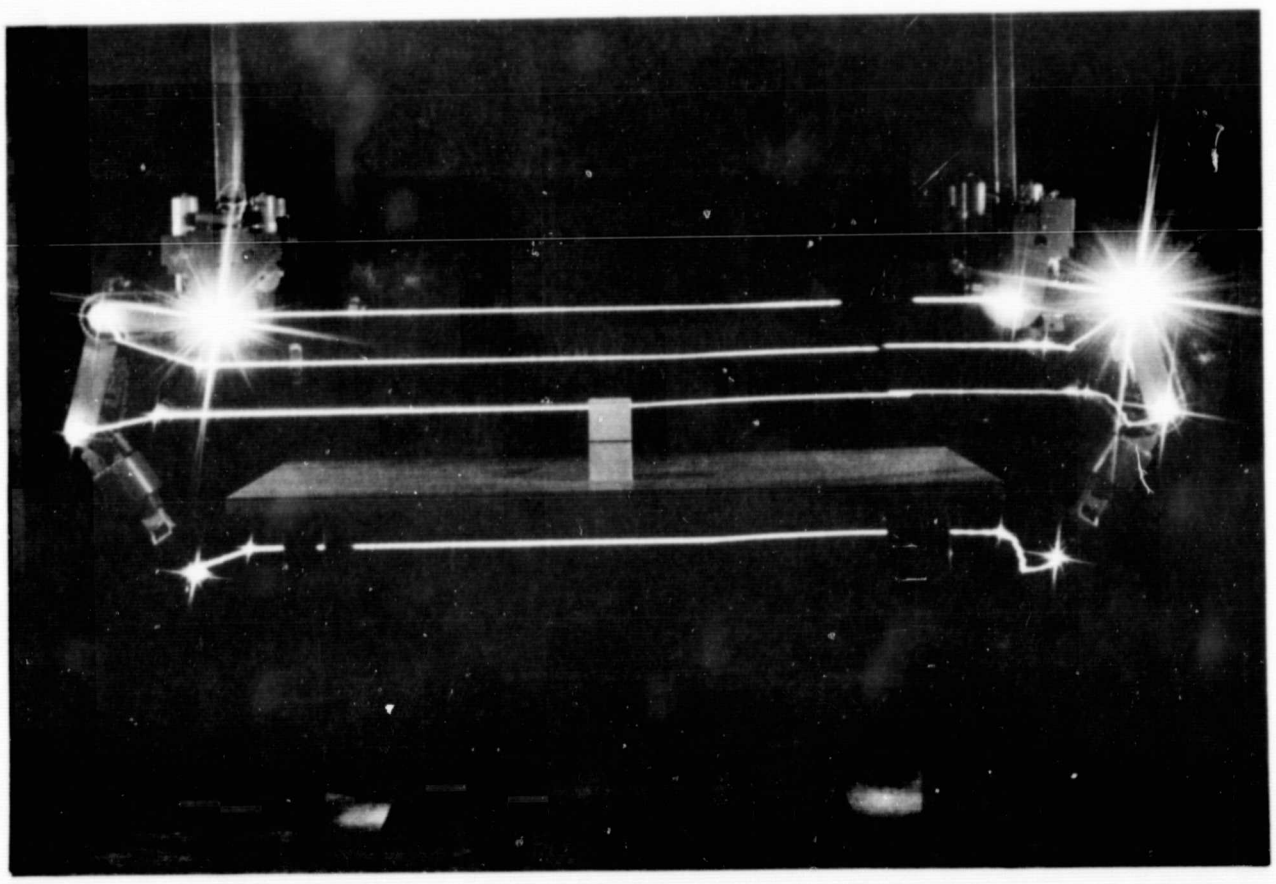


Figure 3-28. SUBG Flow Diagram

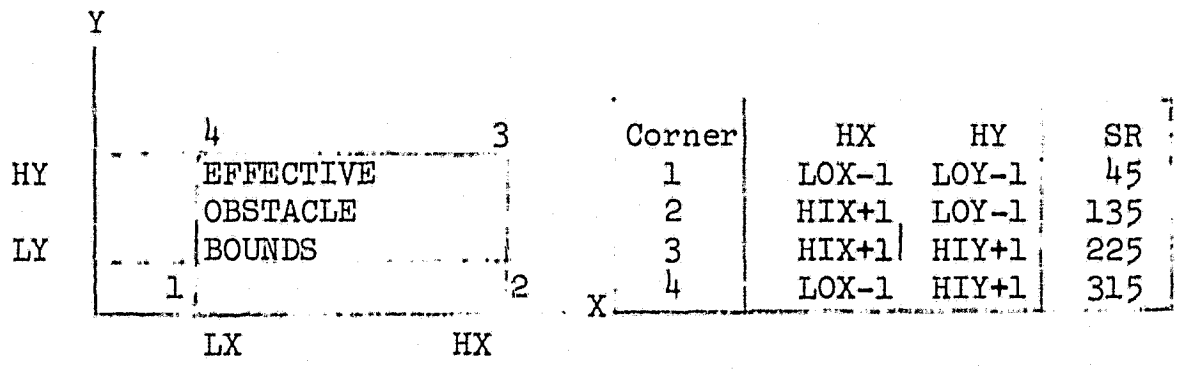


a. Path over obstacle



b. Path around obstacle

Figure 3-29. Photographs of manipulator paths generated to avoid obstacles



Corner	HX	HY	SR
1	LOX-1	LOY-1	45
2	HIX+1	LOY-1	135
3	HIX+1	HIY+1	225
4	LOX-1	HIY+1	315

Figure 3-30 Values of HX, HY, and HZ for Directing the Manipulator Over or Around an Obstacle

A flow diagram of the "SUBG" routine appears in Figure 3-28 and the photographs in Figure 3-29 demonstrate the path followed by the manipulator when directed over and around obstacles by the SUBG routine.[15]

3.3.5 Special Restrictions and Overrides

In order to provide for a greater degree of operator control over the operations performed by the manipulator, several special features have been included to permit the operator to modify the parameters used in determining the operations. First, the operator can selectively restrict the range of suitable arm configurations from the keyboard; and second, the operator can perturb or modify the moves made under computer control by using the manual controls.

3.3.5.1 Range Restriction

This mode of override is applicable only when the computer selected operations utilize the Hand mode of operation. Basically the operator may, as indicated in Figure 3-31, specify a group of regions in which the final arm configuration will be acceptable. This information is formatted and encoded so as to match the Wrist pivot Constraint Table and is applied by the routine "CONTAB." The parameter information is input through the routine "ICONTA." A flow

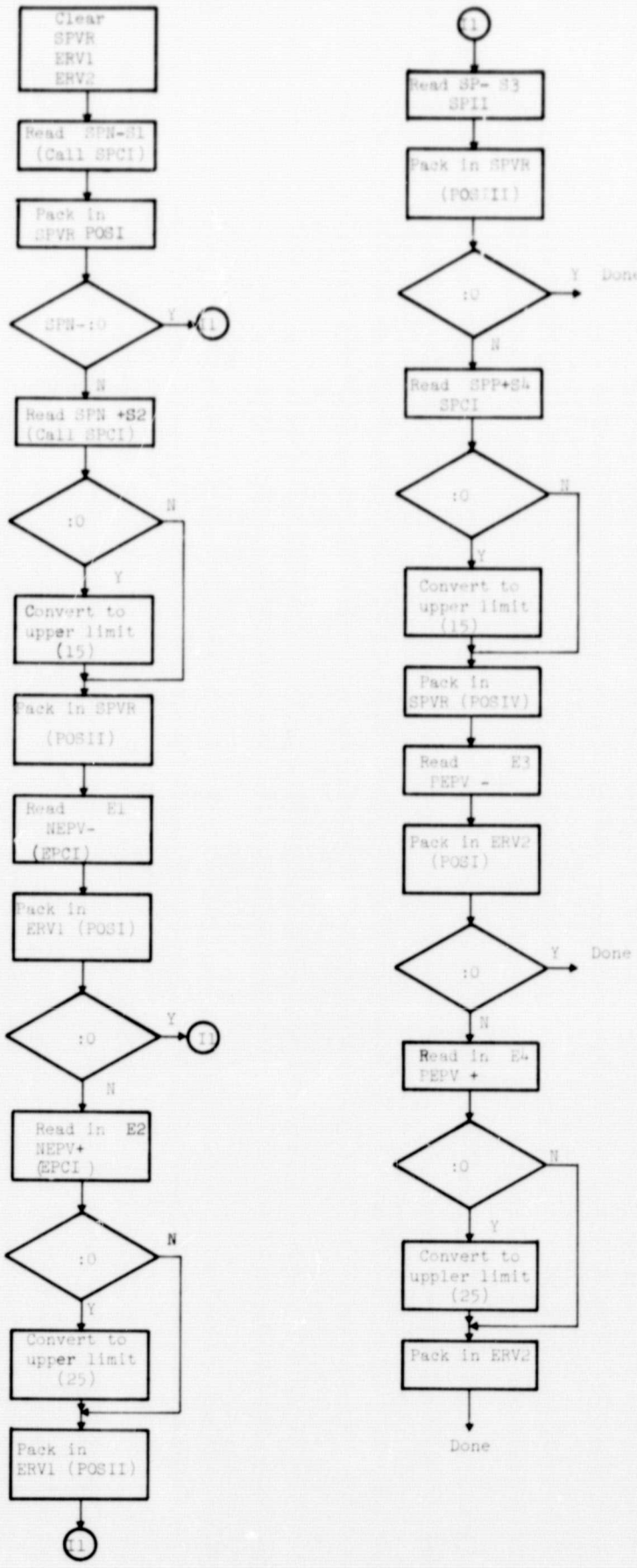


Figure 3-31 ICONTA Flow Diagram

diagram for CONTAB is given in Figure 3-32. An explanation of the input to this routine is given under the Z option for the TTY Exec an Appendix II. Basically these routines are used to set the sign bit on the table entries if the indicated table region is not allowed. The GRID routine examines this bit before allowing the configuration for this table entry to be used and then flags this region as unacceptable.

This pair of routines was added to permit the operator to restrict the motion of the manipulator in cases where the physical configuration of the manipulator had an effect on the feasibility of the task. Several examples of this are given in Appendix VI. In this case it was necessary to restrict the configuration of the limbs so as to permit the use of VECTOR moves. If the restrictions had not been applied, then it would have been necessary to generate Manipulator mode moves, and a relative operation could not have been recorded.

3.3.5.2 Overrides

As indicated in Section 2.8.2.1, it is possible for the operator to override the control commands given to individual manipulator axes by using the manual control on the operator's panel. If such an override is desired, it is possible to process this in several ways. The first technique would permit the operator to perturb the motion of the manipulator axes while not affecting the endpoint for the move. This requires that the computer leave the manipulator axis register alone and that the count control to this axis be inhibited. If this is

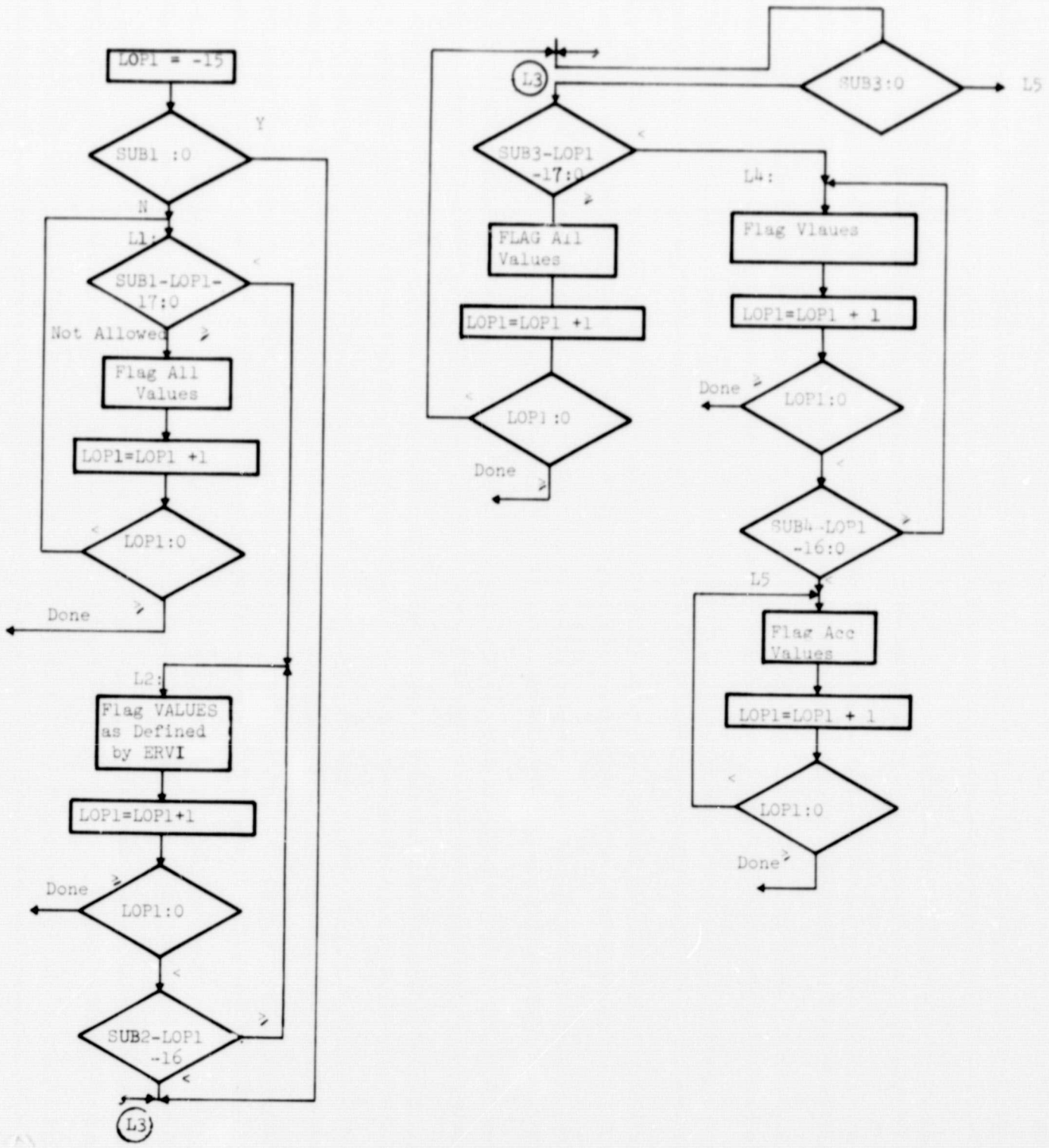


Figure 3-32 Contab Flow Diagram

done, when the operator releases the control, the drive will once again receive the original setpoint information and will drive the affected axis to its specified position. Another action would call for allowing the operator to control the actual position of that axis and not perturb the setpoints for the other axes. To allow this it is necessary that the new endpoint be determined. This may be done in either of two ways: 1. the counting inputs to the specified buffer register could be enabled; or 2. the computer could track the motion utilizing a Software Analog to Digital conversion technique. The first technique has the disadvantage of a long conversion time if the error between the actual endpoint and the setpoint is large. In addition since control reverts to the setpoint as indicated by the register, it would be necessary for the operator to maintain a slight error input until the register had converged. The software A/D technique has the advantage of a fixed conversion time. It has the disadvantage that if the manipulator is moving when the conversion is made the value obtained may be off by several quanta. This is not a severe problem since the operator does not have a good feel for the actual position but only relative position and because this type of error is reduced when the motion is made at a slow rate. This would be the case when the operator is applying only a slight error as in fine position moves. This latter technique has been selected and is implemented in the experimental Teleoperator System. The Software A/D technique used is explained in Section 3.5.

In order to utilize this tracking technique it was necessary

to determine when the operator took control of any of the manipulator axes. This is done by allowing the detectors on the manual controls to generate an interrupt when this type of manual override is allowed. Due to conversion time restrictions, this tracking mode is only possible when the routine ETST is being used to wait for the final endpoint. Due to this restriction, the tracking mode is available only in Hand and Manipulator moves that do not utilize the Rate Simulation mode of operation.

The routine ETST allows interrupts from the external interrupt of the ICGPI, and from the Real Time Clock. The ICGPI is used to indicate when the manual control is first actuated. When the interrupt is processed, a switch on the operator's panel is sampled to determine whether the tracking mode is desired. If so, a bit is set in a conversion flag word. After checking each of the axes, those axes that are perturbed are masked off, and control is returned to the interrupted program. The RTC is used to increment a pseudo clock in core which is used to time the motion, and is used to check the status of the manual inputs. If any register is not interrupting, it is masked on again and the routine returns to the interrupted routine.

The ETST routine basically waits for the manipulator to complete its motion. It first checks to see if the time limit has been exceeded. Then it checks for zero error. Finally it checks to see if any of the axes have been perturbed and are to be tracked by checking the flag word and a status word maintained internally. If no axes have been perturbed ETST goes back to check the time limit. If registers are to be tracked, ETST calls the routine FSAD with a word

indicating which axes are to be tracked and updates the position buffers associated with these axes. When the move is completed, either due to reaching the endpoint or exceeding the time limit, the routine returns control to the calling routine.

3.4 Automatic Mode

By means of the Automatic mode it is possible for the operator to record or preprogram a specific task. These tasks can be recorded in either a relative or in an absolute format. They permit the operator or someone associated with the device to prepare operations for execution at a later time or to prepare operations that must be performed many times so as to reduce the fatigue and actual operator input required.

