

AD/A-006 555

**CLOSED LOOP REAL-TIME EVALUATION OF
MISSILE GUIDANCE AND CONTROL COMPONENTS:
SIMULATOR HARDWARE/SOFTWARE
CHARACTERISTICS AND USE**

A. C. Jolly, et al

CODE Research Corporation

Prepared for:

Army Missile Command

August 1974

DISTRIBUTED BY:

NTIS

**National Technical Information Service
U. S. DEPARTMENT OF COMMERCE**

AD A00 6555

FINAL TECHNICAL REPORT

U. S. ARMY MISSILE COMMAND

CONTRACT NO. DAAH01-74-C-0193

**CLOSED LOOP REAL-TIME EVALUATION OF
MISSILE GUIDANCE AND CONTROL COMPONENTS:
SIMULATOR HARDWARE/SOFTWARE CHARACTERISTICS
AND USE**

BY

**A. C. JOLLY
L. H. HAZEL
R. J. VINSON**

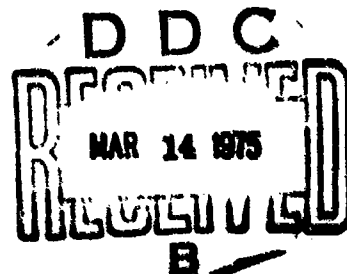
**Reproduced by
NATIONAL TECHNICAL
INFORMATION SERVICE
US Department of Commerce
Springfield, VA. 22151**

REPORT NO. 74/3

**Approved for public release, distribution
unlimited.**

**CODE RESEARCH CORPORATION
1363 S. STATE COLLEGE BLVD.
ANAHEIM, CALIFORNIA 92806**

AUGUST 1974



CONTENTS

<u>SECTION</u>		<u>PAGE</u>
1.0	INTRODUCTION	1-1
2.0	COMPUTER EQUIPMENT	2-1
2.1	Digital	2-1
2.1.1	Central Processor Unit	2-1
2.1.2	Input/Output	2-1
2.1.3	Magnetic Core Storage	2-6
2.2	Analog	2-6
2.2.1	Analog Components	2-6
2.2.2	Logic Components	2-20
2.2.3	Summary	2-27
2.3	Analog/Digital Interfaces	2-31
2.3.1	Signal Transfer	2-31
2.3.2	DAC and ADC	2-31
2.3.3	Control and Sense Lines	2-33
2.3.4	Interrupts	2-33
2.3.5	Clocks (Interval Timers)	2-33
2.4	Angular Motion Simulator	2-34
2.4.1	Hardware Characteristics and Axes Systems	2-34
2.4.2	Phase Compensation	2-38
2.4.3	Simulation Hardware Protection	2-38
2.5	Target Simulators	2-41
2.5.1	Angular Relationships	2-41
2.5.2	Simulation of Target Closure	2-46
2.5.3	Hardware Characteristics	2-50
2.5.4	Control and Patch Panels	2-51
3.0	SOFTWARE	3-1
3.1	Monitor/Operating System	3-1
3.2	Compilers and Assemblers	3-4
3.3	Hybrid Computer Control and I/O Routines	3-5
3.4	CSSL	3-6
3.5	Graphics Software	3-7
3.6	Utility Routines	3-10

II

SECTION**PAGE**

4.0	EXAMPLES HYBRID SIMULATION PROBLEM	4-1
4.1	Mathematical Model Description	4-1
4.1.1	Vehicle Geometry, Mass and Inertia	4-1
4.1.2	Engine Thrust	4-1
4.1.3	Axis Systems and Reference Frames	4-2
4.1.4	Aerodynamic Forces and Moments	4-2
4.1.5	Translation and Rotation Equations	4-3
4.1.6	Guidance and Control	4-4
4.1.7	Target Intercept	4-5
4.2	Partitioning of the Problem	4-7
4.3	Design of the Digital Program	4-9
4.4	Analog Programming and Patching Diagrams	4-21
4.5	Results	4-21
5.0	REFERENCES	5-1
	APPENDIX A Program Listing	

III

LIST OF FIGURES

<u>FIGURE</u>		<u>PAGE</u>
2.1	Sigma 5 System Configuration	2-2
2.2	Integrator/Summer Patching	2-7
2.2A	Logic Board Comcor 5000 for High Gain and Network Borrow Patching	2-8
2.3	Multiplication/Division Patching	2-10
2.4	Analog and Logic Patching Arrangement for a Resolver	2-11
2.5	Resolver 1R Mode Inputs and Outputs	2-12
2.6	Resolver 2R Mode Inputs and Outputs	2-13
2.7	Rate Mode and Sine/Cosine Generator Inputs and Outputs	2-14
2.8	Comparator Patching	2-16
2.9	DA Switch Patching	2-17
2.10	Potentiometer Patching, Servo Manual	2-19
2.11	Delay Flip-Flop Logic Patching	2-21
2.12	General Purpose Flip-Flop Logic Patching	2-21
2.13	RS Flip-Flop in JK Mode	2-22
2.14	Logic Board for External Clock Control	2-23
2.15	CI-500 DAC and ADC Patch Connections	2-24
2.15A	Integrator Mode Control Logic	2-27
2.16	Pitch Channel Frequency Response	2-36
2.17	Yaw Channel Frequency Response	2-37
2.18	Analog Computer Mechanization of Lead Filter	2-39

LIST OF TABLES

<u>TABLE</u>		<u>PAGE</u>
I	Elements of the Sigma 5 Digital Computer System	2-3
II	DIO Characters (8 bits)	2-4
III	Logic Board Addresses and ADC/DAC Control	2-25
IV	CI-500 Clock Patching	2-26
V	Trunkline Identification between CI-5000 and CI-500	2-29
VI	Analog/Digital Interface Elements	2-32
VII	Angular Motion Simulator Performance Specifications	2-35
VIII	Mirror Correction Voltages for Zoom Position	2-48
IX	Patch Board Contents	2-57
X	Patchboard Operational Amplifier Connections	2-58
XI	HRBM Operator Keys	3-3
XII	Graphics Software File Storage	3-10
XIII	Aerodynamic Functions	4-2
XIV	NAHELIST Input to the Digital Program	4-13
XV	DAC and ADC Usage	4-15
XVI	Potentiometer Settings	4-21
XVII	Sample Pitch Plane Hybrid Simulation Results	4-31

V

<u>FIGURE</u>		<u>PAGE</u>
2.19	Analog Computer Mechanization of Gyro Angle Run-Abort Control	2-40
2.20	Mirror 1 Angles and Screen Axes	2-42
2.21	Target-Missile Line of Sight Angles	2-42
2.22	Laser Energy Density Variation with Aperture Position	2-45
2.23	Simulator Layout (Horizontal Plane)	2-49
2.24	Mirror Frequency Response in the Vertical (Y) Axis	2-52
2.25	Mirror Frequency Response in the Horizontal (X) Axis	2-53
2.26	Dual Gimbal Mirror Calibration - Up/Down	2-54
2.27	Dual Gimbal Mirror Calibration - Left/Right	2-55
4.1	Seeker Block Diagram	4-4
4.2	Pitch Autopilot and Actuator	4-6
4.3	Partitioned Problem	4-10
4.4	Frame Time Usage	4-11
4.5	Subroutine Linkage in the Program	4-14
4.6	Program Overall Flow Chart	4-16
4.7	Rotational Pitch Rate Analog Diagram	4-23
4.8	Pitch Euler Angle	4-24
4.9	Gyro Angle	4-25
4.10	Angle of Attack	4-26
4.11	Pitch Autopilot	4-27
4.12	Pitch Actuator	4-28
4.13	Analog Variables as Functions of Time (1)	4-29
4.14	Analog Variables as Functions of Time (2)	4-30

1.0 INTRODUCTION

The simulation facility currently operational at the Guidance and Control Directorate, Missile Research Development and Engineering Laboratory, U. S. Army Missile Command consists of a hybrid digital/analog computer system interconnected to a pair of target simulators and to a three degree of freedom angular motion simulator. The hybrid computer comprises a medium scale digital computer, as the controlling element of the facility, interfaced to a pair of analog computers; target simulators permit the simulation of laser designated targets and optically tracked targets.

By placing guidance and control hardware components on the angular motion simulator which is driven to simulate the closed loop missile angular response, and arranging for the target simulator to generate the appropriate closed loop line of sight angles and angular rates, the performance of the actual flight hardware can be studied and evaluated. This requires, of course, a real-time solution of the missile flight equations to be provided by the hybrid computer.