As shown in Figure 3-2, the AUTO Exec is permitted to call the basic manipulator control algorithms after transferring required information to the specific buffers. The WAIT command requires the AUTO Exec to wait for a period of time before preceding the next operation. This can become important when an object is carried in the Hook and must be given time to settle out before the next operation is performed.

The constraint table change permits the programmer to restrict the configurations used for a given sequence of operations. The tool changing permits the programmer to specify a tool change when necessary. Actual tool changing was not implemented on this device because the mechanical system did not have the necessary stiffness and resolution to permit this task to be performed with any degree of reliability. The TTY Return command was added to permit

the programmer to return control to the operator for various reasons. The object reference points are used for relative operation and permit the generation of a relative program for use at a later time anywhere in the manipulator working area. Grip control was provided to permit the preprogramming of some of the tasks that must be performed by the Hand. Relative control of Hand moves was included due to restrictions which were applied to limit the processing of relative Hand positioning.

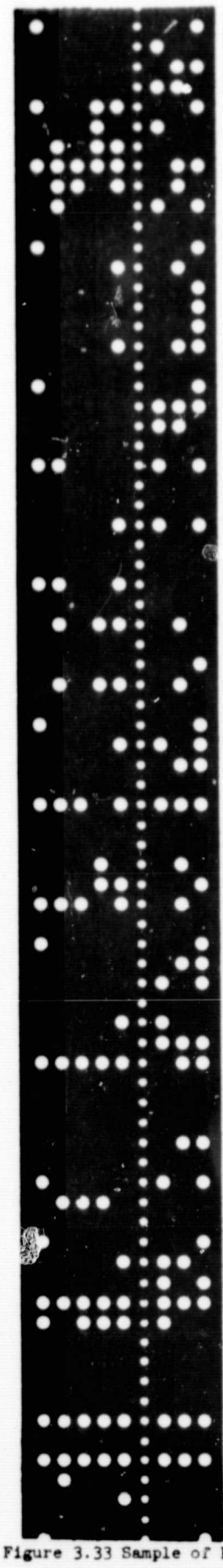
Appendix VI gives an example of a prerecorded program which can be used with the Case Teleoperator System for the partial dissassembly of a Reactor Model.

3.4.1 Record Data -- Punch Paper Tape

The "PUNCH" routine permits the operator to program a series of operations for use by the AUTO Exec routine. In general, positional information may either be specified by the operator or may be determined by the actual position of the manipulator. Figure 3-33 gives an example of some of the blocks of data that are used by the Punch routine. Figure 3-34 gives a flow diagram for the PUNCH routine. In general, each punch command includes:

1. a flag to the AUTO Exec to indicate which operation is to be performed; (followed by)
2. a parameter indicating the number of words of data; (followed by)
3. the data; (and)
4. a checksum for parity checks.

Following the checksum, a blank frame is recorded by the next punch



Start Code
Block Type 4 (Origin)
3 words of data
X

Y

Z

Parity
Blank (OK)
Start Code
Block Type 10 (Request Tool)
1 word of data
New Tool (Hook)
Current tool
Parity
Blank (OK)
Start Code
Block Type 7 (Grip Hand Control)
6 words of data
CMND
(E) Absolute hand and grip position
Time (not a timed move)

Turn (13 turns/inch)

Gsiz
.200 inches
Wang
90 degrees
Wtns
1 turn
Parity
Blank (OK)
Start Code
Block Type 13 (Constraint table)
3 words of data
SPVR

EPV1

EPV2

Parity
Blank (OK)
Start Code
Block Type 3 (Relative Hand Variables)
5 words of data
Elevation

Azimuth

X

Y

Z

Parity
Blank (OK)
Start Code
Block Type 14 (Relative Hand Hand Move)
5 words of data
ΔElevation

ΔAzimuth

ΔX

ΔY

ΔZ

Parity
Blank (OK)

Figure 3.33 Sample of Paper Tape Block Structure [16]

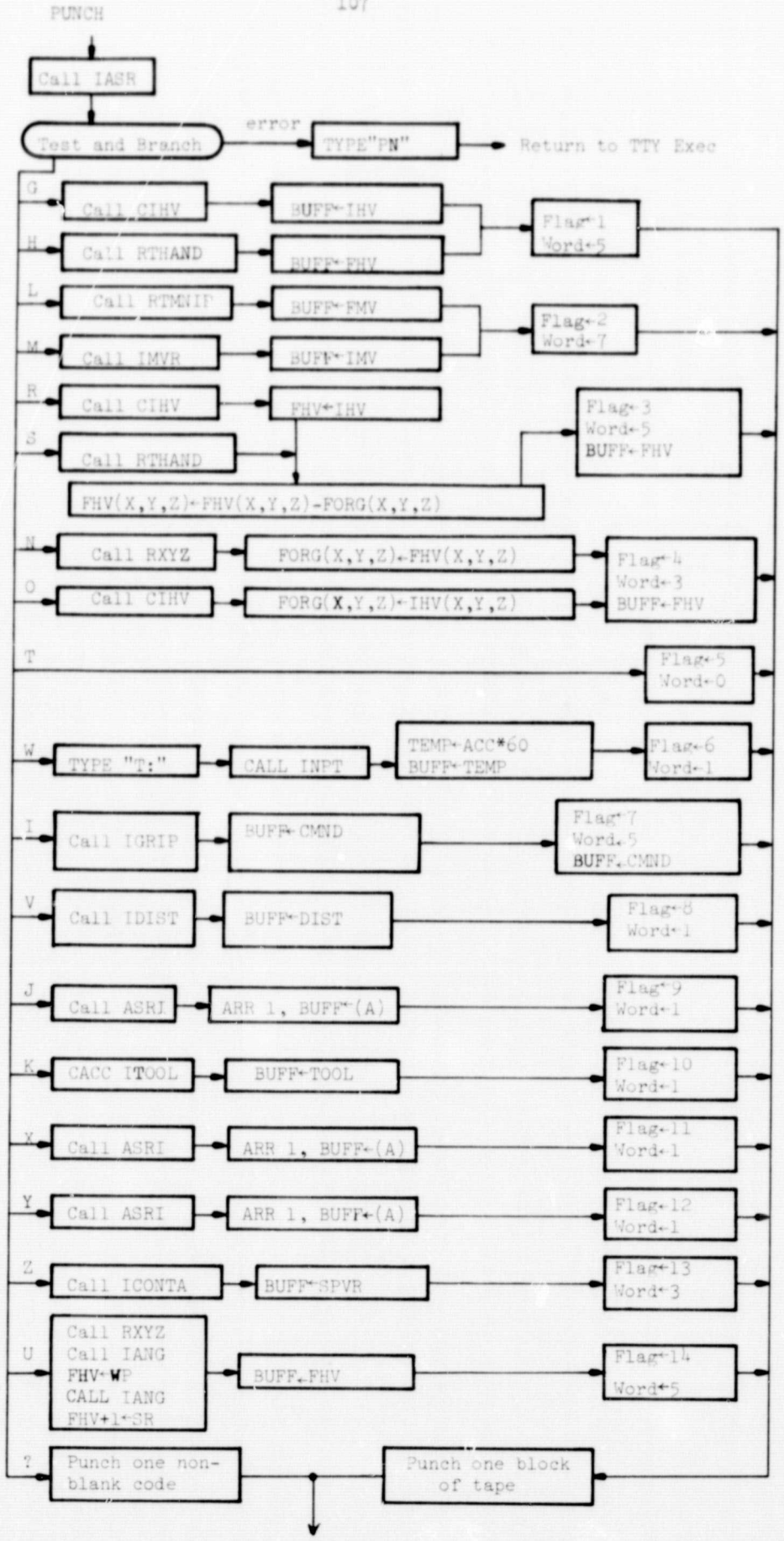


Figure 3-34 PUNCH Flow Diagram

operation, unless the operation is ? which indicates that the previous block is in error.

3.4.2 AUTO Exec -- Paper Tape Read Routine

Once control has been transferred to the AUTO Exec, a block of paper tape is read from the high-speed paper tape reader. After checking to determine if this is a proper block, control is transferred to the routine requested. By typing a "Q" the operator may return control to the AUTO Exec without reading a new block of data. This has been provided in case the routine is interrupted by accident or the operator desires to reexecute a command. Control may be transferred to the TTY Exec routine by one of the six conditions listed in Section 3.2.2.2. A block diagram of the AUTO Exec is given in Figure 3-35.

3.4.3 Coordinate Transformation

The utility of the Automatic mode is extended considerably by the use of coordinate transformation routines. By using these routines it is possible to specify hand variables relative to a specific object rather than in terms of the manipulator coordinates. It is this feature that has permitted the generation of a tape for the automatic removal of the endbell and core on the reactor model. By using this feature it is possible to perform these operations in any area of the manipulator space that is not required for the final depositing of the reactor parts. This operation is detailed in Appendix VI and is a mixture of several types of relative operations and several absolute moves. In general, the operations that are concerned with the reactor assembly are recorded in a relative format; and the operations that are

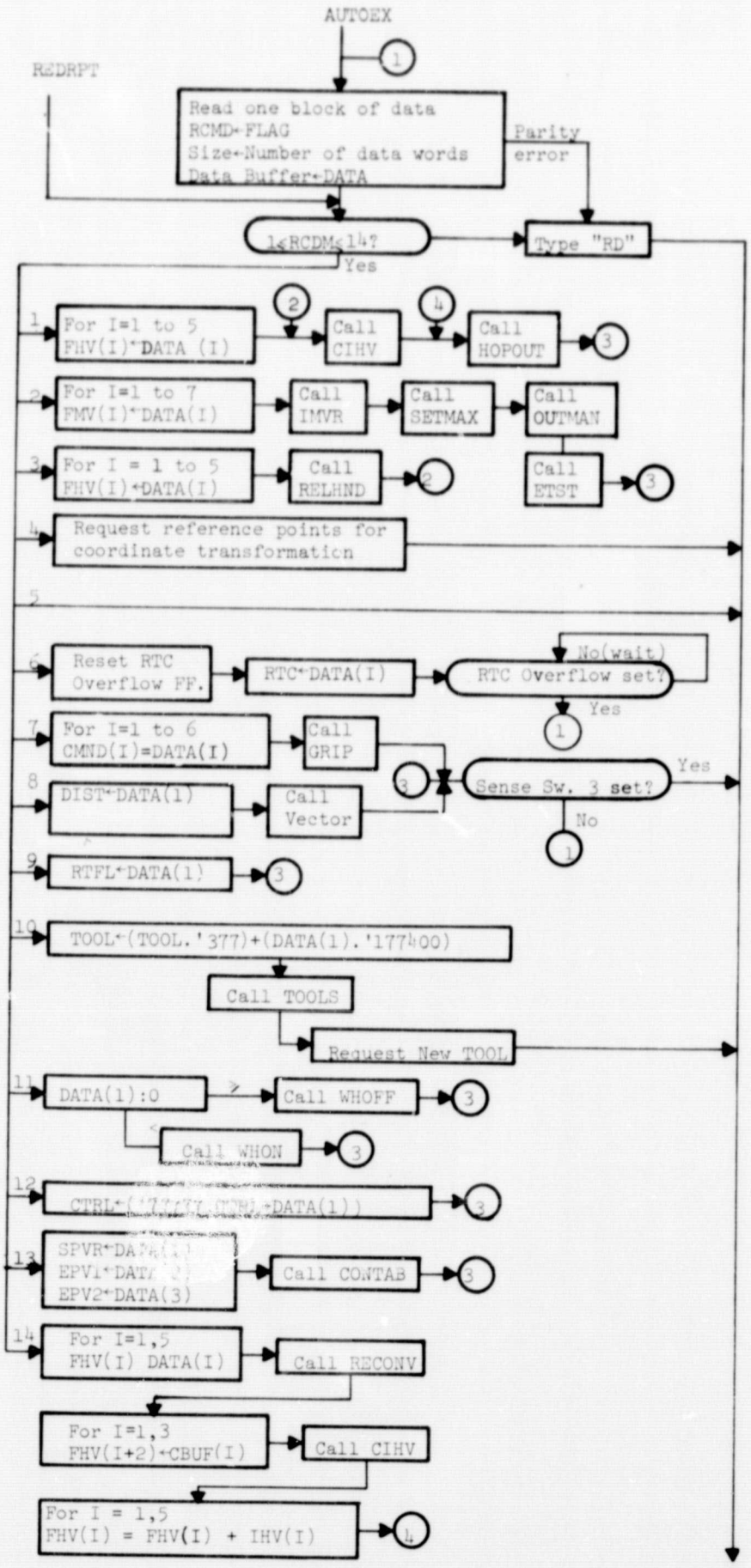


Figure 3-35 Flow Diagram For AUTOEX[18]

concerned with the disposition of removed elements are recorded in an absolute format. This is possible since only the operations recorded in two special types of relative hand operations are transformed. The first type is a relative hand variable. This position is relative to the origin of the relocatable object space which permits the generation of a move to a specific place relative to the subject object. The second type of move is a relative hand move which is similar to the incremental hand move allowed by the TTY Exec. This permits the recording of an incremental hand move in X, Y, Z, SR, and WP relative to the current position with the X, Y, and Z increments transformed.

3.4.3.1 Recording Relative Hand Variables

The relative hand variables may be recorded by going through the operation with the manipulator and punching a relative hand variable block at each desired endpoint. They may also be programmed directly from a set of drawings. In either case, an origin must be assigned to permit the allignment of the operating relative coordinate system with the actual object. Currently the three reference points are assumed to be eight inches from the origin and located at (8,0,0), (0,8,0), and (0,0,8) for points #1, #2, and #3 where the positions are (X,Y,Z). This type of origin and orientation information simplifies the operations which must be performed in order to calculate the transformation matrices.

If the moves are recorded from actual manipulator moves, it is necessary to align the subject with the manipulator coordinate system

so that the object X, Y, and Z axes are parallel with the manipulator axes. In addition, it is necessary to punch an origin block so that origin information will be requested. If the origin is specified form the console, P_o=(0,0,0) should be specified.

3.4.3.2 Recording Relative Hand Moves

Relative hand moves can only be recorded in one way. This requires that the operator determine the increments desired for the five hand variables. These increments can be determined from calculations based on drawings or by performing the operation using the HI option of the TTY Exec and utilizing the inputs to this mode. By typing a PU the operator will enter the relative hand move punch routine and the five hand variable increments will be requested. Once the increments have been supplied they will be encoded and punched in a special block on the paper tape.

3.4.3.3 Executing "Relative Hand Data"

Before relative hand data can be executed, it is necessary for the operator to specify the origin and alignment information for the object. Once this information has been supplied, the operator is required to request the computation of the transformation matrices. The preparation of these matrices is described in Section 3.4.3.4.

Whenever relative hand data is encountered, the X,Y, and Z information is transformed from the object coordinate system to the manipulator coordinate system. For the relative hand variables it is also necessary to transform the Azimuth and Elevation values. The recorded angular values are used to compute the wrist pivot position relative to the object coordinates. This position is transformed and used to

define WP and SR values in terms of the manipulator coordinates. Figure 3-36 indicates that there are two possible combinations of angles. In this system the value of WP is restricted to

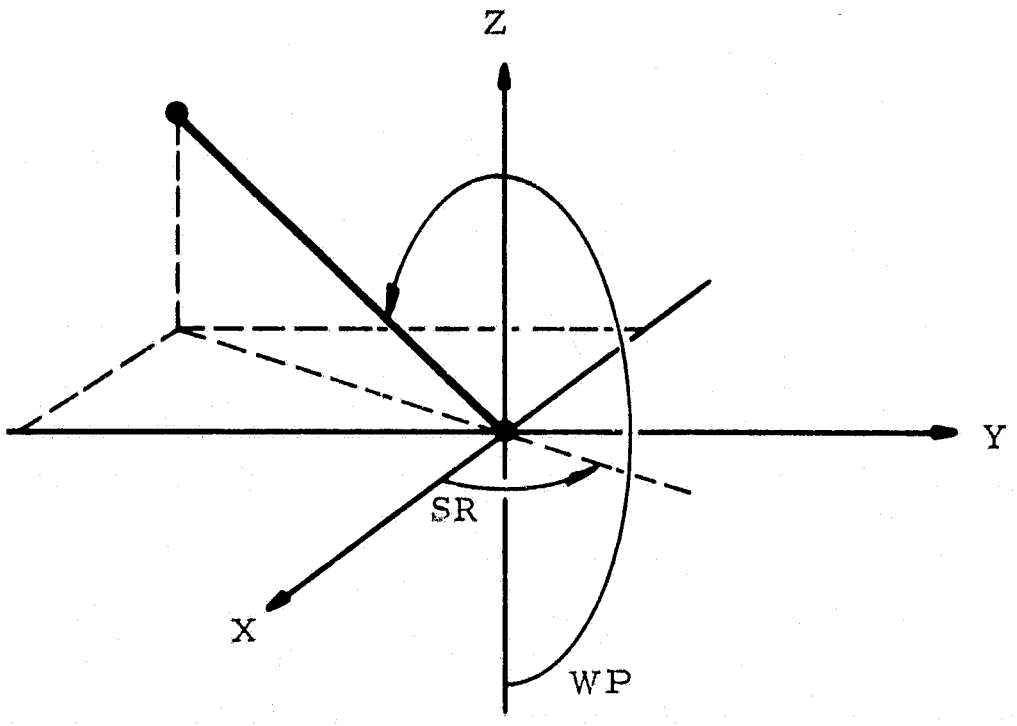
$$0 \leqslant WP < 180^\circ \qquad (9)$$

in order to form a unique solution. This restriction is the main reason for the second form of relative hand data. It was made necessary since several of the operations required for the removal of the endbell and the core require that the WP angle be taken from -3° to +1°. If these operations were performed using the hand variable routine, the manipulator would have to go through a SR move to 180°. This was not possible since the structure could not revolve about a point and it would have greatly increased the time required if it had been possible. In addition, to avoid this problem, several sequences of instructions were recorded with the WP angle held slightly positive so as to force the direction of approach to the Model. This was particularly important when the hook was in use since it must approach the hookeye from the proper side in order to engage the eye for lifting operations.[19]

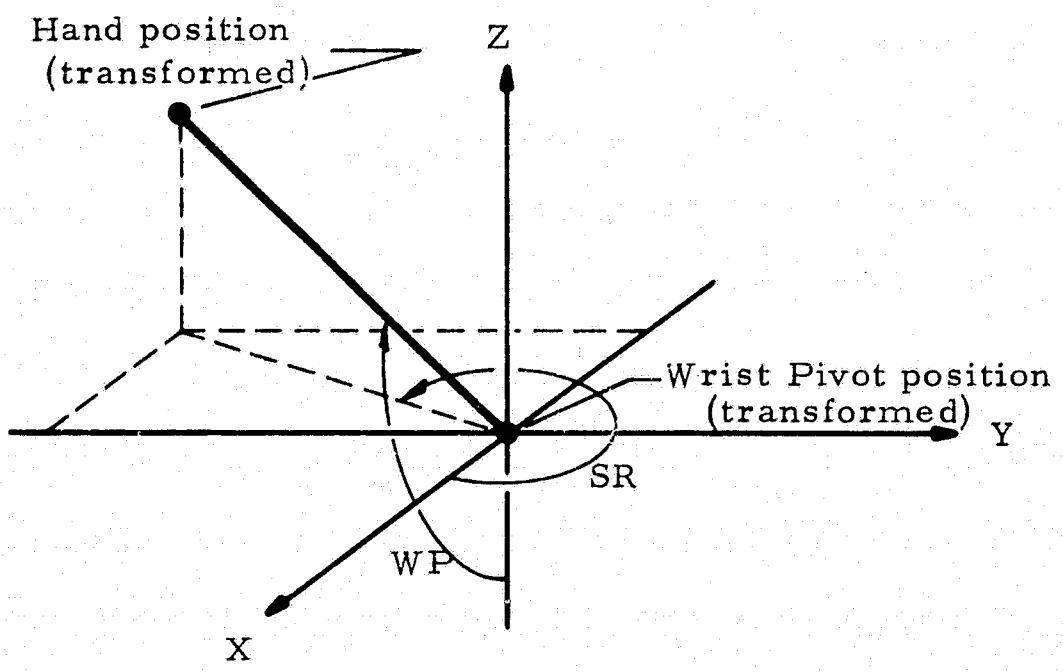
Flow diagrams of the routines used for transformations are shown in Figures 3-37, 3-38, and 3-39.

3.4.3.4 Transformation Equations

An "object" cartesian coordinate system is defined and all "relative hand positions" are specified in terms of this coordinate system during the recording process. After the object is placed in the task area, this coordinate system is rotated and translated to coincide with the manipulator coordinate system. A point P(X,Y,Z) measured in terms of the object coordinates is then transformed to the



a. WP > 180°



b. $0 \leq WP \leq 180^{\circ}$

Figure 3-36. Transformation of WP and SR [20]

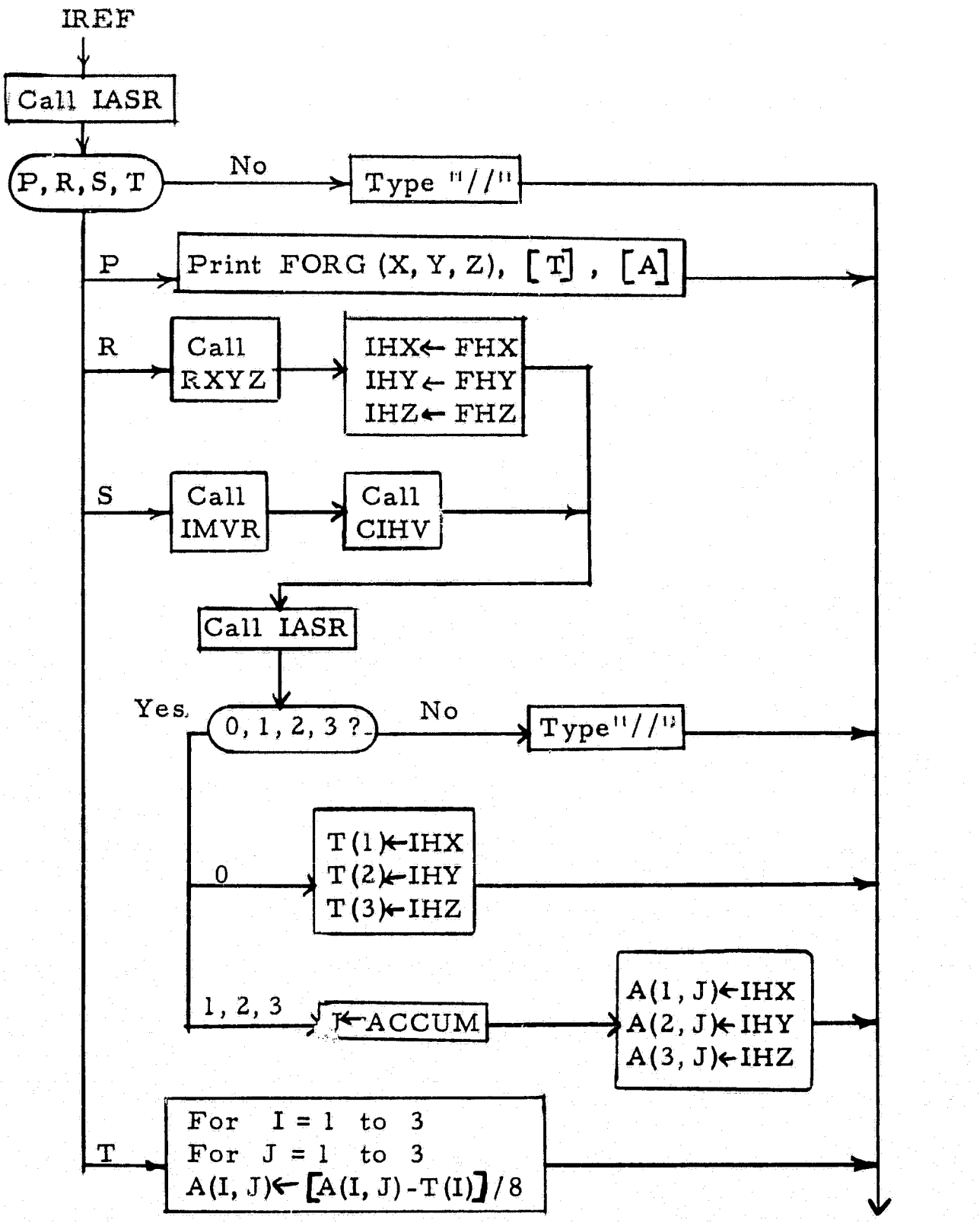


Figure 3-37. IREF Flow Diagram [21]

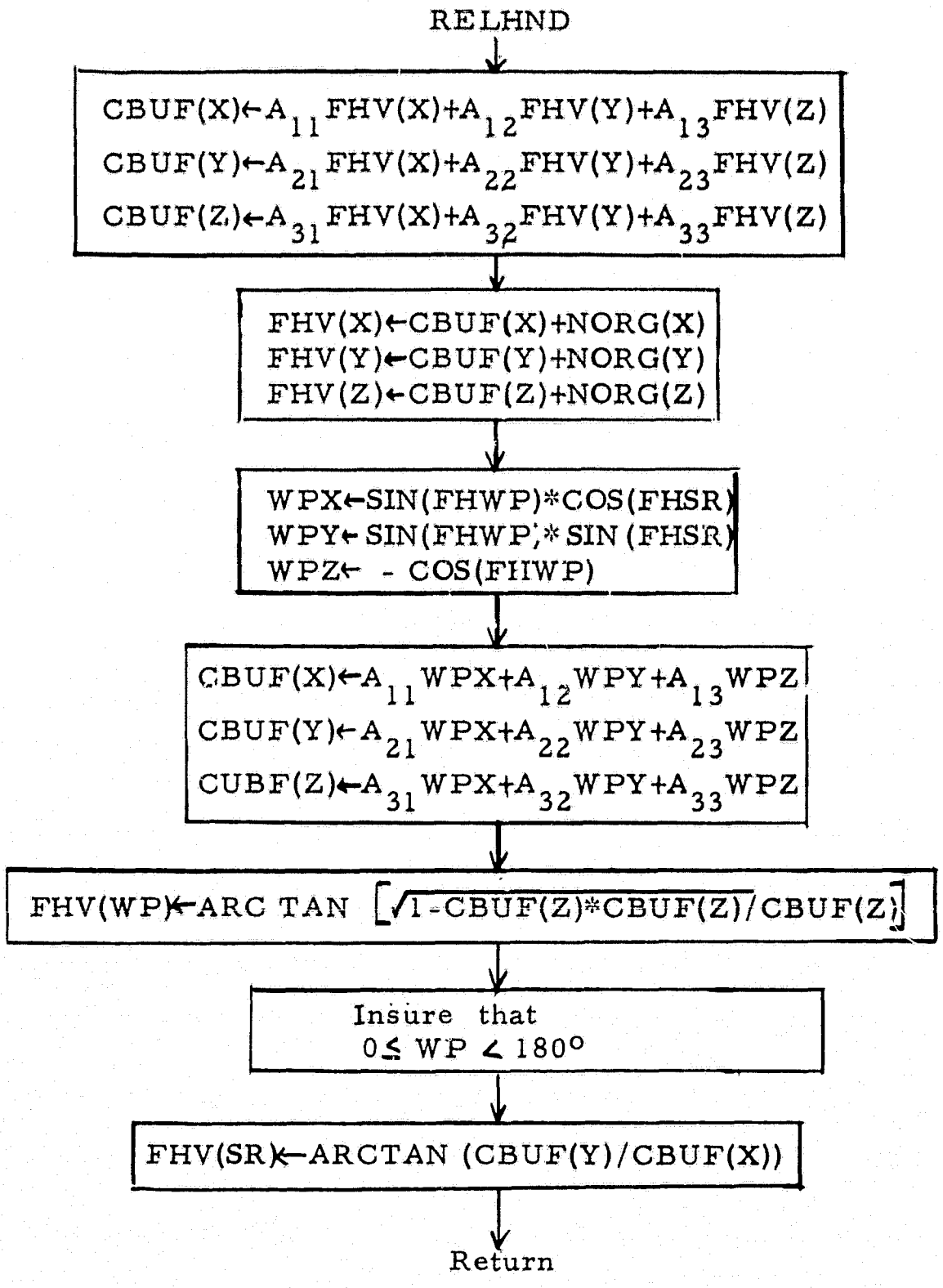


Figure 3-38. RELHAND Flow Diagram[22]

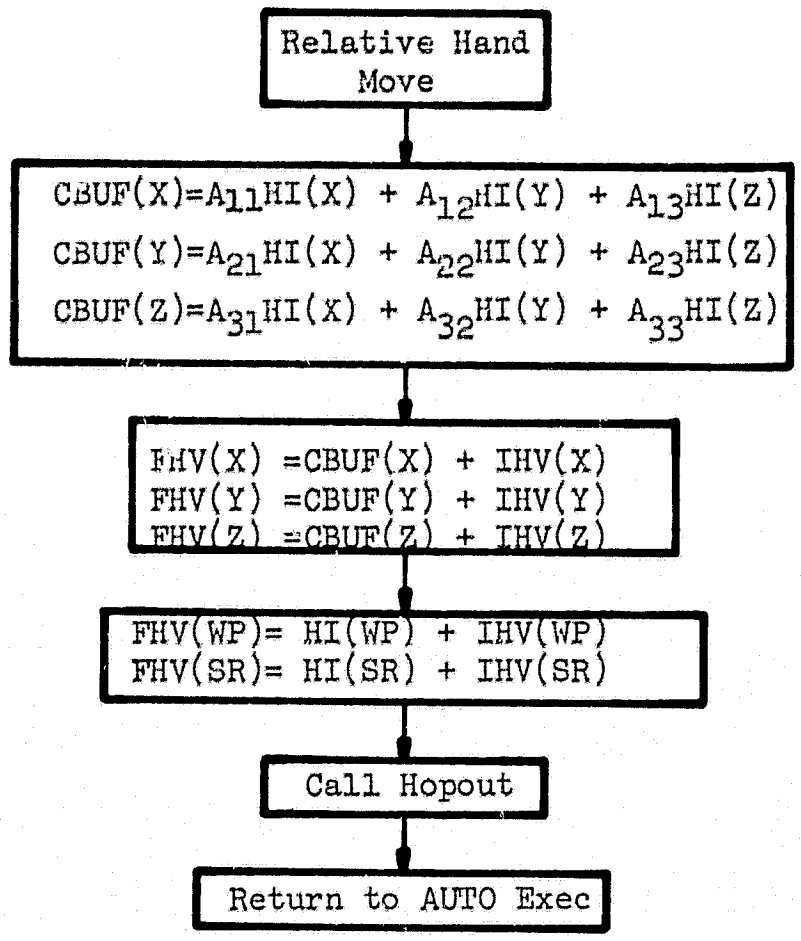


Figure 3-39 Flow Diagram For Relative Hand Move

position P'(X', Y', Z') measured in terms of the transformed (manipulator) coordinates. The standard transformation equation is

$$[P'] = [A]\,[P] + [T] \qquad (10)$$

where the matrices [A] and [T] represent the rotation and translation of the coordinate system respectively.

In order to determine the nine coefficients of [A] and three coefficients of [T], the origin, P_0, and three reference points are marked on the object. In the routines implemented these reference points in terms of the object coordinates are located at

$$P_0(X_0,Y_0,Z_0) = (0,0,0)$$

$$P_1(X_1,Y_1,Z_1) = (8,0,0)$$

$$P_2(X_2,Y_2,Z_2) = (0,8,0)$$

$$P_3(X_3,Y_3,Z_3) = (0,0,8) \qquad (11)$$

After the object is placed in the manipulator area, these four points are located and their positions are specified by the operator in terms of the manipulator coordinates. Since the values of P and P' are thus known for these four reference points, the coefficients can then be determined. From Equations (10),

$$[T] = [P'_0] + [A]\,[P_0]$$

$$[A]\,[P_1] = [P'_1] - [T]$$

$$[A]\,[P_2] = [P'_2] - [T]$$

$$[A]\,[P_3] = [P'_3] - [T] \qquad (12)$$

Substituting equations (11) into (12),

$$[T] = \begin{bmatrix} X'_0 \\ Y'_0 \\ Z'_0 \end{bmatrix} \qquad (13)$$

and

$$[A] = 1/8 \begin{bmatrix} X'_1 - X'_0 & X'_2 - X'_0 & X'_3 - X'_0 \\ Y'_1 - Y'_0 & Y'_2 - Y'_0 & Y'_3 - Y'_0 \\ Z'_1 - Z'_0 & Z'_2 - Z'_0 & Z'_3 - Z'_0 \end{bmatrix} \qquad (14)$$

Note that it would be possible to select any set of reference points which would produce a non-singular matrix; however, unless they are located along the axes, any other

choice would require the solution of three sets of simultaneous linear equations of three variables.[23]

3.4.3.5 Future Extensions

The relative operations permit a great deal of flexibility with regard to the actual location of the reactor model. However, there are still several restrictions which are applied, such as the limit on the final angle for relative hand position operations. If the rotation of the hand were to be taken into account, it would remove the possibility of two moves for a given configuration and would thus permit the removal of the restriction on final WP angle.

The current vector technique of utilizing only the X, Y, and Z axes for motion could be expanded to utilize all of the manipulator axes if the incremented X, Y, and Z positions were processed by the hand algorithms. This would reduce the number of operations which had to be subdivided due to the insufficient range on the Z axis.

When using relative hand operations, several extensions would be helpful:

1. the addition of a set of programs to solve simultaneous equations to permit the use of arbitrary reference points.

2. the ability to generate a transformation matrix for an object so that operations could be recorded for an object that was not aligned with the manipulator coordinate system.

3. the addition of a set of reference points so that moves could be recorded relative to several different points in one sequence.