Descriptions of the simulation facility hardware components, their performance characteristics and methods of utilization are scattered in a diversity of reference and instruction manuals which do not lend themselves to yielding a rapid overall grasp of the capability and operation of the facility. The intent of this report is to remedy this situation by providing a summary of the list of components which comprise the facility, a brief description of their characteristics and methods of use and any other salient points of particular interest to the potential user. It should be noted that the hybrid computer software components are considered essential parts of the simulation facility.

A large part of the information contained in Sections 2 and 3 is available elsewhere, and references are given wherever appropriate. However, some of the information is not generally available in written form.

Finally, an example hybrid simulation program is given which consists of a missile intercepting a target with both restricted to planar motion (3 degrees of freedom). A discussion of the method of solution of the problem is given to illustrate the use of the hybrid computer and to indicate a general approach to setting up a real-time simulation.

2.0 COMPUTER EQUIPMENT

2.1 Digital

The digital section of the hybrid computer facility consists of a Xerox Sigma 5 digital computer system. The system configuration is shown in Figure 2.1, and a list of the system components, including Xerox model numbers and brief performance specifications, is given in Table 1. The table does not include details of the analog computer interfaces; these are given in Section 2.3 below.

2.1.1 Central Processor Unit

The operational environment within the CPU is controlled by a 64 bit Program Status Word (PSW), which allows rapid switching of the environment in response to external stimuli or under internal program control. The central processor instruction set includes control logic for execution of word (32 bits), double-word (64 bits), half-word (16 bits), and byte (8 bits) oriented processor commands. Additionally, a floating point unit handles arithmetic on single precision (32 bits) and double-precision (64 bits) operands. Floating point format comprises a biased 8 bit exponent, with base 16, at the most significant end of the word or double-word, and a 24 bit or 56 bit mantissa.

General purpose registers, which are used for all arithmetic, logic and indexing operations, consist of blocks of 16 32-bit words. The current configuration contains 4 register blocks (numbered 0, 1, 2, 3) with a capability of expanding to 16. Register block usage is controlled by the Register Pointer bits in the PSW.

All computer instructions occupy 32 bits with a 17 bit address field which allows direct access to the maximum memory address range of 0 to 131,071 words. Addressing modes include direct, indexed (using 1 index register), indirect (1 level), immediate, and combined indexed and indirect (post-indexing).

The system contains two real-time clocks with basic clock rates of 8000 per second and 500 per second respectively (i.e. resolutions of .125 and 2 milliseconds). Clock interrupts may be generated such that a user-chosen multiple of the basic rates may be maintained. Details of the methods of use and programming at the CPU-hardware level are given in Reference 1.

2.1.2 Input/Output

As indicated in Figure 2.1 input/output operates through three paths connecting magnetic core memory, with the I/O devices.

FIGURE 2.1 Sigma 5 System Configuration

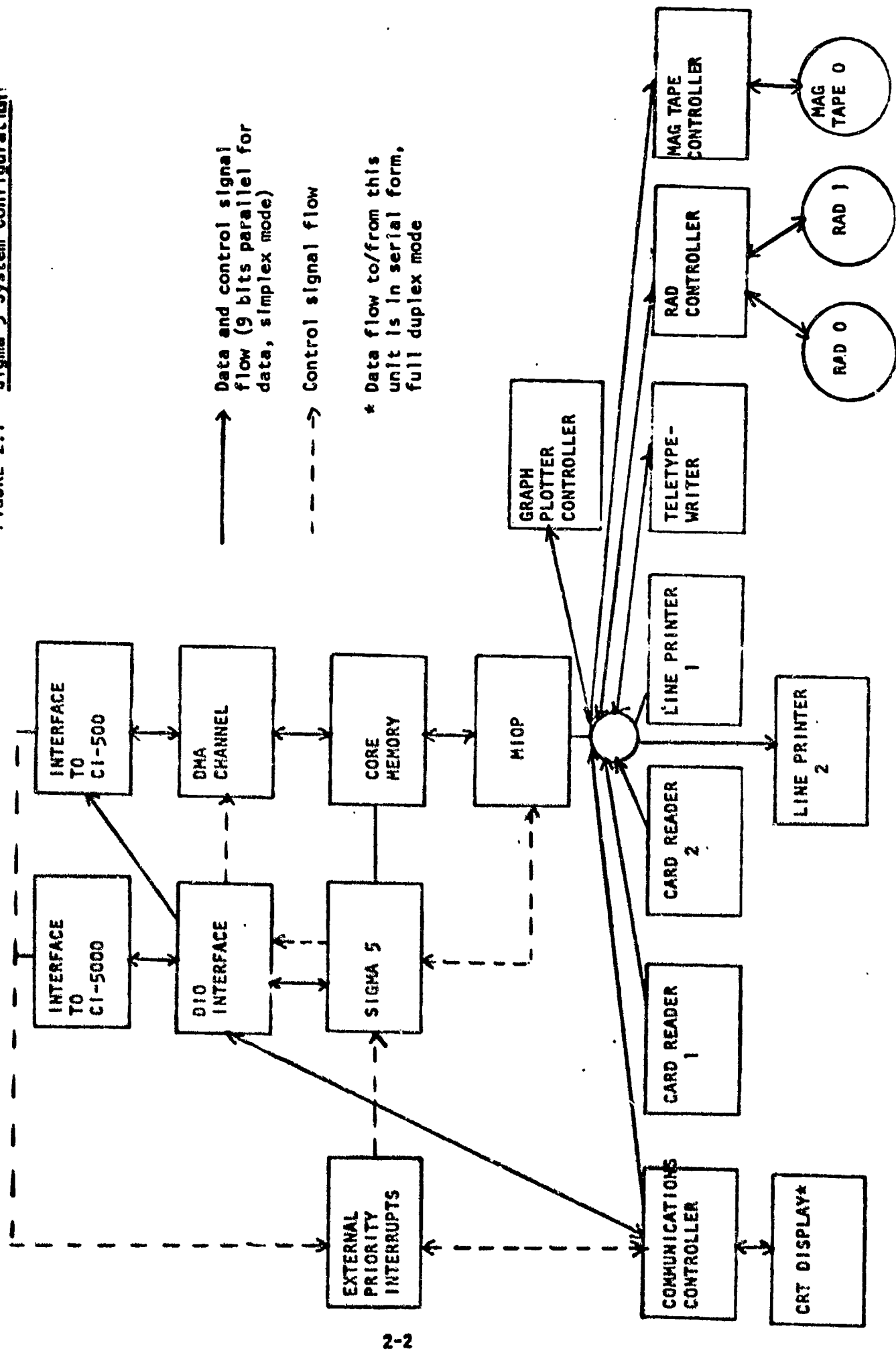


TABLE I

Elements of the Sigma 5 Digital Computer System

Description	Xerox Model Number	Performance Characteristics	MIOP Channel Assignment
Sigma 5 CPU	8202	See Ref. 1	-
MIOP	8273	400 KB/sec.	-
Core Memory	Various	.85 usec cycle	-
Card Reader 1	7122	400 cpm	3
Card Reader 2	7121	200 cpm	4
Line Printer 1	7441	1100 lpm	2
Line Printer 2	7440	600 lpm	5
Keyboard/Printer (Teletype ASR-35)	7020	10 char/sec	1
RAD Controller	7201	188 KB/sec	Fx
RAD File 0	7204	3 Megabytes	F0
RAD File 1	7204	3 Megabytes	F1
Magnetic Tape Controller	7320	60 KB/sec	8x
Magnetic Tape Drive	7322	75 ips	80
Graph Plotter Controller	7534	.01 Inch Inc	7
Communications Controller	7611	1800 Baud	F
CRT Display	4002A*	600, 1800, 2400 or 9600 Baud	-

*Tektronix Inc. Model Number

TABLE II

DIO Address Characters (8 Bits)

Device Description	Address Character (Upper 8 bits)
CI-510 (Interface to CI-5000 Analog)	E
CI-500 Interface and DMA Control Signals	D
Communications Controller* (Printer Output)	3

* The controller is numbered 1 so that all address fields are of the form '301x' where x specifies the required function.

Data transfer operations via the DIO (Direct Input/Output) takes place through CPU registers and occupy the CPU for the duration of the transfer. Each word transferred requires a 16 bit address field in the computer instruction to identify the I/O operation and the device involved. Table II shows the most significant byte of the address field for the hardware currently attached.

The MIOP (Multiplexer Input/Output Processor) transfers data in parallel with CPU operation and occupies the CPU only during the time required to transfer control signals to or from the CPU. The MIOP can handle up to 32 I/O channels each operating simultaneously, provided the overall data transfer rate does not exceed the MIOP maximum rate bandwidth of about 400 kilobytes/second. If the maximum rate is exceeded at any time then a 'data overrun' error condition is signalled for some or all of the peripheral devices involved.