3.5 Software A/D Conversion

The "SAD" and "FSAD" subroutines, illustrated in Figures 3-40

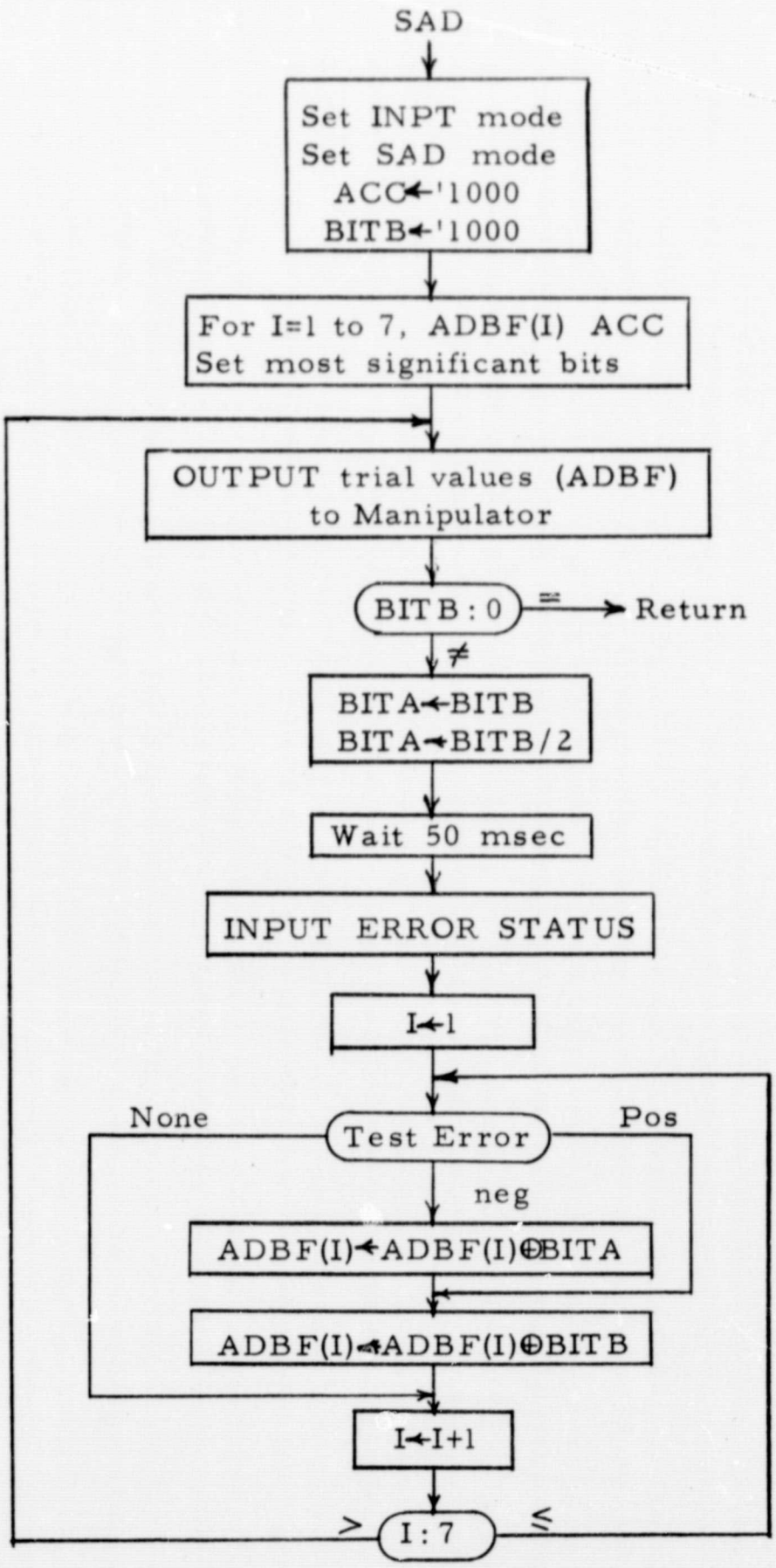


Figure 3-40. SAD Flow Diagram[24]

and 3-41, are used to determine the manipulator position without using the buffer registers as counters. Both of these programs utilize the same technique for determining the final value. The technique selected is a binary search since it gives a fixed conversion time. In general, a trial value cuts the possible range in half and the sign of the error determines which half to further subdivide. As mentioned in Chapter II the fixed conversion is restricted by the delay introduced by the active filters. The SAD subroutine is used when all seven of the major manipulator axes are to be converted. The FSAD routine is used by ETST when an individual axis must be tracked during a computer controlled move. The FSAD routine selectively performs the software A/D conversion. It leaves the computer control mode set so it is not applicable when the manipulator is not moving since the manipulator will try to follow the approximations if the automatic input to the drive is connected.

3.6 Miscellaneous Routines

Several routines were written to aid the operator in performing tasks and to simplify the testing of the equipment.

3.6.1 Position and Status Typeout

The "TYPE" subroutine permits the operator to select various position and status information and print it out on the teletype console. The options available from this routine are detailed in Appendix II.

3.6.2 Adjust Effective Hand Length

An (L) keyin to the TTY exec will permit the operator to specify

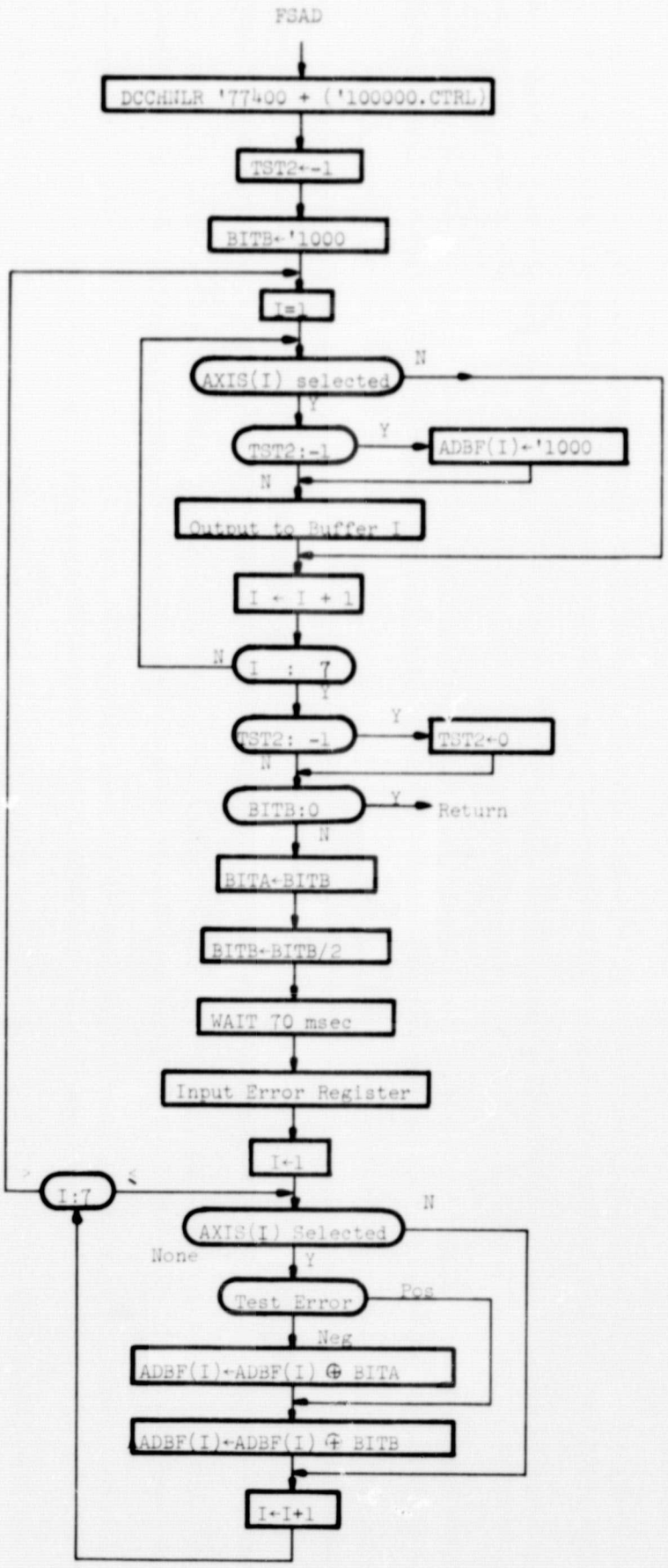


Figure 3-41 Flow Diagram For FSAD

the length of an object in the Hand of the manipulator. This results in the effective hand length being specified to allow for the object. This is applicable for any tool installed. The actual hand length is automatically modified for any tool change indicated by either a (K) keyin or by a tool change block in Automatic mode.

3.6.3 Tool Changing

As mentioned above, it was not possible to implement automatic tool changing due to mechanical restrictions. However, a tool changing routine has been supplied to permit the operator to generate tapes which will request a new tool, and to automatically modify the effective hand length to correspond to the tool fitted.

3.6.4 Control Special Modes

Entries have been supplied through the TTY Exec and, in some cases through the AUTO Exec, to permit the operator to specify certain control functions for the teleoperator system. The options available in both available in both Execs include the modification of the Rate Simulation Mode flag, the control of HR and HG power source, and the status of the CLOSED loop mode bit in the AMCI. In addition, the operator may force the Automatic Mode, set the SAD Mode, or force the Manual Mode through the TTY Exec. These last three operations are intended primarily for system checkout and maintenance.[25]

FOOTNOTES

CHAPTER III

1. See Jon T. Beckett, A Computer Aided Control Technique for A Remote Manipulator, (Report 1-67-44, Digital System Laboratory, Case Institute of Technology), pp. 35-38.
2. Modification of Beckett, p. 39.
3. See Beckett, pp. 40-42.
4. Modification of Beckett, p. 41.
5. Ibid., p. 45.
6. Beckett, p. 49.
7. See Beckett, pp. 48-50.
8. Beckett, p. 51.
9. Ibid., p. 52.
10. Modification of Beckett, p. 57.
11. Beckett, pp. 53-64.
12. Ibid., p. 65.
13. See Beckett, pp. 64-66.
14. Beckett, pp. 66-79.
15. Ibid., pp. 79-91.
16. Modification of Beckett, p. 94.
17. Ibid., p. 95.
18. Ibid., p. 96.
19. See Beckett, pp. 92-98.
20. Beckett, p. 100.
21. Ibid , p. 101.
22. Ibid., p. 102.

23. Ibid., pp. 103-105.

24. Ibid., p. 109

25. See Beckett, pp. 105-108.

CHAPTER IV

SUMMARY AND SUGGESTED FUTURE EXTENSIONS

4.1 Introduction

The initial phases of the Man Machine Communication Project at Case Western Reserve University were to investigate the feasibility of using a small general purpose computer in the control of a remote manipulator. It was the task of this investigator to take the results of the previous work and to expand and modify the system to permit the performance of a specific task. This task, the disassembly of a model of a reactor from a nuclear engine, required that several additional capabilities be added to the system. In addition, the system was expanded to provide a more general base for future work and to increase the control elements available to the operator.

4.2 Achievements

The control system implemented permits several modes of control of the manipulator hardware via the computer. Recent additions permit the operator to partially control the motion of the manipulator while allowing the computer to control the remainder of the system. In addition closed loop control has been implemented for two additional axes and has thus greatly increased the controllability of the manipulator especially in automatic mode. Several additional modifications have been made to the mechanical structure in an attempt to provide more precise positioning where medium to heavy loads are involved. These modifications, the shifting of the position feedback points, have helped. However, the mechanical structure of the manipulator is not rigid

enough, and error is compunded by displacements which cannot be transduced to the control system. These errors include the twisting of the bridge structure when loads are held at the end of an extended arm which can only be removed by a more rigid structure or by techniques that would greatly reduce the application of this device.

4.3 Recommendations

It would appear that one of the most important contributions that can be made to this project is the modification or replacement of the physical manipulator structure to provide a more stable system with which to work. Some additional tasks which would be applicable for future work include:

1. The addition of sensory feedback to the structure. The provision has already been made in the control system through the maskable buffer registers in the AMCI to input this status information when such signals become available.

2. In line with the suggestion of a new or improved physical system, it would appear that an arm of the servo master-slave type would be much more practical for operator manual control. Such an arm could be driven either by the computer in the present manner, or it could be driven in either an absolute or a relative mode under manual control.

3. The current Analog to Digital conversion technique requires a rather long time to obtain the position. Several different techniques are available which would permit a more rapid conversion. In addition, a technique similar to the one used for the HR and HG axes could be utilized. This would permit the quanta size for the system to be greatly reduced without the necessity of using long and expensive D/A Converters.

4. With the refinement of the hardware, it would be possible for the operator to perform more complex

tasks. In this light the operator's performance should be evaluated with various portions of the system to determine the portions which most improve the performance.

5. As mentioned above, the relative hand operations could be extended if the orientation of the Hand with regard to its origin were taken into account. In addition, the facility to generate multiple object files and to cause the program controlled relocation of these files would permit the operator to automatically shift the origin of a particular sequence. This would simplify the generation of repetitive operations which require a common sequence. For this type of operation, an addressable prerecorded sequence library would be indispensable.

4.4 Summary and Conclusions

The use of a small general purpose computer greatly increases the capability of a remote manipulator when the control system for the manipulator has been adapted to accept setpoint type data. The servo control utilized with this system has greatly improved the operator's ability to perform tasks, but a better manual control device would be helpful. It would appear that the master-slave type of control utilizing a mixed incremental and absolute output would provide the best type of manual control. The computer utilized for these operations was rather small for the task assignment because it had only 8192 sixteen bit words. For an actual operating system, it would be possible to remove some of the routines that were utilized with this system. Several different routines of similar nature were used in some cases for comparitive purposes. In general, the time required to perform the calculations for any given operation was insignificant relative to the time required for the device to complete the execution of that operation. In this light, it would be possible to use the same computer to control several manipulators since the majority of the

time is spent by the computer waiting for the operations to be completed. The fitting of the TV camera to the manipulator structure has helped performing manual tasks. In addition, it is much easier for the operator to observe the actions of the manipulator when working at a distance from the viewing window or when the view is partially blocked by the reactor model.

In summary, the use of a computer greatly aids the operator in the performance of tasks which require repetitive or precise motions. The servo-control system has greatly simplified the control problem for both the computer and the operator since it provides proportional control. Mechanical system modifications are desirable before any additional work is performed since the structure itself now restricts the types of operations that can be performed.

BIBLIOGRAPHY

1. Beckett Jon T., A Computer Aided Control Technique For a Remote Manipulator, Report 1-67-44, Digital Systems Laboratory, Case Institute of Technology.

2. Beckett Jon T., Detailed Documentation of the Subroutine Written for the DDP-116 to Control The Case Experimental Manipulator, Digital System Laboratory, Case Institute of Technology.

3. Diederich Norman F., A Computer Aided Teleoperator, a 16mm color motion picture of a reactor disassembly, Digital System Laboratory, Case Western Reserve University, December, 1968.

4. Diederich Norman F., Detailed Hardware Description of Integrated Circuit General Purpose Interface and Auxiliary Manipulator Control Interface, Digital System Laboratory, Case Western Reserve University.

5. Diederich Norman F., Detailed Documentation of Subroutines Written for the DDP-116 to Control the Case Teleoperator System, Digital System Laboratory, Case Western Reserve University.

6. Hammond, P. W., A Hybrid Servo-Positioning System for a Computer Controlled Manipulator, Report 1-66-41, Digital System Laboratory, Case Institute of Technology.

7. Honeywell, Incorporated, Computer Control Division, DDP-116 Programmers Reference Manual, February, 1966.

8. Honewell, Incorporated, Computer Control Division, Interface Manual for the DDP-116, 1965.

9. Honewell, Incorporated, Computer Control Division, S-PAC Instruction Manual (lmc), 1965.

10. Johnsen, Edwin G., and Corliss, William R., Teleoperators and Human Augmentation, NASA SP-5047, Technology Utilization Division NASA, December, 1967.

BIBLIOGRAPHY

11. LaPlatney, C. A. and Gillespie, D. R., Manipulator Wrist Rotation and Hand Grasp Control System, Electrical Engineering Department, Case Western Reserve University, May 1969.

12. Mergler, H. W., et al. Digital Control Systems Engineering, Case Institute of Technology Summer Program Notes, Cleveland, Ohio, 1962.

13. Programmed and Remote Systems Corporation, Operation and Maintenance Manual for G.M.I. Model 100 Manipulator System, St. Paul, 1965.

14. Taylor, Richard J., A Digital Interface for Computer Control of a Remote Manipulator, Report 1-66-49, Digital System Laboratory, Case Institute of Technology.

APPENDIX I

MANIPULATOR COORDINATE SYSTEM, SCALING OF MANIPULATOR VARIABLES AND QUANTIZATION

The manipulator programs assume a cartesian coordinate system for the linear (X, Y, Z) variables. In this system the origin is placed at the center of travel for three manipulator axes. The SP, EP, and WP angles are measured from the vertical (e.g., the negative Z axis). The manipulator has been color coded to facilitate the recognition of directions when in an arbitrary configuration and orientation. This makes it possible to determine the sign of a move from any configuration. The device has been painted so that a motion toward the dark (blue) direction is positive: when the dark side of the limb is up the angle is positive. Likewise, when the light side of the limb (red) is up, the angle is negative: a negative move would be toward the light side. The shoulder rotation is positive when a counter clockwise move is made as viewed from above. The Hand Rotation is divided into 360 increments. A rotation is positive when a clockwise move is made as viewed by one looking toward the wrist pivot from the hand. The Hand Grasp has a zero reference point at the fully closed position. A positive motion will open the hand.

For the seven basic manipulator axes the operator is required to specify variables in inches and degrees as required. For the HR axis the inputs are degrees and turns: while for the HG axis the

PRECEDING PAGE BLANK NOT FILMED.

IMV(AXIS)=MPB(AXIS)-BIAS(AXIS)*SCAL(AXIS)/('1000) (A-I.1)

MPB(AXIS)=FMV(AXIS)*('1000)/SCAL(AXIS) + BIAS(AXIS) (A-I.2)

SCAL(AXIS) represents the value (in inches or fractions of a revolution) of the variable for a change of the manipulator position buffer (MPB) of '1000. BIAS represents the zero point of each axis and is '1000 for all axes except SR. The value of BIAS for SR is '533 which corresponds to alignment of the limbs with the plus X axis. The BIAS and SCAL tables are loaded with the control programs and may be recomputed using an additional program group.