The MIOP currently contains 16 data channels, of which 9 are assigned to peripheral device controllers. Two further groups of 8 channels (Xerox model number 8276) may be added. The Graph Plotter controller shown in Figure 2.1 is intended to drive a 12 inch Calcomp plotter; however, the plotter is not currently attached to the system.

Data transfer to and from the communications controller (Xerox model 7611) uses both the DIO and MIOP data paths. Input is via the MIOP and output is through the DIO on a character-by-character basis with an external interrupt generated following the transmission or receipt of each character. Note that the communications controller does not use the internal I/O interrupt as do all other devices which use the MIOP. Data transmission speed for the single communication channel currently used is set at 9600 Baud. The 7611 has the capability of expansion to 64 channels.

The third I/O path consists of a Direct Memory Access channel which is used exclusively to connect the ADC's and DAC's in the interface to the CI-500 Analog computer to core memory. This channel operates in parallel (asynchronously) with CPU operation but is controlled by signals transmitted via the DIO.

The DMA channel operates at high speed (limited by ADC and DAC conversion rates), transferring data in 32 bit parallel format. Completion of any I/O operation may be signalled to the CPU by use of an external interrupt. It should be noted that the CI-500 ADC's and DAC's are also accessible via the DIO channel. However, it is not possible for the DMA and DIO to access DAC's or ADC's simultaneously.

2.1.3 Magnetic Core Storage

Magnetic core storage consists of 49,152 (48K where K = 1024) words of memory, divided into 3 banks of 16,384 words each. Banks are grouped into 2 memory units, each of which have separate access ports for the CPU, MIOP and DMA. Address interlacing between banks and units is used to achieve overlap in accessing particular memory locations.

Memory locations 0 through 15 are not available since any reference to these addresses in a computer instruction refers to the general register of that number in the block selected by the Register Pointer.

2.2 Analog

The analog section of the hybrid computer facility includes two analog computers - Astrodata Comcor CI-5000 and Astrodata Comcor CI-500 machines. These machines are solid-state, general-purpose analog computers with operating ranges of ± 100 volts. They each include a good range of linear and non-linear analog elements plus powerful logic control sections. A brief discussion of the analog and logic components in both machines is given below.

2.2.1 Analog Components

Operational Amplifier

The operational amplifier may be used as a combination summer/integrator with input gains of 10 or 1 available. Additional gain as an integrator may be obtained by choice of capacitor. Gain due to a capacitor is equal to $1/C$ (since input resistors are 1 megohm) where C is given in micro-farads. Capacitors available are 10 uf, 1 uf, .1 uf, .01 uf and .001 uf. Only odd numbered amplifiers may be used as integrators. Capacitor selection is determined from the logic board as shown in Figure 2.2.

Referring to Figure 2.2, amplifier 003 is connected as an integrator using electronic switching (S^3 to A003) in the real-time mode (RT to .1) with gain of 10 ($1/C$). Reset and hold conditions are determined from the keyboard (RR to R, HH to H). In order to use relay switching remove (S^3 to A003). For the timescale mode connect TS to desired capacitor. In the patching shown in Figure 2.2 RT and TS are determined from the keyboard. Reset and hold may also be determined at the logic panel by patching D14 to R, D15 to H where D14 and D15 are set by clock 1 via the logic board for repetitive operation.

It should be noted that the output of an operational amplifier is the inverse of the input, i.e.

$$E_o = -AE_{i_n}$$

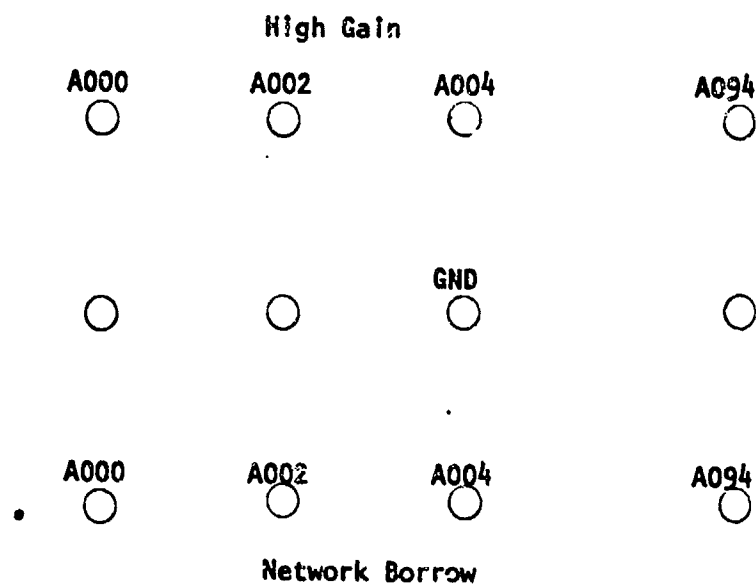


Figure 2.2A Logic Board Comcor 5000 for High Gain and Network Borrow Patching

where E_o is the output voltage, E_{in} is the input voltage and A is the gain across the summer/integrator.

Additional inputs if needed may be obtained by connecting spare resistors located on patchboard to SJ of the operational amplifier.

Each of the integrators has an initial condition (IC) input from which IC's may be set via potentiometers or DAC's. There is a sign change between IC input and the output of the integrator.

Several static test outputs are available via the analog patchboard. With the analog in the static test mode +100 volts is applied via these terminals from which amplifiers and potentiometers may be checked statically.

Even numbered amplifiers (A000 through A094) may be patched as high gain amplifiers by grounding the high gain connection located at the top of the logic patchboard thus eliminating the feedback network. Input networks may be borrowed from the even-numbered amplifiers by patching the network borrow connections at the top of the logic patchboard to ground. See Figure 2.2A for logic patchboard.

Each amplifier on the Comcor 500 may be connected as a high gain amplifier whereas network borrows are available only on the even-numbered amplifiers.

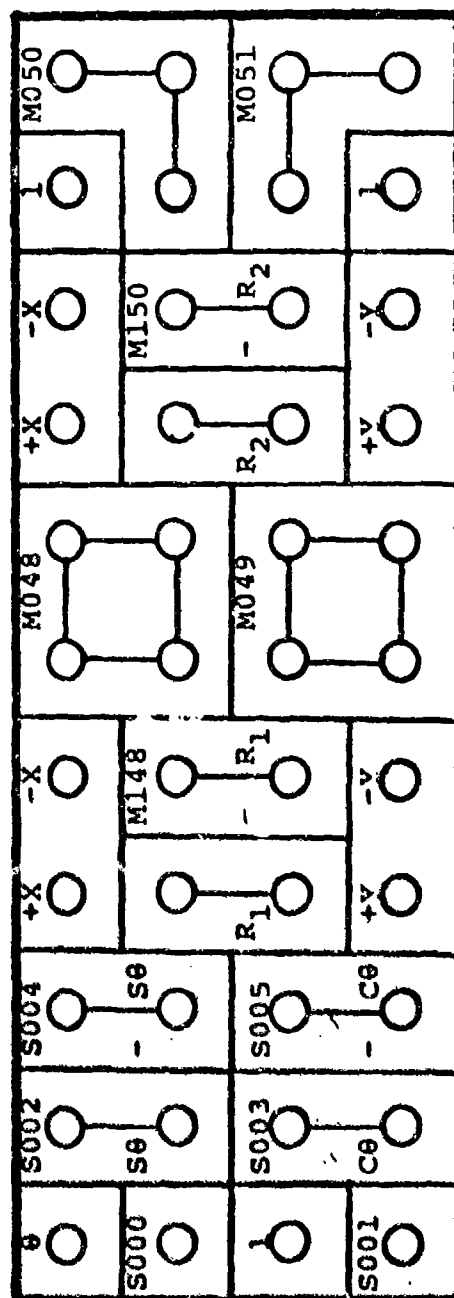
Multiplier

The internal circuit and patchboard layouts (signal and logic) are shown in Figure 2.3. This unit performs division or multiplication according to the selected option.

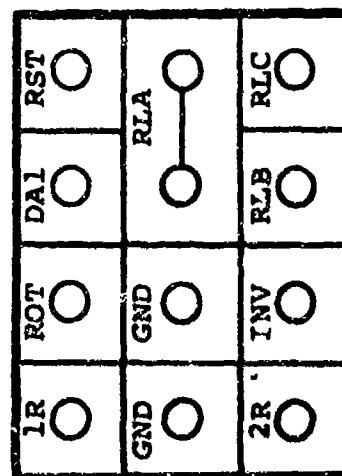
To perform the division operation connect dividend analog input to Z and divisor analog input to Y. For division it is also necessary to connect indicated multiplier patch point on the logic board to ground.

To multiply, plus and minus Y inputs and the +X input must be generated. It should be noted for the multiplier shown minus X is generated internally. However, on some of the multipliers which can easily be recognized, minus X is not generated internally.

Spare amplifiers which may be used as invertors (M100 to M123) are available when multipliers are not being used.

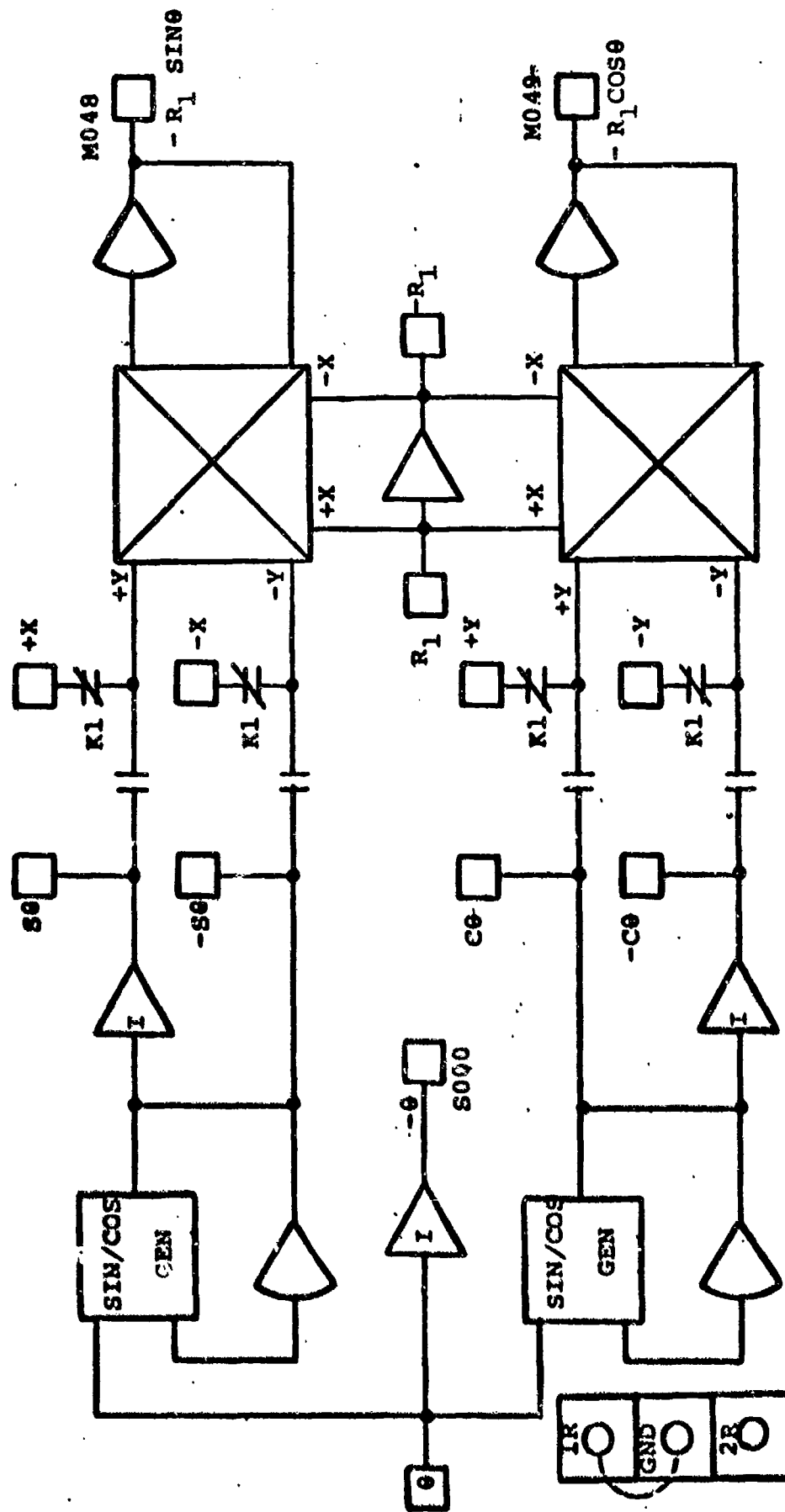


ANALOG PATCHBOARD



LOGIC PATCHBOARD

Figure 2.4 Analog and Logic Patchboard Arrangement for a Resolver



INPUTS: θ AND R_1
 OUTPUTS: $-\theta$, $S\theta$, $-S\theta$, $C\theta$, $-C\theta$, $-R_1 \sin \theta$, AND $-R_1 \cos \theta$

Figure 2.5 Resolver 1R Mode Inputs and Outputs

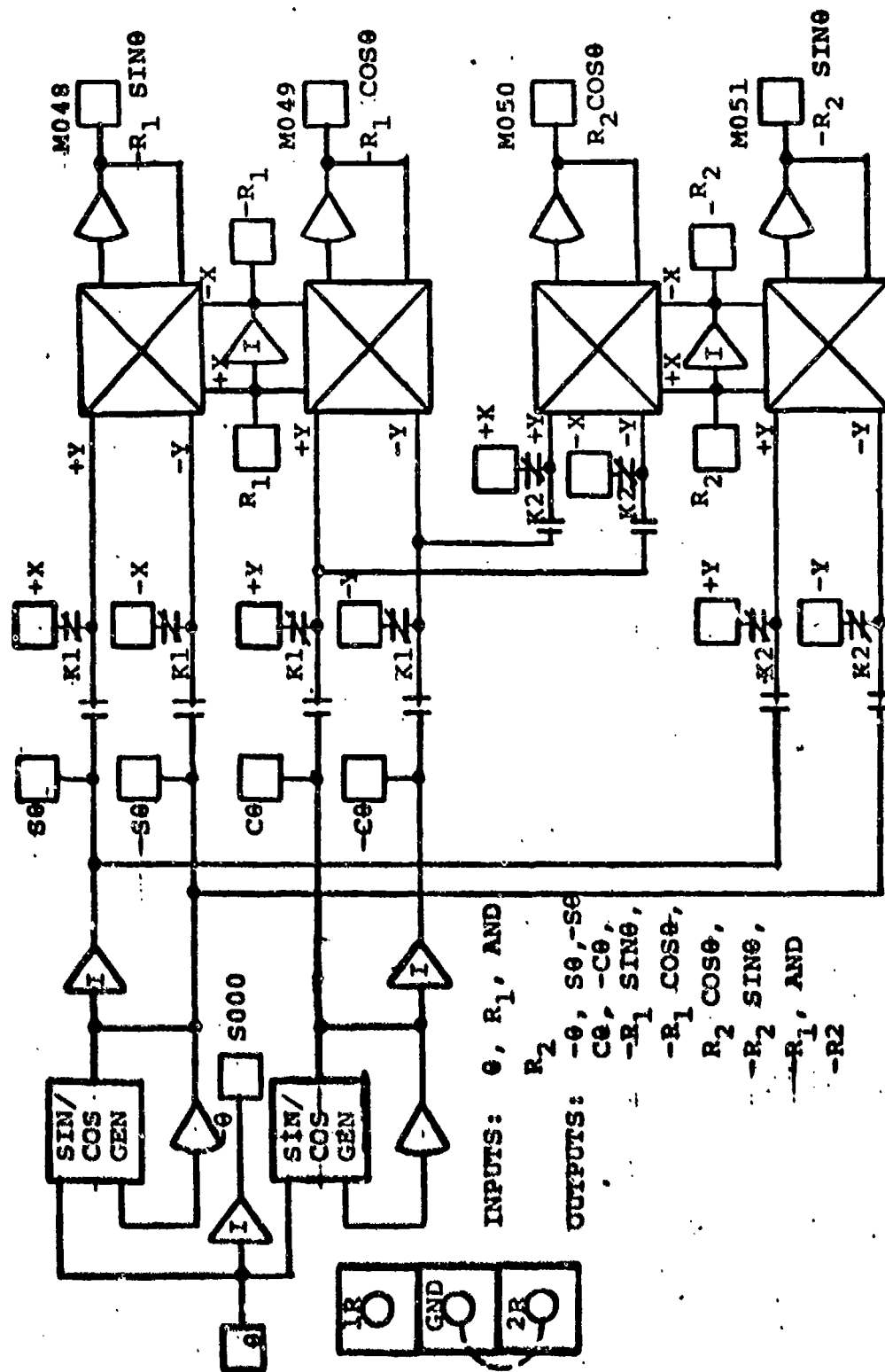
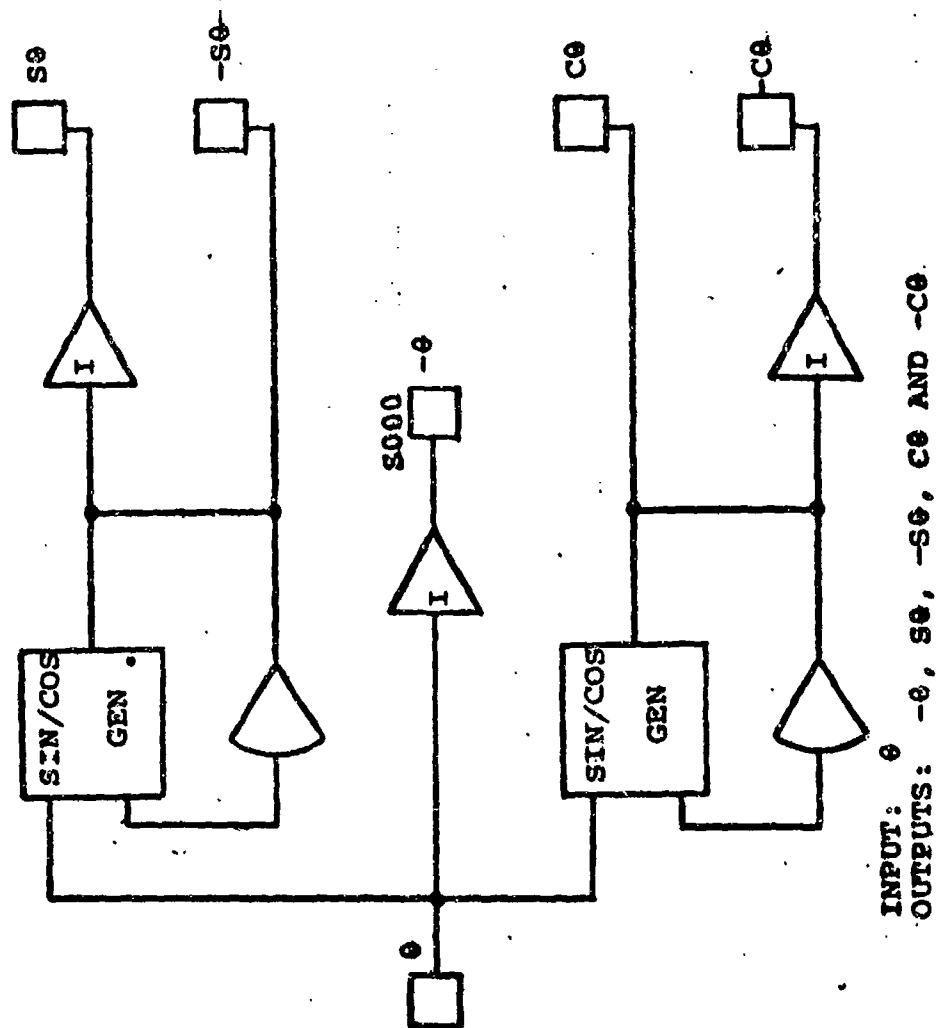