The Hand Rotation angle is obtained by

1. throwing away the upper seven bits;
2. dividing the remaining portion by '100;
3. taking the integer portion of the result and multiplying by 45;
4. adding the remainder from the division as indicated by A-I.3.

ANGLE=('777 AND ANGLE) A-I.3.1

ANGLE=INTEGER(ANGLE/'100)*45 + REMAINDER (ANGLE/'100) A-I.3.2

Grip position is obtained by extending the most significant bit ("4") to the sign bit ("1").

The quantization for the individual manipulator axes can be obtained using the data required to calculate SCAL:

AXIS	QUANTA	RANGE	1 QUANTA
X	'1602	153"	.170"
Y	'1632	135"	.146"
Z	'1646	50.5"	.0540"
SP	'1457	180°	.221°
EP	'625	180°	.444°
WP	'521	180°	.534°
SR	'641	360°	.862°
HR	360	360°	1°
HG	1000	1"	.001"

TABLE A-I.1 QUANTA FOR MANIPULATOR AXES

APPENDIX II

TELETYPE EXECUTIVE CONTROL ROUTINE

A list of the options allowed by the Teletype executive routine (TTYEXEC) is given in Table A-II.1 along with the additional information required for these routines. Distances, angles, and times are specified as inches, degrees, and seconds respectively. Numerical information must be supplied whenever a colon (:) is typed by the Teletype. The number is terminated by a carriage return (CR) or a semicolon (;).

A flow diagram for the TTY Exec is shown in Figure A-II.1.

TABLE A-II.1

TELETYPE COMMANDS

A	Set GPI Output Mode
B	Set GPI Input Mode
D	Set SAD Mode
E	Force Automatic Mode (Set GPI Output Mode, then SAD Mode).
F*	Obstacle Table
	I Initialize table (remove all obstacles)
	T Type contents of obstacle table
	A Name, (LX:, HX:, LY:, HY:, Z) add an obstacle to the table.
	R Name, remove obstacle from the table
G*	Request and accept hand variables; execute hand variables control algorithm (X:, Y:, Z:, WB:, SR) .
H*	Accept mode, axis name, and hand variables; execute hand variable control algorithm

C Center manipulator in hand coordinate system

I, $$X,Y,Z,W,R,U,T$$ Incremental mode

A, $$X,Y,Z,W,R,U,T$$ Absolute mode

Note: T=terminate list; U=reinitialize list; R=SR

I* Control hand grip, rotation and bolt moves

$T,0,1,2,3,4,5,6,7,8,I,P,Q,A,B,C,D,E,F,G,H$
if T specified, $0,1,2,3,4,5,6,7,8$, (TIME:)
if I specified, (Turns per inch:), $P,Q,A,B,
C,D,E,F,G,H$
if P or Q specified (Inches:)
if A through F (G:) then if C through F (T: A:)
if G or H (T:, A:)

Open Loop operation

Note: T--time duration will be specified
0--halt operation
1--close hand
2--open hand
3--rotate CCW
4--rotate CW
5--close hand and rotate CCW
6--close hand and rotate CW
7--open hand and rotate CCW
8--open hand and rotate CW

Closed Loop operation

I--specify turns per inch for threads
P--clockwise rotation and Z move of (INCHES:)
Q--counterclockwise rotation and Z move of
(INCHES:)
A--incremental grasp move (G:)
B--absolute grasp position (G:)
C--incremental grasp, incremental rotate (G:,T:,R:)
D--incremental grasp, absolute rotate (G:,T:,R:)
E--absolute grasp, absolute rotate (G:,T:,R:)
F--absolute grasp, incremental rotate (G:,T:,R:)
G--incremental rotate (T:,R:)
H--absolute rotate (T:,R:)

J* Control rate simulate mode flag ON,OFF

Note: ON(CR) sets rate mode flag
OFF(CR) resets rate mode flag

K* Select tool (Specify desired tool $0,1,2,3,4$)
Note: 0--no tool fitted
1--hook fitted
2--hand fitted
3--socket and short extension fitted
4--socket and long entension fitted

L* Specify length of object in hand (this will adjust the effective hand length) (L:)

M* Accept mode, axis name, and manipulator variables; execute manipulator variable control algorithm

C Center manipulator in manipulator coordinate system

I, $$X,Y,Z,S,E,W,R,U,T$$ Incremental mode

A, $$X,Y,Z,S,E,W,R,U,T$$ Absolute mode

Note: T=terminal list; U=reinitialize list; R=SR

N* Coordinate Transformation Reference Point Data

P Print coefficients of transformation matricies

R, (X:,Y:,Z:), $0,1,2,3$ Specify position and reference point number

S, $0,1,2,3$ Specify reference point (use hand position)

T Compute transformation matrix

P* Punch paper tape for automatic mode

G Compute present hand variables and punch (absolute)

H, (S:,Y:,Z:,W:,R:) Specify desired hand variables and punch (absolute)

I, $T,0,1,2,3,4,5,6,7,8,I,P,Q,A,B,C,D,E,F,G,H$
if T specified, $0,1,2,3,4,5,6,7,8$, (TIME:)
if I specified, (Turns per inch:), P,Q,A,B,C,D,E,F,G,H
if P or Q specified (Inches:)
if A through F (G:) then if C through F (T:, A:)
if G or H (T:, A:)
Punch hand and grip control block (see TTY Exec "I" command for notation)

J, ON,OFF punch rate mode control block as SET or RESET

K, $0,1,2,3,4$ Punch tool change request block

L, (X:,Y:,Z:,W:,R:,S:,E:) Specify desired manipulator variables and punch

M Input present manipulator variables and punch

N, (X:,Y:,Z:) Specify object origin and punch

O Computer present hand position and punch origin block

R Compute present hand variables and punch (relative)

S, (X:,Y:,Z:,W:,R:) Specify desired hand variables and punch (relative)

T Punch TTY Exec return code

U, (X:,Y:,Z:,W:,R:) Punch incremental relative hand move (X,Y,Z will be transformed W and R will not be transform when executed)

V, (DIST:) Punch straight line path control block

W, (T:) Punch wait block (T=wait time)

X, ON,OFF punch hand power control block

Y, ON,OFF punch CLOSED LOOP MODE control block

Z, (S1:,S2:,E1:,E2:,S3:,S4:,E3:,E4:) Specify grid table constraint modification and punch (see TTY Exec "Z" command for notation) Punch ignore code

Q Repeat execution of last block

R Transfer control to AUTO Exec

S Perform Software A/D conversion

T* $0,1,2,3,4,5,6,7,H,M,A,X,Y,Z,G,R,S,T$ Type variables and status information

Note: 0--Exact contents of ten bit manipulator buffers
1--FMV,FHV buffers, scaled
2--IMV,IHV buffers, scaled
3--FMV buffer, scaled
4--FHV buffer, scaled
5--IMV buffer, scaled
6--IHV buffer, scaled
7--FMV buffer and TMAX, scaled
A--absolute axis indentification header, ten bit buffers
M--manipulator axis identification header, IMV,FMV,scaled
H--hand axis identification header, IHV, FHV, scaled
X--grip and hand status and position (G+R+S+T)
Y--same as X plus M
Z--same as X plus H
G--grip size
R--hand angle in degrees
S--status of hand axes in octal
hand axis status word
bit 111111
123456
11 "1" if Grip SCR power connected
12 "1" if Grip error is zero
13 "1" if Rotation SCR power connected
14 "1" if Rotation error is zero
15 "1" if Rotation index flag on
16 not used
T--tool fitted

U* (TRACE FLAGS: See Appendix V for list of flags and output, !)
Note: list terminated by !

V* Straight line path control (DIST:)

W Transfer stored FHB to FHB and fill IMV and IHV (used when operator intervention is requested from obstacle avoidance routines)

X* ON,OFF control SCR drive switch for HR and HG
Note: ON(CR) enables SCR drives, OFF(CR) disables SCR drives

Y* ON,OFF control CLOSED LOOP mode flag
Note: ON(CR) sets flag and OFF(CR) resets flag

Z* (S1:,S2:,E1:,E2:,S3:,S4:,E3:,E4:) Specify and apply grid table constraint parameters.

S1 ≤ S2 < S3 ≤ S4, specify allowed regions of shoulder pivot angle (degrees. There are two regions S1-S2 and S3-S4. If only one region is desired it should be S3-S4. To delete the first region type (N) when S1 is requested. If (N) is typed when S3 is requested, there will be no constraints applied beyond S2. If both S1 and S3 are (N), no constraints are applied.

E1 ≤ E2 specifies the limits of elbow pivot travel for S1-S2. E3 ≤ E4 specifies the limits for S3-S4. If E1 or E3 is (N) then no constraint will be applied against EP in the indicated region. If E2 or E4 is (N), only the region from -EPmax to (E1 or E3) will be inhibited in the specified region.

*Additional information must be specified by the operator

Legend: (subroutine requests required data)
$ one additional command is necessary $
$$ an axis name, then a position or increment must be specified, list is terminated by a special command $$

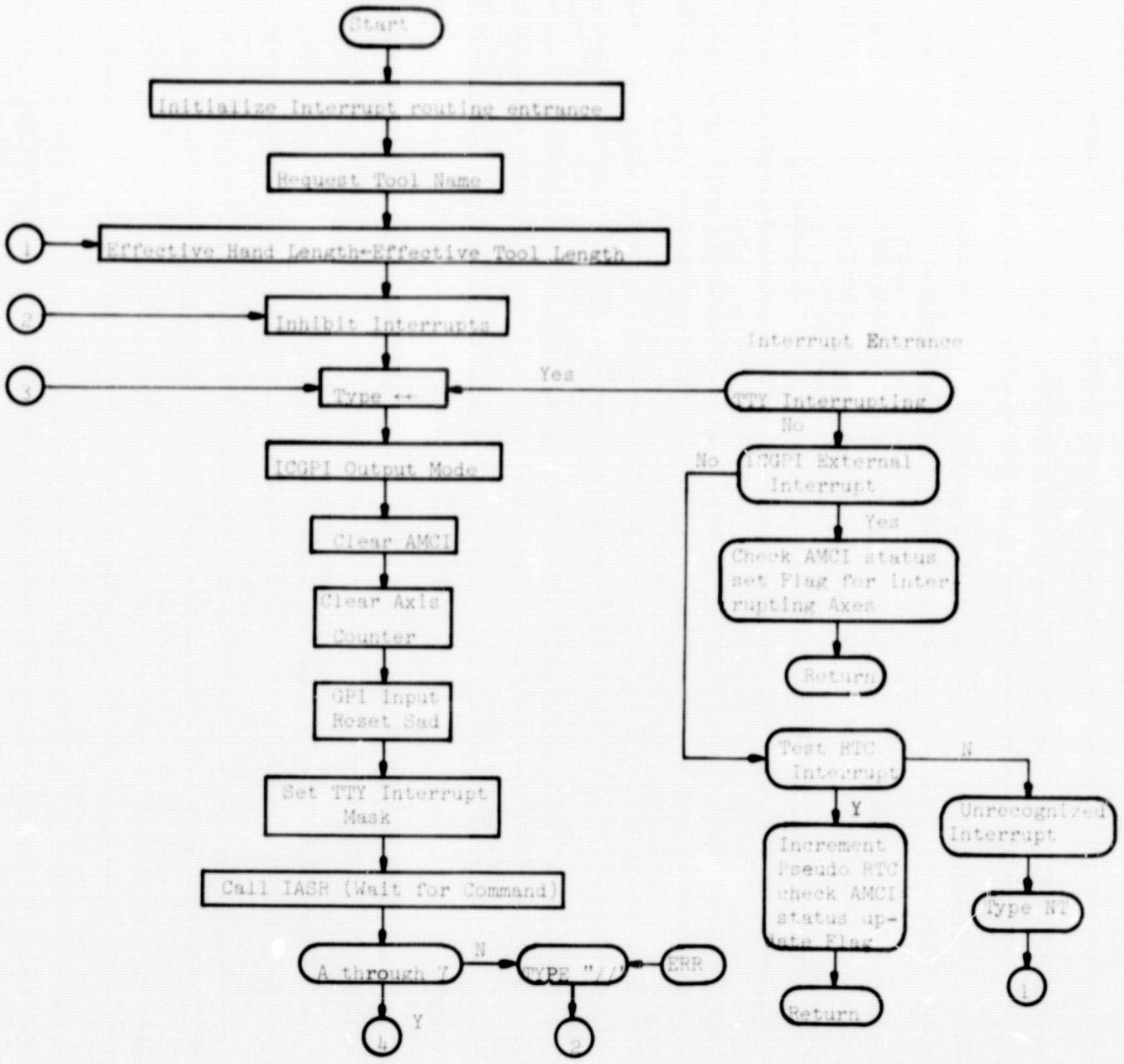


Figure A-II.1a TTY Exec Flow Diagram

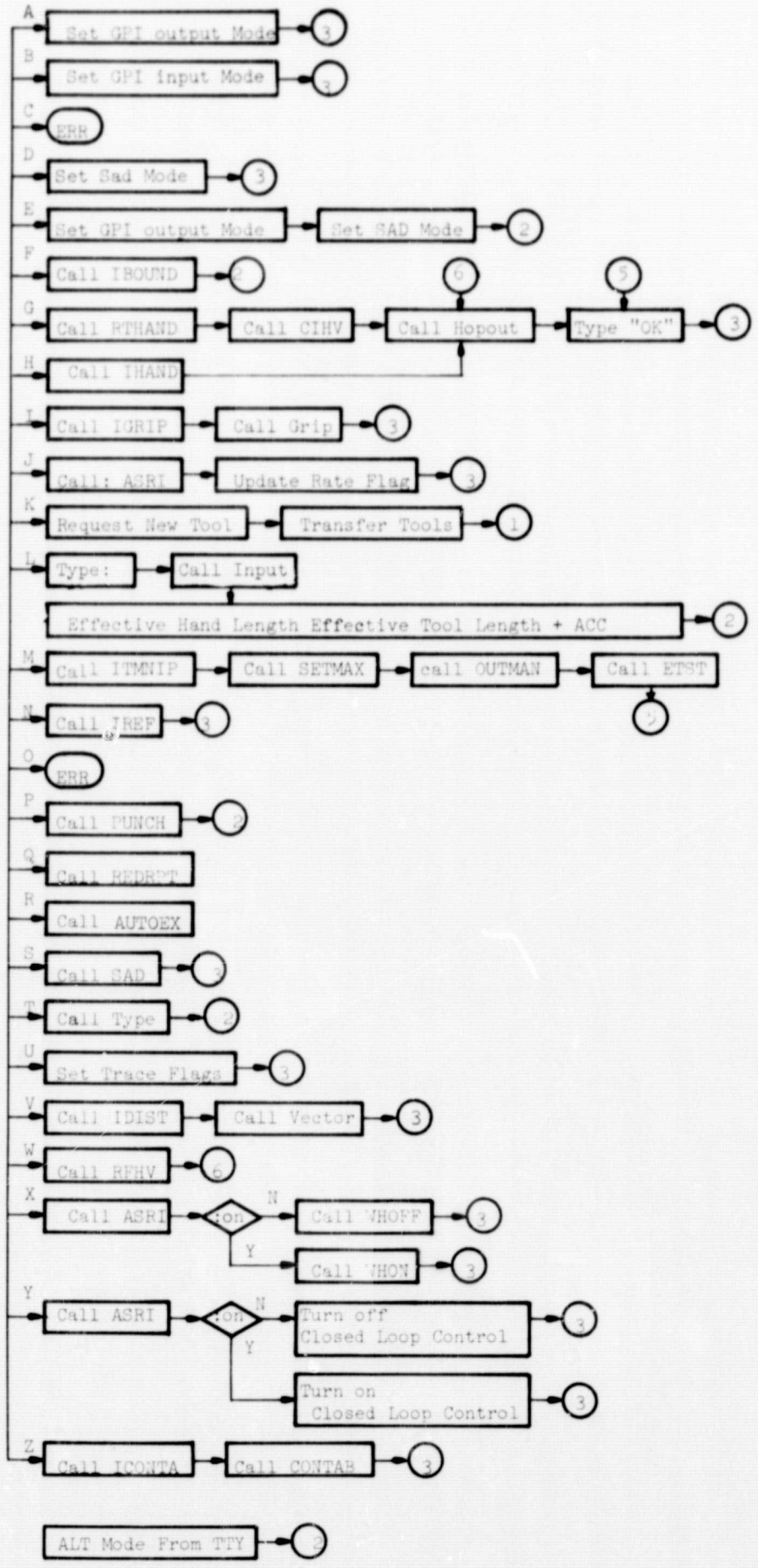


Figure A-II. Ib TTY Exec Flow Diagram Continued

APPENDIX III

SUBROUTINE LIBRARY FOR EXPERIMENTAL COMPUTER AIDED TELEOPERATOR SYSTEM

This section contains a list of the subroutines written to implement control functions and algorithms described in Chapter III. An extensive set of input/output routines has been provided for the teletype and for General Purpose Interfaces to facilitate information transfer between the operator and the manipulator via the computer. These routines are coded in the DAP assembly language used by the DDP-116 machine and utilize only fixed point arithmetic operations. The restriction to fixed point arithmetic increases the computation speed and reduces the memory requirements for the operating system. The current operating system requires '15200 words of program storage and an additional '1555 words of table storage to give a total storage requirement of '16755 words. A detailed description of these routines appears in separate reports.(2) (5)

TABLE A-III.1

ROUTINE SUMMARY

TTY Exec:*	a. Accept commands from operator via TTY console
	b. Enable TTY interrupts during execution of algorithms
	c. Route program control to correct interrupt routine on occurrence of a priority interrupt
AUTOEX:	Control fully automatic mode
	a. Read paper tape

b. Transfer data from paper tape to proper buffer

c. Transfer control to proper operating routine

REDRPT: Return to AUTOEX without reading a new paper tape block and reexecute the last block read

VECTOR: Control path of manipulator so that it approximates a straight line along a line defined by an extension of the centerline of the hand

GRIP:* Control hand grip and hand rotation (open loop) also hand grip and hand rotation (closed loop) and control hand rotation and Z axis move for bolt operations.