Figure 2.6 Resolver 2R Mode Inputs and Outputs



2-14

Resolver

Six operating options are available when using the resolver. These options, and the necessary patching, are described as follows:

- (1) Sine and cosine mode - connect input which must be scaled 1.8 deg/volt to θ input. $+\sin\theta$, and $+\cos\theta$ are available outputs. No patching is required on the logic patchboard (see Figure 2.4).
- (2) 1R mode - It is necessary to patch 1R on logic board to logic ground. Shown in Figure 2.5 are necessary inputs and available outputs.
- (3) 2R mode - It is necessary to patch 2R on logic board to logical ground. Shown in Figure 2.6 are required inputs and available outputs.
- (4) ROT Mode - Rotation or transformation of axis mode requires that ROT on logic board be patched to logic ground.

Inputs

θ
 X_1 to R_1 input
 Y_1 to R_2 input

Outputs

$$\begin{aligned} Y_2 &= Y_1 \cos\theta - X_1 \sin\theta \\ X_2 &= X_1 \cos\theta + Y_1 \sin\theta \end{aligned}$$

- (5) Rate Mode (continuous resolution) - All of the forward resolution modes (sine, cosine, 1R, 2R, and ROT) can be extended to include continuous resolution. An external integrator is necessary. To use the Rate Mode the following is necessary:

DAI (Digital to Analog switch) and RLA on logic board must be connected and RST must be connected to computer RESET so that the resolver starts in the same quadrant every time.

For analog patching connect θ to I input and S001 to an external integrator through a gain of 2 as shown in Figure 2.7.

- (6) Inverse Mode (rectangular to polar) - It is necessary to patch INV on logic board to logic ground.

Inputs

X_1 into R_1
 Y_1 into R_2

Outputs

$$\begin{aligned} \theta &= (\tan^{-1} X_1/Y_1) \quad \text{M048} \\ -R &= (\sqrt{X^2+Y^2}) \quad \text{M049} \end{aligned}$$