Punch:* Request or obtain desired commands as specified by TTY input and record data on paper tape for later execution by AUTOEX.

SAD: Perform software A/D conversion for the seven basic manipulator axes.

FSAD:* Perform software A/D conversion for selected axes while not disturbing unselected axes.

IREF: Accept reference points for object coordinate system, compute transformation matrix, or type coefficients of transformation matrix

RELHND: Transform relative hand variables from object coordinate system to manipulator coordinate system

IMVR: Input exact manipulator coordinates and scale to 16 bit computer words (see Appendix I)

CIHV: Call IMVR, then compute present hand variables

FBUF: Scale 16 bit computer words into ten-bit manipulator positions (see Appendix I)

OMANIP: Output the ten-bit manipulator positions to the manipulator buffers

SETMAX: Compute approximate travel time based on initial and final manipulator variables

RATE: Use initial and final manipulator variables and transit time to output fine increments to manipulator for rate mode simulation

ETST:*	Wait for manipulator to reach final variables. Check flag set by ICGPI and update manipulator position buffers as required. If transit time is exceeded by a significant amount, operator assistance is requested.
OUTMAN:	Call RATE if rate mode flag is set; otherwise call FBUF and OMANIP.
HOPOUT:	Control hand position; optimize terminal manipulator configuration; and avoid obstacles.
GRID:*	Optimize terminal manipulator configuration, fill GRID time table, and compute TMAX. Takes into account mechanical constraints imposed by Wrist Pivot Constraint Table and program constraints generated by CONTAB.
CFES:	Use Grid time table pointer to compute FSP and FEP, then call CFMV.
CFMV:	Use FSP, FEP, FWP, and FSR to compute FMX, FMY, and FMZ.
FSRT:	Use azmuth and ISR and find FSR to minimize rotation while not exceeding rotation limits.
IBOUND:	Add obstacle bounds to table, initialize table, delete an obstacle or type effective bounds of all obstacles in the table.
PHCK:	Call RDTC and DTCT to test for possible collisions--if none are detected, output variables to manipulator.
TXYZ:	Test X,Y, and Z positions with obstacle bounds.
DTCT:	Test for collisions between manipulator and an obstacle when the normal output mode is used.
RDTC:	Test for collisions between manipulator and an obstacle when the rate simulation output mode is used.
TEST:	Given a set of manipulator variables, compare positions of shoulder, elbow, wrist pivots, and hand with the obstacle bounds.
FLAGGT:	Flag current time value in Grid time table and search table for a new minimum time value--call DFES to find new manipulator variables.

ISCT:	Test line connecting initial and final hand positions for intersection with the X and Y obstacle bounds.
SUBG:	Test possible paths over and around an obstacle; select the one requiring the least approximate transit time.
CADS:	Use manipulator angular variables and length of arm limbs to compute projection of the upper arm, forearm, and hand on the X, Y, and Z axes.
COSX2:	Compute cosine of specified angle.
SINX2:	Compute sine of specified angle.
ATNX2:	Compute angle (between -1/8 revolution and +1/8 revolution) which satisfies specified arctangent.
SQRX2:	Compute square root of specified number.
DIV:	Divide and round result.
SRND:	Round result of multiplication or shift operation.
TOOLS:*	Transfer tool specifications.
CONTAB:*	Apply constraint parameters to Wrist Pivot Constraint Table.
ICGPI:*	Check status of Auxiliary Manipulator Control Interface.
IRTC:*	Increment pseudo RTC and call ICGPI.
RECONV:*	Transform X,Y, and Z according to transformation matrix generated by IREF.
PUT:*	Output data to specified axis register under control of AMCI.
MPUT:*	Disable RTC and ICGPI interrupts and call PUT, enable RTC and ICGPI interrupts before return.
GET:*	Input data from specified axis register under control of AMCI.

MGET:*	Disable RTC and ICGPI interrupts and call GET, enable RTC and ICGPI before returning.
WHON:*	Enable SCR drive for Hand Rotate and Hand Grasp.
WHOFF:*	Disable SCR drive for Hand Rotate and Hand Grasp.
GRIN:*	Input grip position from AMCI and extend sign.
ZONE:	Try to find path around obstacle.
TOPTST:	Test path broken in three pieces for obstacle intersections.
TOPOUT:	Output path checked by TOPTST.
EDGE:	Attempt to generate path around edge of obstacle.
TRAC:	Input trace flags.
TFLG:	Test trace flags.
FTME:	Call GRID and then find time for move.
FWZI:	Form increments for wrist and Z moves.
FFHV:	Form move sequence around corner of an obstacle.
CANG:*	Compute relative angle of Wrist Rotation between zero and three hundred and fifty-nine degrees.
RTHAND:	Request and accept terminal hand variables.
RTMNIP:	Request and accept terminal manipulator variables.
RXYZ:	Request and accept X,Y, and Z variables (stored in FHV buffer).
ITOOL:*	Request and accept new tool number.
ITHAND:*	Accept mode and then accept axis and position or increment as appropriate for hand moves.
ITMNIP:*	Accept mode and then accept axis and position or increment as appropriate for manipulator moves.
IDIST:	Request and accept distance for vector subroutine.
IGRIP:*	Accept hand grip or rotation command for either open loop or closed loop control. Also accept commands and parameters for bolt moves.

ICONTA:*	Request and accept parameters for Wrist Pivot Constraint Table restrictions.
ASRI:*	Input "ON" or "OFF" and convert to flag for mode control.
RFHV:	Restore FHV buffer from SAVE buffer in case computer asks operator for help.
SPCI:*	Request and accept Shoulder Pivot constraint limit and convert to scaled value.
EPCI:*	Request and accept Elbow Pivot constraint limit and convert to scaled value.
GANG:*	Compute actual Hand Rotation Angle.
FRME:*	Compute hand rotation move error.
CXYZ:	Compute X, Y, and Z positions for Elbow pivot, Wrist pivot and Hand.
HGINC:*	Output Hand Grasp moves as an increment.
HGOUT:*	Output absolute Hand Grasp move.
HRINC:*	Output incremental Hand Rotation move.
HROUT:*	Output absolute Hand Rotation move.
RSEP:*	Read one character from TTY, test for N if not N the call INPTT.
IANG:	Accept angular values as degrees, convert to fractions of a revolution and scale.
GRPIN:*	Request and accept grip size (check in range).
ROTIN:*	Request and accept Hand Rotation commands (check within range).
IASR:*	Input one character from the TTY with interrupts inhibited. Simulate an interrupt if SS4 is set while waiting, or if the character is ALT MODE.
INPT:	Accept mixed fraction or integer via TTY and scale as specified.
INPTT:*	Special call on INPT to permit examining first character before doing conversion.

OUTO:	Type an octal number on the TTY (suppress leading zeros).
OUTP:	Output scaled mixed fraction TTY.
TANG:	Convert angular variables to degrees and type value.
TYPE:*	Type routine with input selection to determine information to be typed.
TYPF:	Type contents of FMV and TMAX buffers.
TIMV:	Type contents of IMV buffer.
TIH:	Type contents of IHV buffer.
TFHV:	Type contents of FHV buffer.
TYPI:	Type contents of IHV and IMV buffers.
CRLF:	Type a carriage return followed by a line feed.
T2CR:	Call CRLF, then type two characters.
OUT2:	Output two characters from the accumulator.
MSSG:	Type a message on the TTY.
TOLPRI:*	Print out tool name.
CLSET:*	Initialize RTC to 1/60 second and enable RTC interrupt.
ICOFF:*	Mask off ICGPI and RTC interrupts.
ICON:*	Mask on ICGPI and RTC interrupts.
IMOF:*	Mask off ICGPI.
IMON:*	Mask on ICGPI interrupts.
MAOF:	Reset TTY interrupt mask to inhibit TTY interrupts.
MAON:	Set TTY interrupt mask to allow TTY interrupts.

*Programs which are either new or have been modified since Beckett.(1)

APPENDIX IV

ERROR MESSAGES FROM PROGRAMS IN EXPERIMENTAL COMPUTER AIDED TELEOPERATOR SYSTEM

The routines in the experimental computer aided Teleoperator System check input and results to minimize the number of improper actions that will occur. Table A-IV.1 contains a list of the error messages that result from improper conditions as well as the origin of the error message and the meaning of the message.

TABLE A-IV.1

PROGRAM ERROR MESSAGES

ERROR	ROUTINE	MEANING
NT (mask,A)	TTYEXEC	unrecognized interrupt
//	TTYEXEC	improper input
RD	AUTOEX	parity, buffer overflow, illegal flag
PN	PUNCH	illegal specification
GP	GRIP	negative time, turn overflow
//	IGRIP	illegal command
MO	FBUF	10 bit buffer overflow
IM (WP,EP,SP)	HOPOUT	initial position of indicated pivot inside obstacle
CE	KINT (grid)	overflow
WL	GRID	wrist angle out of range

OP	CMMT(grid)	overflow
RH	RELHND	overflow
//	TYPE	illegal command
SG	SUBG	IHZ & FHZ above obstacle
IR	IRTE	rate out of range
XF	IREF	unrecognized command
RC	RDCT	overflow
VC	VECTOR	out of bounds move
IH	CIHV	overflow
RO	RATE	overflow
TM	SETMAX	overflow from FSAD
HELP←+-ERROR	FSAD	bad input
HELF! +-ERROR	SAD	bad input
DV(return address)	DIV	divide overflow
IB	IBOUND	illegal option, table full, no obstacles
//	INPT	bad input
OT	OUTP	improper calling sequence
FM	CFMV	overflow
AI	ITHAND, ITMNIP	improper option
//	ITHAND, ITMNIP	bad input

//	IANG	bad input
IM	IMVR	scale overflow
IS	ISCT	overflow, table limit hit
//	INPO	bad character
//	ROTIN	angle out of range
//	GRPIN	grip out of range
GN	HGOUT, HGINC	negative grip position
HA	HROUT, HRINC	hand angle overflow
//	EPCI	out of range input
//	SPCI	out of range input

APPENDIX V

TRACE FLAGS FOR DEBUGGING SOFTWARE AND HARDWARE

In order to permit the operator to obtain additional information from the manipulator programs without using external debugging routines, a system of TRACE flags and print routines has been incorporated which will indicate the routines entered and some of the temporary results generated in these routines. Table A-V.1 lists the trace flags and the routines which these flags affect. In addition the data output by these routines is listed. To enter a trace flag for inspection the operator will type a "W": the trace flags are then requested and accepted. The list of trace flags is terminated by an exclamation point (!). To initiate trace printout, the operator SETS sense switch 2 on the main console. Setting sense switch 1 will terminate the printout of any long lists, but does not suspend the trace option. If SS1 is set on the next entry, the operator will be questioned whether he still wants it to be set.

TABLE A-V.1

TRACE FLAGS AND MEANING

FLAG	ROUTINE	OUTPUT
IC	ICONTA	SPVR, ERV1, ERV2 constraint table parameters
CT	CONTAB	modified wrist pivot constraint table
FM	FBUF	MBUF manipulator output buffer
OM	OUTMAN	FMV, TMAX scaled output buffer and time

SE	TTIA(grid)	TSP, TEP
KT	KINT(grid)	KO-K5
BN	BXYZ(grid)	XYZ time bounds
CM	CMMT(grid)	optimized times
GT	GTBL(grid)	complete grid time table
RH	RHND(relhnd)	X,Y,Z before and after, FHV
SG	SUBG	FMV, TACC
OB	DTCT, RDCT	name DTCT or RDCT
OB	CXYZ	XYZ:
PH	CXYZ	pivot positions
VC	VECT(vector)	ΔX,ΔY,ΔZ & increment X,Y,Z
IM	IMVR	octal info from manipulator buffers
SG	FTME	FHV, TACC
IS	ISCT	X,Y,Z at intersection with obstacle boundary.

APPENDIX VI

SAMPLE PROGRAM FOR AUTOMATIC MODE:

REMOVAL OF UPPER ENDBELL AND CORE FROM REACTOR MODEL

To indicate some of the features incorporated in the manipulator system a sequence of operations was designed and recorded on paper tape to permit the removal of the endbell from the reactor model once the location of the model had been determined by the system with the aid of the operator. Following its removal, the endbell is deposited at a prespecified location, and the manipulator then moves in to remove the core. Once removed, the core is placed at another preprogrammed location. The manipulator is then available for the next operation.

This preprogramming was performed by generating a sequence of operations. Since the model was available, the manipulator was run through the operation utilizing both the semi-automatic and the manual mode. The necessary moves and positions were recorded in relative form when they concerned the location of the model. They were recorded in absolute form when they concerned the desired destination points for the elements of the disassembled model. This entire operation could have been performed without operating the manipulator if one were given a detailed set of drawings which included the dimensions of the reactor. In performing these operations certain physical limitations of the manipulator structure were considered. The wrist pivot constraint table was modified at different stages in the operation.

This modification forced the manipulator to go into configurations which would lend themselves to the sequence of operations to follow. If the move had been optimized for just the specified operation,in several instances, the next operation would have been unrealizable.

Table A-VI.1 lists the basic operations that are required for the disassembly program and the punch command utilized in performing them.

Table A-VI.2 lists the operations that the operator must perform in one way or another in order to use this prepunched tape.

TABLE A-VI-1

PREPROGRAMMED OPERATIONS FOR ENDBELL AND CORE REMOVAL

TASK	COMMAND
Request origin data	PO
Request that HOOK be installed	PK (1)
Initialize HOOK orientation and opening	PIE (G:200,T:1,A:90)
Establish constraints on physical configurations	PZ (S1 N,S3 90,S4 N,E3 75,E4 N)
Position over eyebolt with hook above hookeye and wrist pivot angle slightly positive (.5°)	PR (Hook had been moved to desired position).
Position hook behind hookeye and set wrist pivot angle to -2.5°	PU (X:0,Y:-2,Z:-1,5,W:-3,R:0)
Open HOOK to 1.5 inches	PIB (G:1500)
Position HOOK into eyebolt	PU (X:0,Y:2,Z:0,W:2.5,R:0)
Close HOOK to 1 inch	PIB (G:1000)
Position structure down 0.3 inches.	PU (X:0,Y:0,Z:-.3,W:0,R:0)
Close HOOK to 0.7 inches	PIB (G:700)
Lift bell fifteen inches	PV (DIST: -15)
Move bell clear of reactor by moving sixty inches in the negative X direction (RELATIVE TO MODEL) also rotate 90 degrees and set WP slightly positive (.5°)	PR (after making the desired move the endpoint was recorded in a relative format)
Remove artificial constraints	PZ (S1 N,S3 N)

TASK	COMMAND
Position endbell over final position	PG (after doing an absolute hand move to X:-80, Y:42)
Lower endbell to floor	PG (after doing an incremental Hand move of Z:-62.25)
Open HOOK to 1.5 inches	PIB (G:1500)
Move HOOK back 5 inches from eyebolt	PG (after making an incremental hand move of X:5)
Apply constraints on ARM position to permit reaching into reactor for CORE removal	PZ (S1 N,S3 105,S4 N, E3 105,E4 135)
Position HOOK above damping rods and slightly inside dimensions of reactor	PR (after making desired move, including setting WP angle at 0.5°)
Move HOOK above and slightly behind eyebolt on CORE	PR (after making desired move)
Move HOOK down and behind eyebolt, swing WP angle to -2.5 degrees to permit moving the HOOK for the final connection	PU (X:2.2 Y:0,W-2.5,R:0)
Close HOOK in several stages and position structure to prevent using HOOK motor to lift CORE	PIB (G: 1000) PU (X:0,Y:0,Z:-.3,W:0,R:0) PIB (G:700)
Lift CORE out of reactor housing	PV (DIST: -39)
Modify arm constraints to force desired configuration on lift	PZ (S1 N,S3 105,S4 N,E3 120,E4 150)
Move CORE clear of reactor housing	PU (X: -60,Y:0,Z:0,W:0,R:0)
Remove arm constraints	PZ (S1 N,S3 N)

TASK	COMMAND
Move to desired final position and rotate to 180 degrees	PG (after first doing an absolute hand move to X:-80,Y:-11,R:180)
Set core down on the floor and straighten WP angle to zero.	PG (after making an incremental hand move of Z:-63.2,W:-60)
Open HOOK	PIB (G:1700)
Move HOOK back 5 inches to clear eyebolt	PG (after first making an incremenental hand move of X:5)
Position structure in area of viewing window for operator inspection	PG (after moving to desired area)
Return control to TTYEXEC	PT

TABLE A-VI.2

OPERATIONS REQUIRED BY OPERATOR TO USE PROGRAM OF TABLE A-VI.