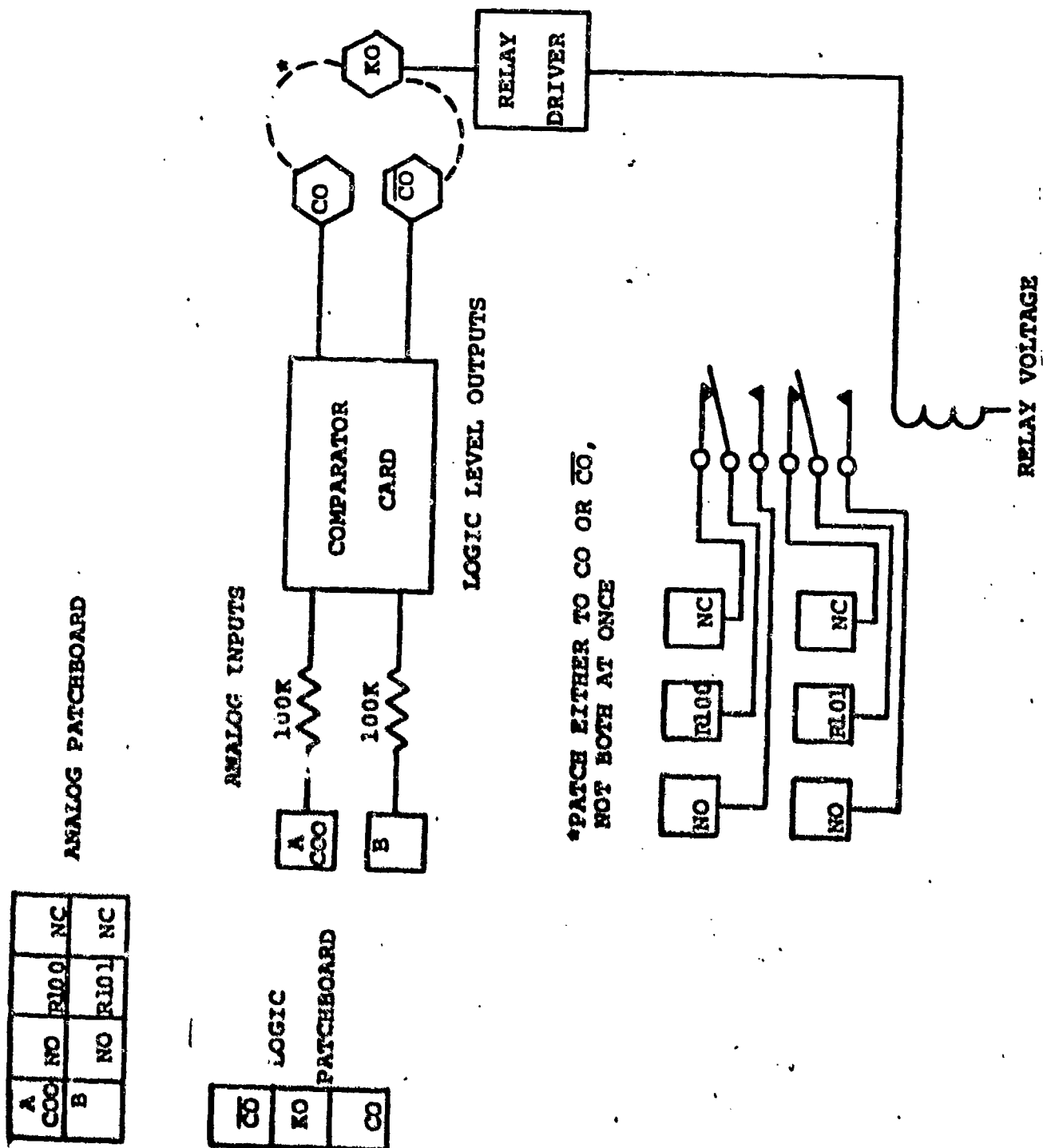


Figure 2.8 Comparator Patching

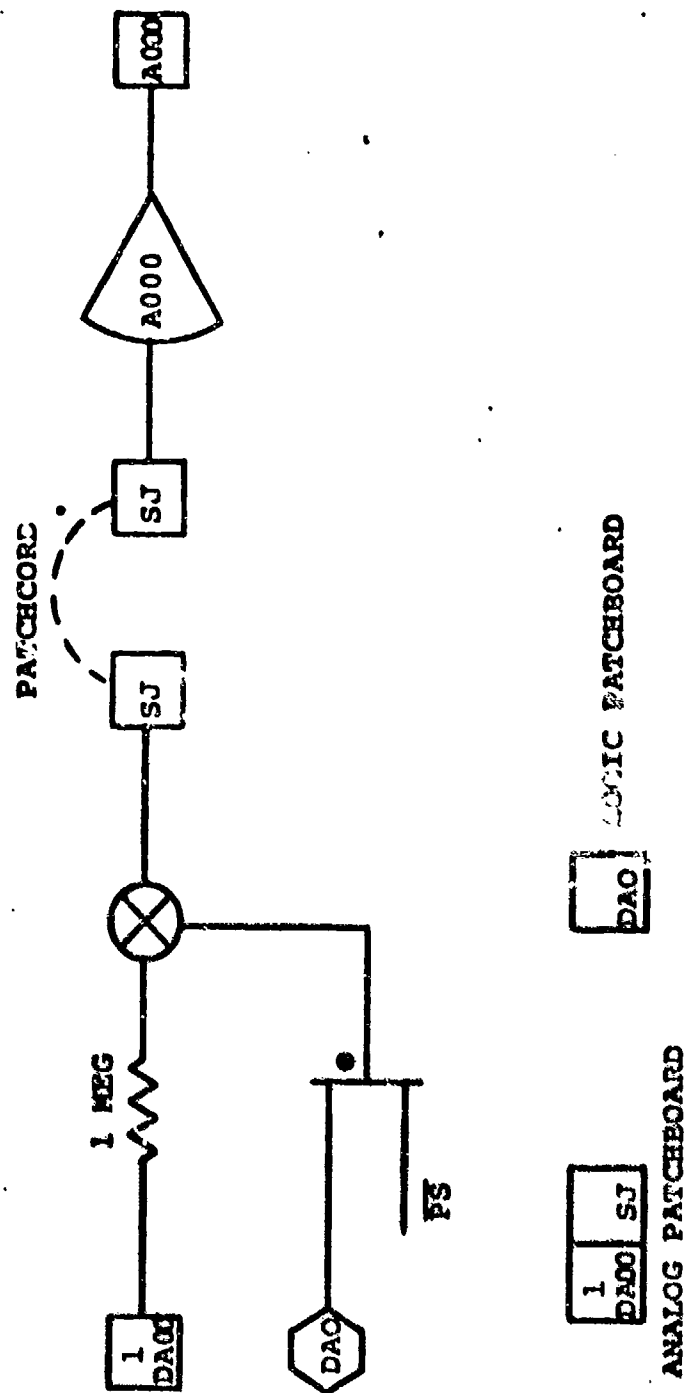


Figure 2-9 DA Patching

Comparator

The comparator compares two input voltages and produces a logic 'true' or 'false' depending on whether the first input is greater or less than the second (or reference) input. The logical 'not' of this output is also available.

Analog and logic patchboard for a comparator is shown in Figure 2.8. The comparator output usually drives a relay, as shown in Figure 2.8.

The voltage to be compared is applied to C00 while the negative of the reference level is applied to $\bar{d}$. When the compare conditions are satisfied C0 ($\overline{C0}$) on the logic board change state. Either C0 or $\overline{C0}$ may be used to drive K0 which switches the relay on the analog patchboard. An external signal other than C0 or $\overline{C0}$ may be used to drive K0 if desired. For the relay outputs, the NC (or 'normally closed') position is that for a no logic signal input.

DA Switch

The DA switch provides a means of closing or opening an analog signal path with a logical variable. The analog and logic patchboard connections for a typical DA switch is shown in Figure 2.9. These are available on the CI-5000 only.

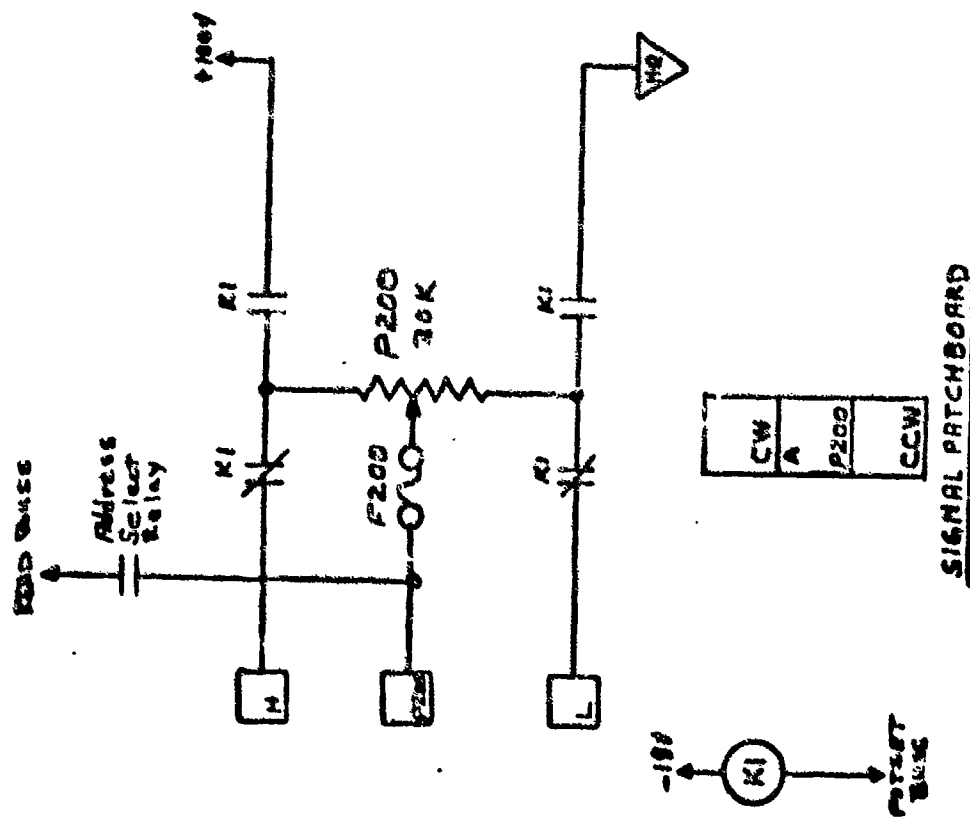
The signal to be passed is connected to DA00 (analog). The output of the DA switch is connected from SJ of DA switch to SJ of an amplifier. Switching is controlled totally by DAO on the logic patchboard. Logic ground ('false') closes the switch while -6V opens the switch.

Potentiometer

Shown in Figure 2.10 are the schematics for the manual and servo set potentiometers. The computer should be in the Potset mode when setting each of the potentiometers. The servo potentiometer is set via the keyboard servo button and DVM (digital voltmeter).

- (1) Place computer in POTSET mode.
- (2) Select pot to be set via keyboard. Pot number is displayed.
- (3) Set DVM to value desired by pressing + on keyboard followed by 4 digit potsetting. Value displayed on DVM.
- (4) Press servo button and pot will set.

MANUAL POTENTIOMETER



SERVO POTENTIOMETER

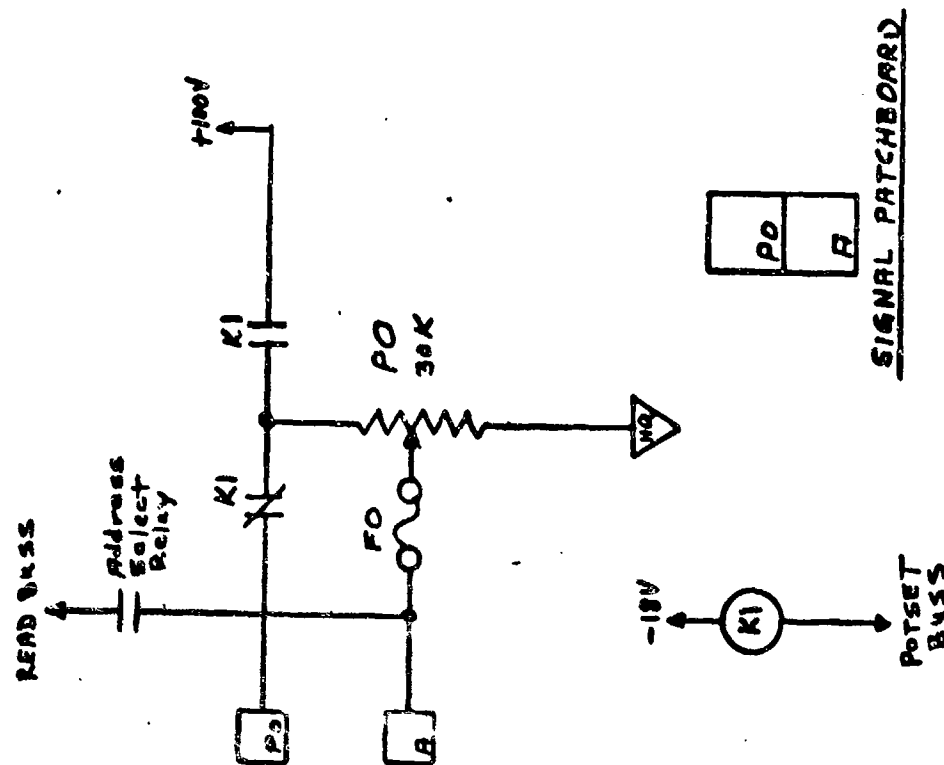


Figure 2.10 Potentiometer Patching, Servo and Manual

Track($\bar{t}$)-No Track($\bar{T}$) Amplifier .

T- $\bar{T}$ amplifiers operate on the Comcor CI-5000 as gain switches (switching input resistors). The switch is a DA switch controlled from the logic board; i.e. ground (false) DS00 on logic board results in a gain of 10 ($\bar{T}$ Input). Amplifiers 100-119 are TT amplifiers while DS00-DS019 on logic board are the logic controls.

Limiter

The limiter, a common non-linear element in a system, is available directly on the analog patchboard. Connect G of limiter to G of the amplifier to be limited and L00 of limiter to the output of the amplifier. With the computer in the POTSET mode set limits via the limit switches and pots located directly above the analog patchboard. With the switch left set lower limit with pot on left. With the switch right set upper limit with pot on right. Monitor output of amplifier when limits are being set.

Function Switches (Analog)

Function switches located to the left of the analog patchboard are double-pole triple-throw switches. Three different signals can be patched into the L, C, and R positions and the output at the switch arm can be selected by setting the function switch to the desired position. On some switches A and B switch arm positions are available permitting simultaneous control of two sets of function with one switch.

Trunks

Trunk lines are available between the Comcor 5000 and the Comcor 500 which makes it possible to transmit signals between the two machines. In addition, it is possible to slave the CI-500 mode to the CI-5000 mode. Trunk lines are also available between the Comcor 5000 and the Angular Motion and Target Simulators.

In some cases it is desirable to use logic signals on the CI-5000 analog patchboard (or vice versa). For this, trunk lines (IP00-IP15) are available between the logic and analog patchboards of the CI-5000.

2.2.2 Logic Components

Single input and 4-input NAND gates, delay flip-flops, RS flip-flops and digital switches are available for generating control logic.

Delay time for the delay flip-flop is determined by a rotary switch to the right of the analog patchboard. Delay times available are 100 usec,

1 MS, 10 MS, 100 MS, 1 second and 10 seconds. Each switch also has a Vernier scale which varies the delay between the given switch setting and the next, thus covering the range of 0 to 10 seconds continuously. To activate the delay flip-flop ground ENB and switch clock input from -6V to ground as shown in Figure 2.11.

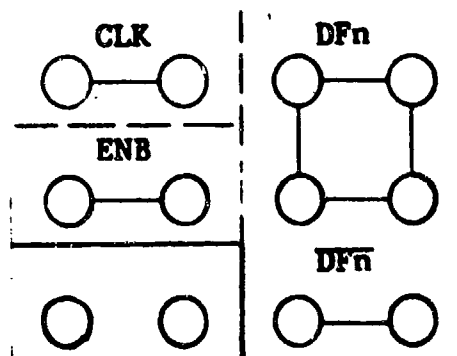


Figure 2.11 Delay Flip-Flop Logic Patchboard

The general purpose flip-flop is shown in Figure 2.12 R0 and S0 inputs provide direct flip-flop control. These inputs override any other flip-flop input and provide asynchronous control. A logical 0 (gnd) applied to either R0 or S0 causes the $\overline{F0}$ or F0 output respectively to be a logical 1 (-6V). If a logical 0 is applied to both R0 and S0, both F0 and $\overline{F0}$ will be logical 1.

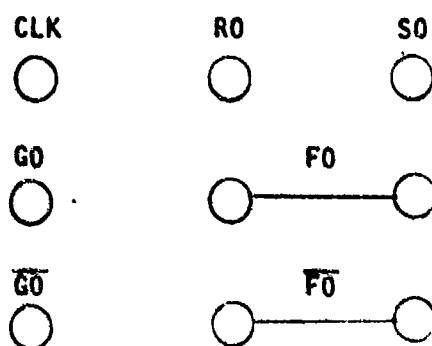


Figure 2.12 General Purpose Flip-Flop Logic Patchboard

The CLK terminal is used for a clocked or asynchronous signal. The CLK driver is connected internally to the G0, $\overline{G0}$ inputs. The transition from a logical 1 to 0 causes the flip-flop to be placed in a state given by the following J-K flip-flop truth table, where J^n, K^n are the present states of G, $\overline{G}$. Q^n is the present output state and Q^{n+1} is the output state after the next clock pulse.

J^N	K^N	Q^{N+1}
0	0	Q^N
0	1	0
1	0	1
1	1	$\overline{Q^N}$

Figure 2.13 RS Flip-Flop in J-K Mode

Digital switches are different between the two analog machines. CI-5000 switches are double-pole, double-throw for switching logic signals. CI-500 switches have single outputs which are 'true', 'false' and pulse 'true' for up, center and momentary down positions.

Control, sense and interrupt lines are available between the digital and analog computers. Control lines are logic (or binary) outputs from digital to analog and sense lines are logic inputs to the digital from the analog computers. Interrupt lines are logic inputs to the digital which cause a program interruption when the digital computer is programmed to accept them. Interrupt inputs should be clocked pulses; control and sense lines are logic levels.

DAC and ADC communication between digital and analog computers is available. Patching for input/output is placed on the analog patchboard, but S/H control for ADC's and DAC update control is performed by patching special control lines on the logic board. Each special control line controls 2 ADC's or 2 DAC's (4 each on the CI-500) and is operated by a pulse on the line. All DAC's are of the multiplying type whereby an analog input signal at the DAC output causes the digital and analog signals to be multiplied together. If required, the analog signal may be constant at ± 100 volts to give a unity multiplier.

External Integrator control and/or repetitive runs are available via the clock 1 input located on the logic board. Clock 1 is driven by clock frequencies shown on the logic patchboard. Counters, which count number of clock pulses while the computer is in the reset, hold, and compute modes are located on the console in respective order. For Clock 1 to be active the computer must be in the compute mode with the clock button set. The associated run, hold control signals (R1, H1) are generated via the logic patchboard and may be used to control integrator states. See Figure 2.14 for setup.