1.	Remove Hook and withdraw actuating rod into hand assembly, zero grasp counter.
2.	Insert program tape in high speed papertape reader.
3.	Enter AUTOEX by typing R.
4.	Program will request origin information.
5.1	If the origin coordinates are known, they may be entered using the NR option of the TTYEXEC.
5.2	If the origin coordinates are known, then the following sequence of operations can be used to obtain them.
5.2.1	Restrict arm position by using Z (S1 N,S3 90, S4 90, E3 90,E4 N)
5.2.2	Indicate that no tool is fitted by using K (0)
5.2.3	Set WP to zero HA (W:0,T)
5.2.4	Position actuating rod over center of eyebolt in cover of endbell. Lower arm until rod just touches eyebolt and is centered over center of endball. Enter this position as point "0" using NS (0).
5.2.5	Enter this position except add eight inches to the Z axis as the third reference point. This can be done with (HI Z:8,T; NS (3)).
5.2.6	Return Z to original value and position over reference point #2, extend the actuating rod to 2.85 inches (IB (G:2850)), when this point has been located, indicate it with (NS (1)).
5.2.7	Same as 5.2.6, except for point 3.
5.3	Calculate transformation matrix with NT.
6.	Move manipulator back into area of operator to facilitate tool installation.
7.	Reenter AUTOEX with a R keyin.
8.	Device will request that HOOK be installed and will return to TTYEXEC.

9. Operator installs HOOK and checks origin for Hand Rotation and Hand Grasp axes.

10. Reenter AUTOEX with R keyin.

11. Watch operation as endbell and core are removed from reactor model. Program will come back to these operations have been completed.

APPENDIX VII

DISCRETE GENERAL PURPOSE INTERFACE

This section will contain the instructions and connections necessary for use of the general purpose interface. Drive requirements and data formats are included. The general timing for the data and control signals is also included.

The connections required for GPI I/O are listed in Tables A-VII.2 and A-VII.3. These pin numbers refer to Winchester MRAC-50S connectors. The cable used for these connectors should be twisted pair with each signal having its associated ground line grounded in the device. The cable used for this application should be no more than twenty-five feet long.

A block diagram of the GPI is given in Figure A-VII.1; input line termination is indicated in Figure A-VII.2, Signal timing is indicated in Figures A-VII.3, A-VII.4 and A-VII.5. For data transfers and status information the zero volt level is the "1" or true condition; the minus six volt level is the "0" or false condition. This is indicated by following the signal name with a minus (-) sign. If the name is followed by a plus (+) sign, the logical meaning is reversed.

TABLE A-VII.1

DISCRETE GENERAL PURPOSE INTERFACE COMMANDS

OCP	'0040	Input Mode Reset Buffer, Reset Ready
OCP	'0140	Output Mode, Set Ready
OCP	'0240	OCP02-
OCP	'0340	Enable DMC
OCP	'0440	OCP04-
OCP	'0540	OCP05-
OCP	'0640	OCP06-
OCP	'0740	Reset DMC Stop Flip-flop
SKS	'0040	Skip if buffer ready
SKS	'0140	Skip if EXT01 grounded
SKS	'0240	Skip if EXT02 grounded
SKS	'0340	Skip if EXT03 grounded
SKS	'0440	Skip if GPI not interrupting
SKS	'0540	Skip if EXT05 grounded
SKS	'0640	Skip if EXT06 grounded
SKS	'0740	Skip if EXT07 grounded
OTA	'0040	Output to GPI buffer from A if Ready
INA	'0040	Or GPI buffer to A if Ready
INA	'1040	Input GPI buffer to A if Ready

The signals labeled EXTXX refer to the inputs made available to the external device which permits the sensing of external conditions. They are tested by using the SKS XX where the numbers XX are the same, and will give a skip if the line is grounded. If the line is at minus six volts, the SKS will not skip.

The DIPXX- signal is used to DROP IN the data on the INDNN lines if the device is in Input Mode. The SRDYX- line is used to indicate that the External Device is finished with the buffer.

The SHIFT- line is used to shift the buffer from the External Device and will result in a one bit shift toward the next higher numbered bit position. A zero will be shifted into the left (bit "1") end of the buffer.

The OTDNN- lines refer to the outputs of the buffer. These lines are always available although the signal may not be meaningful during a data transfer to the buffer.

The INDNN- lines are the inputs to the buffer from the external device.

The IFRDY- line will be grounded when the Ready flip-flop is set.

For further details one is referred to Taylor.[14]

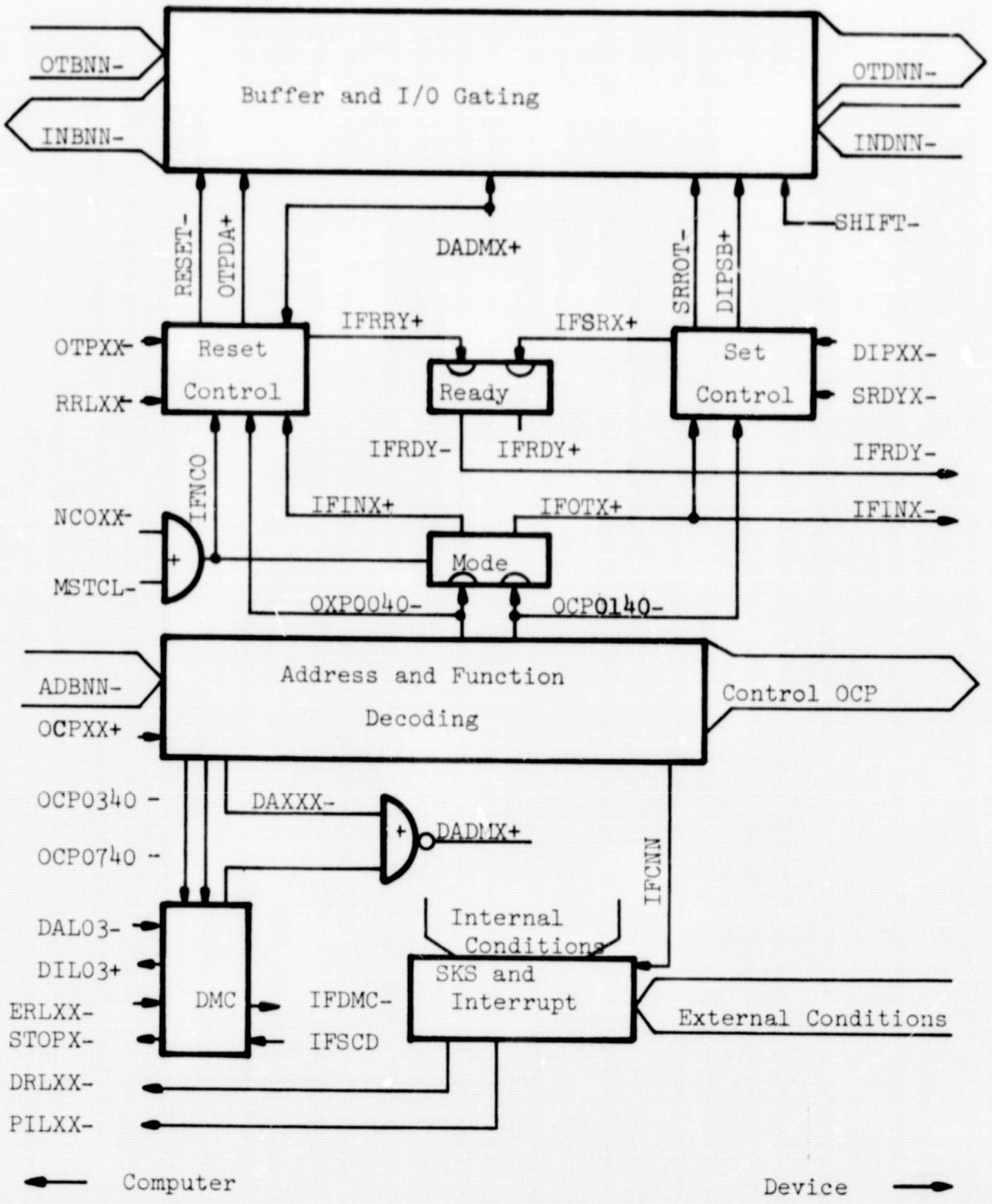


Figure A-VII.1 GPI Block Diagram [1]

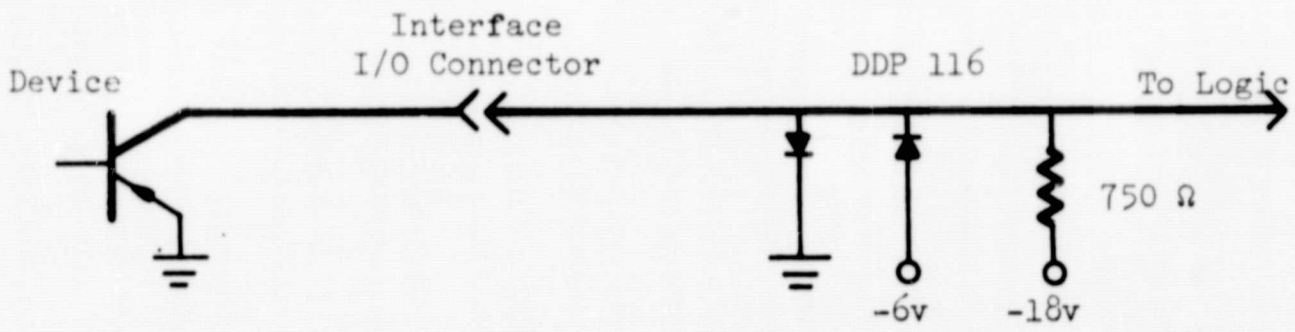


Figure A-VII.2 Input Line Termination[2]

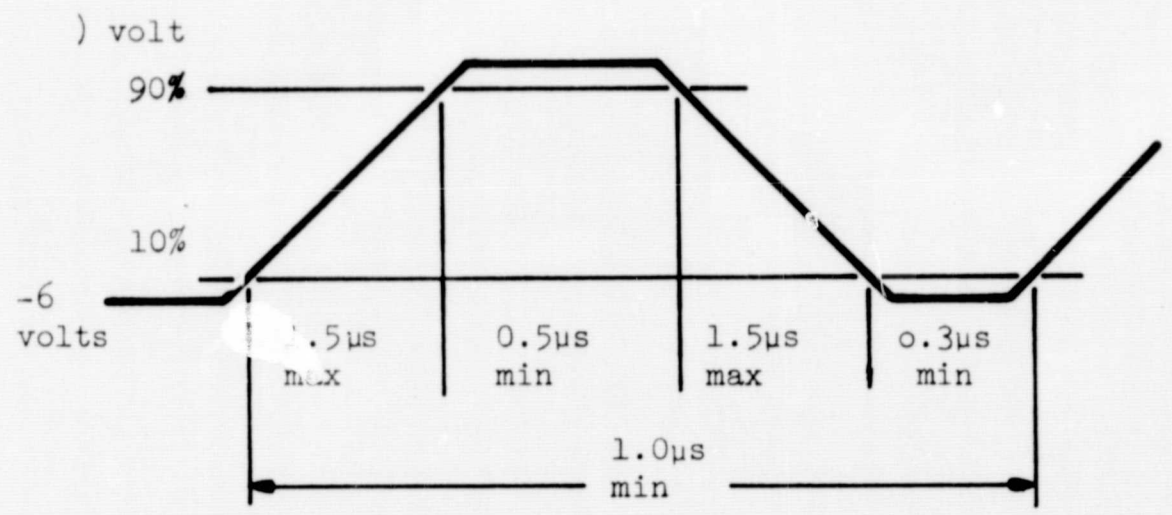


Figure A-VII.3 DIPXX-, SRDYX-, SHIFT- Requirements[2]

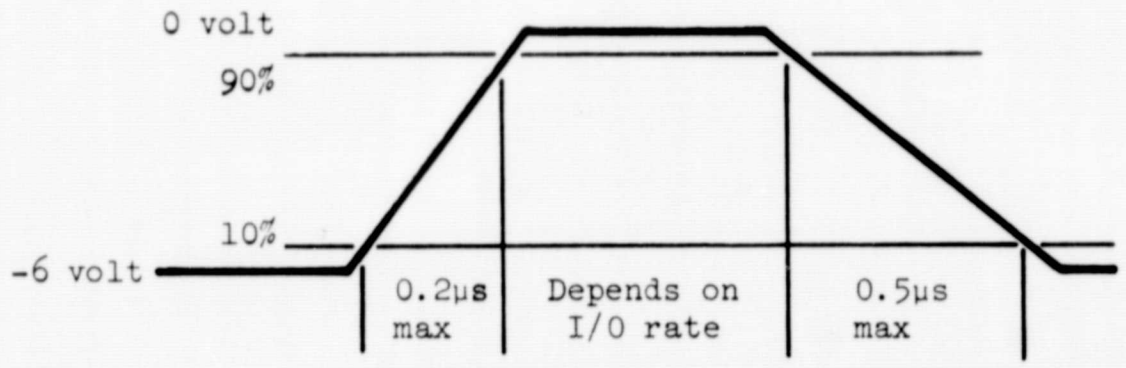


Figure A-VII.4 OTDNN-, IFRDY-, IFDMC-, IFINX+ Characteristics[3]

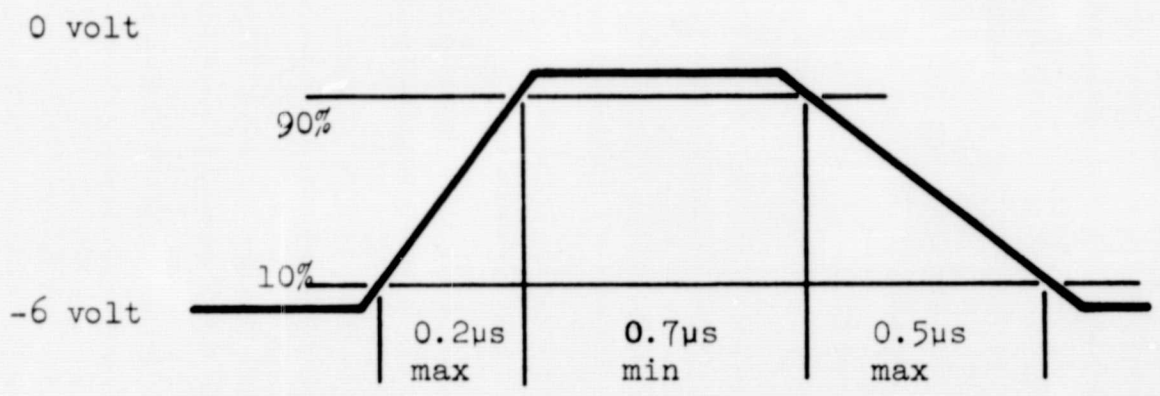


Figure A-VII.5 OCPNN- Characteristics[3]

TABLE A-VII.2^4

GPI I/O CONNECTOR G

PIN	SIGNAL	PIN	SIGNAL
A	EXTØ4-	e	OTDØ6-
B	EXTØ4-GND	f	OTDØ6-GND
C	OCPØ2-	h	INDØ6-
D	OCPØ2-GND	j	INDØ6-GND
E	IFSCD-	k	OTDØ5-
F	IFSCD-GND	m	OTDØ5-GND
H	OCPØ4-	n	INDØ5-
J	OCPØ4-GND	p	INDØ5-GND
K	SRDYX-	r	OTDØ4-
L	SRDYX-GND	s	OTDØ4-GND
M	OCPØ5-	t	INDØ4-
N	OCPØ5-GND	u	INDØ4-GND
P	SHIFT-	v	OTDØ3-
R	SHIFT-GND	w	OTDØ3-GND
S	OCPØ6-	x	INDØ3-
T	OCPØ6-GND	y	INDØ3-GND
U	DIPXX-	y	OTDØ2-
V	DIPXX-GND	AA	OTDØ2-GND
W	OTDØ8-	BB	INDØ2-
X	OTDØ8- GND	CC	INDØ2-GND
Y	INDØ8-	DD	OTDØ1-
Z	INDØ8-GND	EE	OTDØ1-GND
a	OTDØ7-	FF	INDØ1-
b	OTDØ7-GND	HH	INDØ1-GND
c	INDØ7-		
d	INDØ7-GND		

TABLE A-VII-3[4]

GPI I/O CONNECTOR H

PIN	SIGNAL	PIN	SIGNAL
A	EXTØ1-	e	OTD14-
B	EXTØ1-GND	f	OTD14-GND
C	EXTØ2-	h	IND14-
D	EXTØ2-GND	j	IND14-GND
E	EXTØ6-	k	OTD13-
F	EXTØ6-GND	m	OTD13-GND
H	IFDMC-	n	IND13-
J	IFDMC-GND	p	IND13-GND
K	EXTØ7-	r	OTD12-GND
L	EXTØ7-GND	s	OTD1w-GND
M	IFRDY-	t	IND12-
N	IFRDY-GND	u	IND12-GND
P	EXTØ3-	v	OTD11-
R	EXTØ3-GND	w	OTD11-GND
S	IFINX	x	IND11-
T	IFINX-GND	y	IND11-GND
U	EXTØ5-	z	OTD1Ø-
V	EXTØ5-GND	AA	OTD1Ø-GND
W	OTD16-	BB	IND1Ø-
X	OTD16-GND	CC	IND1Ø-GND
Y	IND16-	DD	OTDØ9-
Z	IND16-GND	EE	OTDØ9-GND
a	OTD15-	FF	INDØ9-
b	OTD15-GND	HH	INDØ9-GND
c	IND15-		
d	IND15-GND		

FOOTNOTES

APPENDIX VII

1. Richard J. Taylor, A Digital Interface for Computer Control of a Remote Manipulator, (Report 1-66-49, Digital System Laboratory, Case Institute of Technology), p. 24.

2. Ibid. p. 20.

3. Ibid. p. 18.

4. Ibid. p. 21-22.

APPENDIX VIII

INTEGRATED CIRCUIT GENERAL PURPOSE INTERFACE

The ICGPI is in many ways similar to the GPI except that 1. it does not have internal shift capability and 2. the DMC channel has not been connected to the DMC hardware of the computer. However, the ICGPI does have fully decoded OCP and SKS instructions. In addition the logic levels are ground for "1" or true and are plus five volts for "0" or false. The connector to use for this interface is 1. an ELCO 100-000-004 for a plug if it is desired to build a special cable, or 2. an ELCO 100-000-007 receptical if the existing cable can be used. If a cable is constructed it should be made using twisted pair cable and should be no longer then twenty five feet.

The instructions applicable for this device are given in Table A-VIII.1. The connections are listed in Table A-VIII.2. All inputs to the interface have been terminated in eighty-two ohms to plus five volts. Outputs should be terminated in the same manner. Figure A-VIII.1 shows an example of a typical circuit and is applicable in either direction. For data transfers to the interface buffer from an external device, an automatic set-ready is generated when a drop in signal (DIPXY-) is given. In general any TTL power driver gate will do as a line driver for these lines.

TABLE A-VIII.1

INSTRUCTIONS FOR INTEGRATED CIRCUIT GENERAL PURPOSE INTERFACE

OCP	'0037	Set Input mode, Reset Ready
OCP	'0137	Set Output mode, Set Ready
OCP	'0237	OCP02-
OCP	'0337	Enable DMC (not connected), OCP03-
OCP	'0437	OCP04-
OCP	'0537	OCP05-
OCP	'0637	OCP06-
OCP	'0737	Reset DMC EOR (not connected), OCP07-
OCP	'1037	OCP10-
OCP	'1137	OCP11-
OCP	'1237	OCP12-
OCP	'1337	OCP13-
OCP	'1437	OCP14-
OCP	'1537	OCP15-
OCP	'1637	OCP16-
OCP	'1737	OCP17-
OTA	'0037	Output to ICGPI buffer if Ready Set.
INA	'0037	Input to A from ICGPI buffer if Ready Set. (contents of buffer ORed to contents of A)
INA	'1037	Input to A from ICGPI buffer if Ready Set. (clear A register)

SKS	'0037	Skip if Ready Set.
SKS	'0137	Skip if EXT01 grounded
SKS	'0237	Skip if EXT02 grounded
SKS	'0337	Skip if EXT03 grounded
SKS	'0437	Skip if ICGPI interrupting
SKS	'0537	Skip if EXT05 grounded
SKS	'0637	Skip if EXT06 grounded
SKS	'0737	Skip if EXT07 grounded
SKS	'1037	Skip if Ready
SKS	'1137	Skip if EXT11 grounded
SKS	'1237	Skip if EXT12 grounded
SKS	'1337	Skip if EXT13 grounded
SKS	'1437	Skip if EXT14 grounded
SKS	'1537	Skip if EXT15 grounded
SKS	'1637	Skip if EXT16 grounded
SKS	'1737	Skip if EXT17 grounded

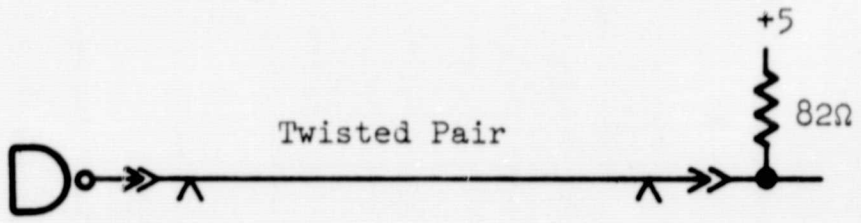
TABLE A-VIII.1 CONTINUED

TABLE A-VIII.2

I/O CONNECTOR SIGNAL PINS

SIGNAL	PIN	SIGNAL	PIN	SIGNAL	PIN	GROUND PIN
INF01-	B2	OTF01-	B4	OCP15-	A5	A3
INF02-	B6	OTF02-	B8	OCP13-	A9	A7
INF03-	B10	OTF03-	B12	OCP12-	A13	A11
INF04-	B14	OTF04-	C13	OCP11-	C15	A15
INF05-	C1	OTF05-	C3	OCP07-	D4	D2
INF06-	C5	OTF06-	C7	OCP06-	D8	D6
INF07-	C9	OTF07-	C11	OCP05-	D12	D10
INF08-	E3	OTF08-	F2	OCP04-	F4	E1
INF09-	E7	OTF09-	F6	OCP03-	F8	E5
INF10-	E11	OTF10-	F10	OCP02-	F12	E9
INF11-	D14	OTF11-	E13	xxx	F14	E15
INF12-	G3	OTF12-	H2	xxx	H4	G1
INF13-	G13	OTF13-	H12	EXT16-	H14	G15
INF14-	J3	OTF14-	K2	EXT15-	K4	J1
INF15-	J13	OTF15-	K12	EXT14-	K14	J15
INF16-	L3	OTF16-	L5	EXT12-	M2	L1
READY-	L9	OCP17-	K6	EXT11-	K8	L7
IFSCD-	L15	OCP16-	L13	EXT17-	K10	L11
SRDYX-	M4	OCP14-	S5	EXT06-	S7	M6
DIPXX-	M8	OCP10-	S9	EXT05-	S1	M10
IFINX-	M12	xxx	S13	EXT13-	S15	M14
EXTIN-	T2	xxx	S1	EXT03-	S3	U1

IFDMC-	U3	xxx	T4	EXT02-	T6	U5
DACLR-A	U7	xxx	T8	EXT01-	T10	U9
IFSFX-	U11	xxx	T12	EXT07-	T14	U13



Signetics
N8855A

Figure A-VIII.1 Typical Input/Output Line

The INF signals are the inputs to the ICGPI data buffer from the external device. They are strobed by the DIPXX- signal when the interface is in INPUT mode as indicated by IFINX- (at zero volts). The suffix- on a signal indicates that it is true or "1" when the signal is at zero volts, and false or "0" when the signal is at plus five volts.

The OTFXX signals are the outputs of the individual bits of the ICGPI data buffer and are enabled at all times. The number XX corresponds to the numbering of the bits of the A Register.

The OCPXX- signals are external control pulses which can be used for any function desired by the device builder.

The EXTXX- lines are available to the device builder for use when it is desired to sense an external condition. EXTIN- is the external interrupt input and can be sensed with an SKS '0437 if the ICGPI (bit 7) is masked on. In addition the Ready and the DMC EOR are connected to the interrupt line. The DMC hardware is included on the control card but the channel assignment has not been made at this time. Therefore, the DALXX and DILXX lines are not connected. These details and others can be found in a separate document.(4)

APPENDIX IX

INSTRUCTIONS AND DATA FORMATS FOR THE AUXILIARY MANIPULATOR CONTROL INTERFACE AND THE MANIPULATOR CONTROL INTERFACE

The interfaces which have been constructed between the Manipulator and the two computer interfaces have a number of special instructions and data formats. This section is intended to bring all of these formats together in order to simplify the programming of these devices.

Table IX-1 lists the instructions which are applicable for the ICGPI and the GPI when the Manipulator hardware is connected. Figures A-IX.1 and A-IX.2 give the formats for the registers in both the MCI and the AMCI.

TABLE A-IX.1

OCP and SKS Instructions

OCP	'0037	Enable ICGPI Input mode, Reset Ready.
OCP	'0137	Enable ICGPI Output mode, Set Ready.
OCP	'0237	Clear AMCI, (Reset Ready Response Inhibit; Set Integrated Circuit Channel Select Register to '100000; Clear Discrete Logic Override Register; Clear HR and HG error registers, and SCR drive enabling Memories; and Reset FORCE mode).
OCP	'0337	Enable DMC (Not used)
OCP	'0437	Inhibit sensor clock.
OCP	'0537	Enable sensor clock.

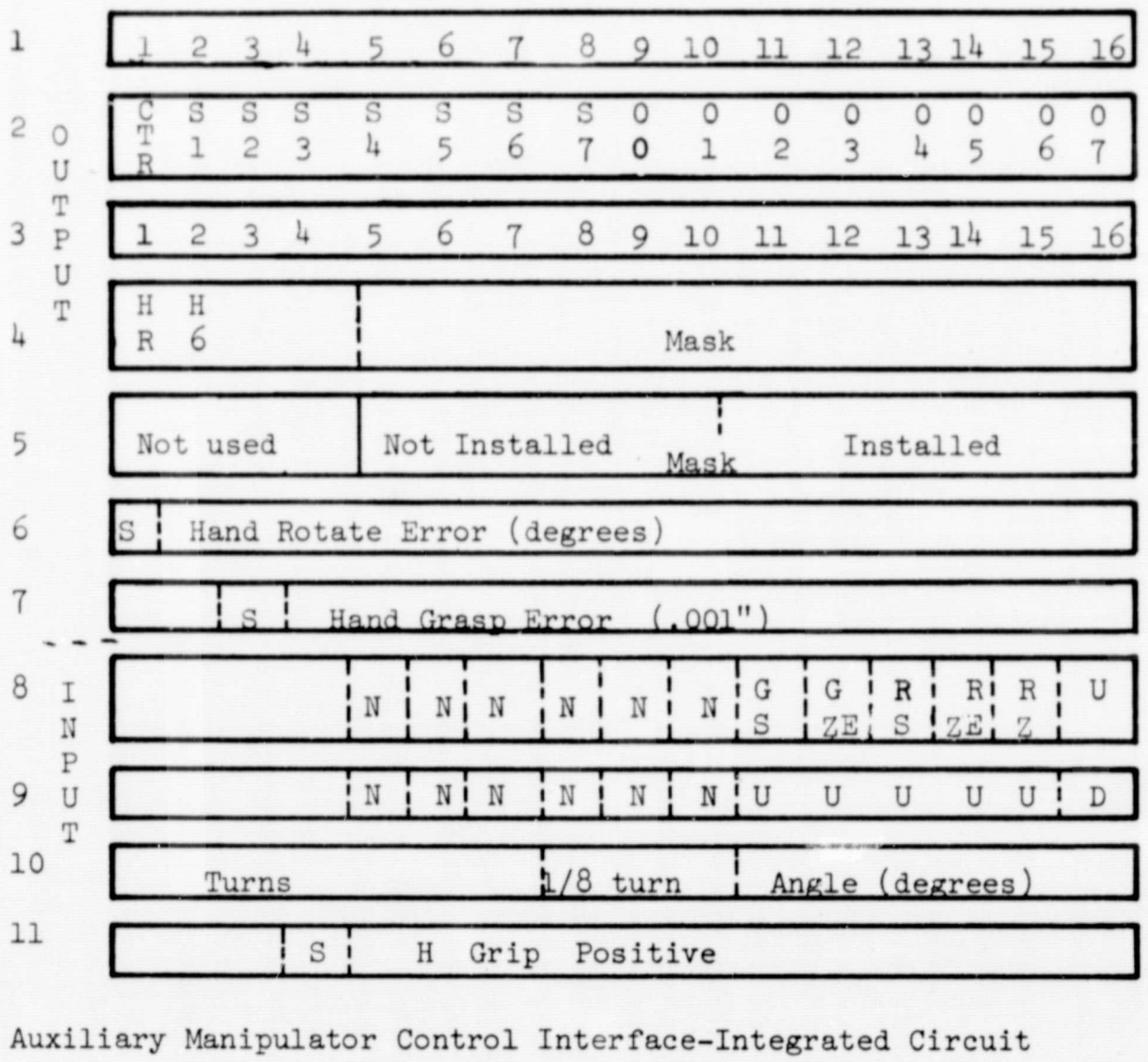
OCP	'0637	Reset FORCE mode Flip-flop, Reset Ready Response Inhibit.
OCP	'0737	Reset DMC EOR flip-flop. (not used)
OCP	'1037	Set FORCE mode Flip-flop
OCP	'1137	Set DCCHNL if ICGPI in Output Mode (enable load of DCCHNLR on next OTA '0037), Reset Ready Response Inhibit.
OCP	'1237	Set ICCHNL if ICGPI in Output Mode (enable load of ICCHNLR on next OTA '0037), Reset Ready Response Inhibit)
OCP	'1337	Inhibit Ready Response (inhibits DIPXX and SRDY, will make one transfer before inhibiting these transfers)
SKS	'0037	Skip if ICGPI Ready.
SKS	'0437	Skip if ICGPI interrupting.
SKS	'0537	Skip if External Status Registers are interrupting.
SKS	'0637	Skip if External Status Register "1" is set (HG and HR status information).
SKS	'0737	Skip if Manual Control Override Switch down on operator's control pannel.
SKS	'1037	Skip if ICGPI Ready.
SKS	'1N37	Skip if External Status Register assigned to MCI axis N (N=1-7) is set.
OCP	'0040	Enable GPI INPUT Mode, Reset READY Immediately force Manual Mode
OCP	'0140	Enable GPI OUTPUT Mode, set READY Enable Computer Control Mode if no error (20 cps clock)
OCP	'0240	Set SAD Mode Clear AXIS Counter, Reset START Set Computer Control Mode if GPI in OUTPUT Mode
OCP	'0340	Enable DMC Mode

OCP	'0440	Set AXIS Counter to one, Set START Caution: First clear AXIS counter, set Manual Mode or Computer Control Mode
OCP	'0540	Reset SAD Mode
OCP	'0640	If GPI in INPUT Mode, transfer ERROR STATUS to GPI data buffer (transfer specified data in Force Mode) If GPI in OUTPUT Mode, transfer GPI data to the four-bit register (transfer specified data in Force Mode)
OCP	'0740	Reset DMC End of Transmission
SKS	'0040	Sense GPI READY
SKS	'0140	Sense no error
SKS	'0240	Sense Computer-Control Mode
SKS	'0440	Sense GPI not interrupting

Note* G.P. INTERFACE ADDRESS = '40
MASK BIT = 8
DMC CHANNEL = 3

---

*See Programmer's Reference Manual (7) for explanation.



Auxiliary Manipulator Control Interface-Integrated Circuit

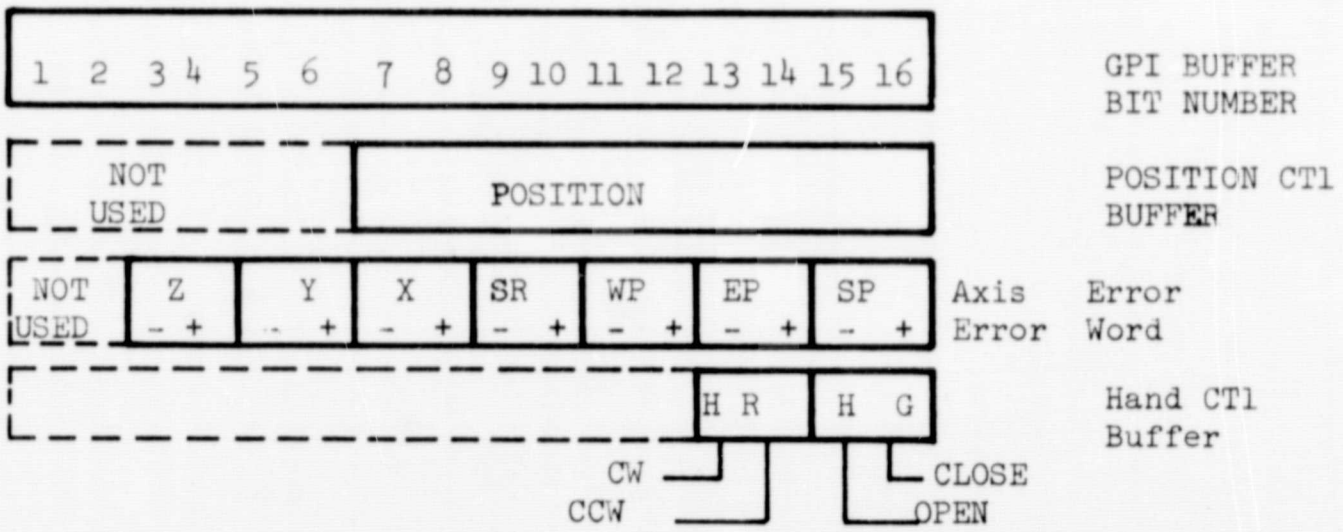
General Purpose Interface Formats

Figure A-IX.1

1. ICGPI Buffer Bit Number
2. DCCHNLR
3. ICCHNLR
4. Channel '1'
5. Channels "2"-"8"
6. Channel "9"
7. Channel "10"
8. Channel "1"
9. Channels "2"-"8"
10. Channel "9"
11. Channel "10"

WORD FORMATS

APPENDIX IX



Data Transferred

Axis Ctr	(Input Mode) to GPI Buff	(Output Mode) from GPI Buff
0	Error	Hand Control
1	SP	SP
2	EP	EP
3	WP	WP
4	SR	SR
5	X	X
6	Y	Y
7	Z	Z

A-IX.2 Manipulator Control Interface—GPI Buffer Data Formats[2]

FOOTNOTES

APPENDIX IX

1. Jon T. Beckett, A Computer Aided Technique for a Remote Manipulator, (Report 1-67-44, Digital System Laboratory, Case Institute of Technology), p. 22.

2. Ibid., p. 23.