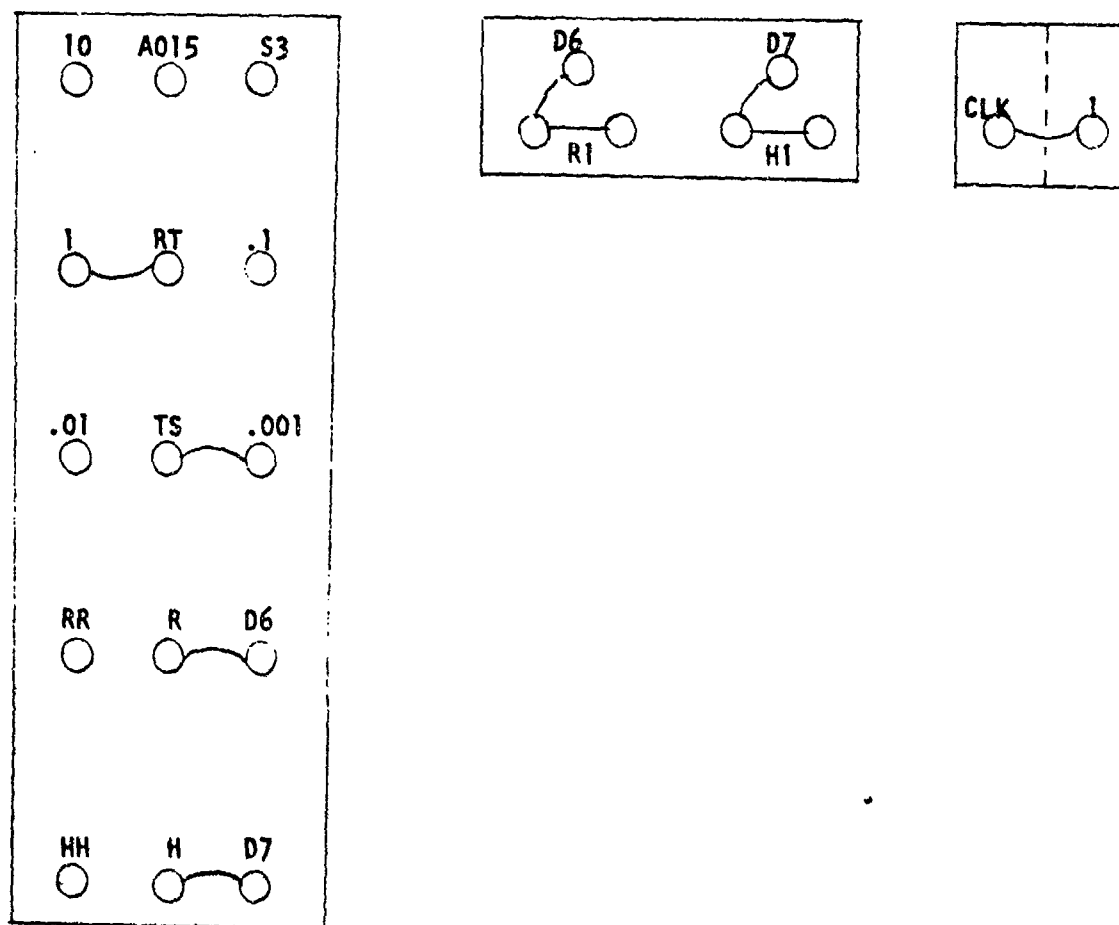


Figure 2.14 Logic Board for External Clock Control

Comcor 500

Analog and logic patchboards of the CI-500 computer contain connections which are not indicated on some patchboards. These connection points are indicated as follows. It should be noted that CI-500 patchboard component numbering is in octal format:

DAC's and ADC's appear on the analog board as shown in Figure 2.15.

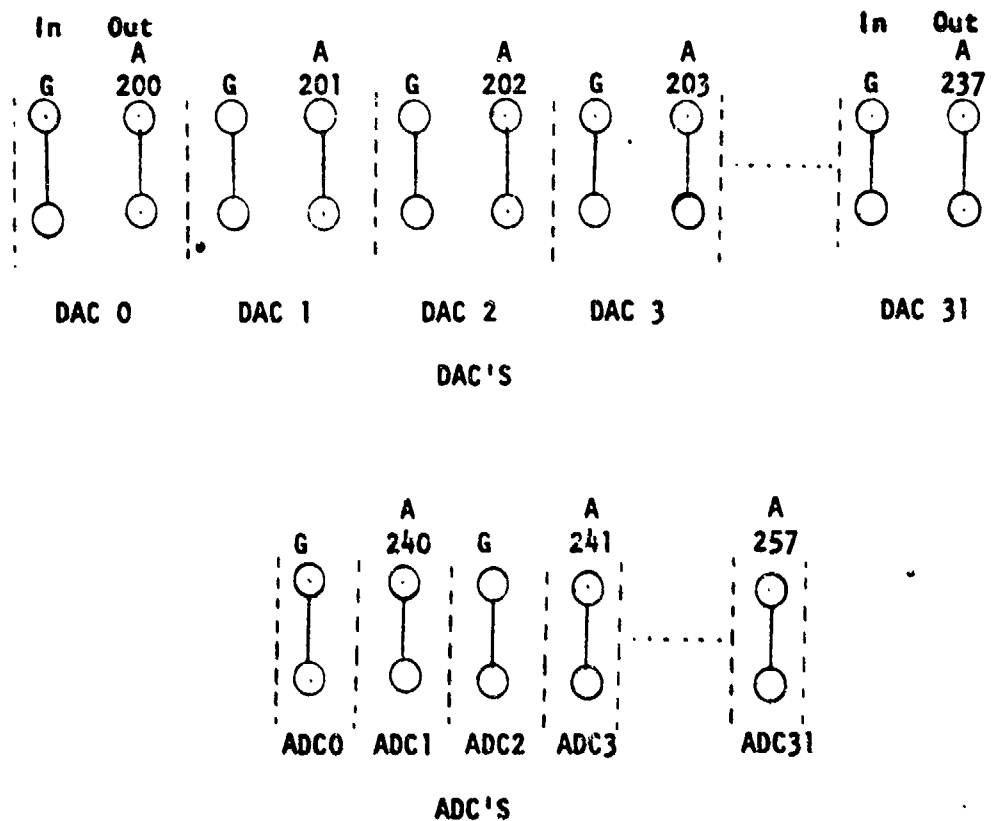


Figure 2.15 CI-500 DAC and ADC Patch Connections

On the CI-500 logic board, the control inputs for ADC Sample/Hold and DAC update are indicated in Table III. There are no special control lines for these functions. Each control input operates on a group of four corresponding ADC's or DAC's.

Patching for the two CI-500 interface clocks is given in Table IV. For these clocks, input control points are available at the analog console, as are status monitoring points. Clock 0 on the CI-5000 interface has only pulse out available on the logic patchboard, plus an input for an external clocking rate to replace the internal rate of 1 microsecond.

Logic Address	S/H Number	Logic Address	DAC Xfer Number
T 00	0	T 010	0
T 01	1	T 011	1
T 02	2	T 012	2
T 03	3	T 013	3
T 04	4	T 014	4
T 05	5	T 015	5
T 06	6	T 016	6
T 07	7	T 017	7

TABLE III Logic Board Addresses and ADC/DAC Control

Logic Address	Clock 1 Function	Logic Address	Clock 2 Function
T 020	RESET IN	*	RESET IN
T 021	STOP IN	*	STOP IN
T 022	RUN IN	*	RUN IN
T 023	RESET OUT	*	RESET OUT
T 024	STOP OUT	*	STOP OUT
T 025	RUN OUT	*	RUN OUT
T 026	PULSE OUT	*	PULSE OUT

TABLE IV CI-500 Clock Patching

*These patchpoints are unmarked, but are located in the white painted area occupying the six upper leftmost pairs of holes (i.e. in 2 columns and three rows).

Multipliers on the Comcor 500 require external amplifiers. The input amplifier is a sign change only while the output amplifier is high gain. The high gain amplifier is generated by patching HG on the logic patch-board for the amplifier used.

To obtain integrator mode control on the Comcor 500 the logic board must be patched as shown in Figure 2.15A.

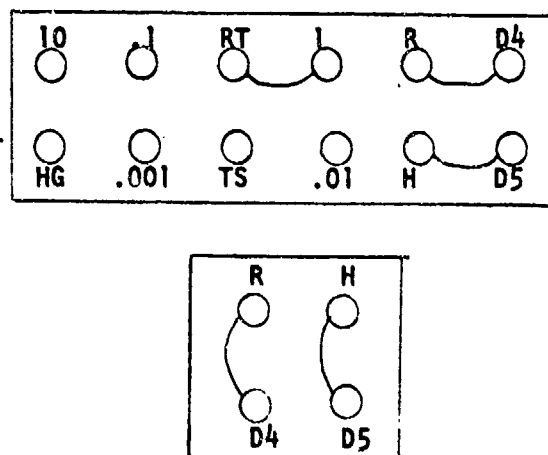


Figure 2.15A Integrator Mode Control Logic

Other operations on the Comcor 500 analog and logic patchboards are similar to the Comcor 5000 operation.

For non-linear functions, diode function generators are available on the Comcor 5000 (servo-set - 10 segment) and the Comcor 500 (hand-set - 10 segment). For a detailed description of operation the reader is referred to the CI-225A Manual.

2.2.3 Summary

Shown below are tables summarizing components of the Comcor 500 and Comcor 500. Also given is trunkline identification between the Comcor 5000 and Comcor 500.

ANALOG

	<u>5000</u>	<u>500</u>
Amplifiers	120	40 (0-47)
Integrators	48	20
Multipliers	48	24
Resolvers	8	4
Comparators	20	8
Relays	16	8
DA Switch	16	0
Potentiometer		
Handset	24	24
Servo-set	100	48
T-T Amplifier	20	0
Limiters	16	0
Function Switch	20	8
ADC's	32	32
DAC's	32	32
Trunks		
500--5000 (31)		
500--Table Room (10) (T140-T141)		
5000--Table Room (27) (T066-T093)		

LOGIC

	<u>5000</u>	<u>500</u>
Nand Gates		
1-Input	24	18
2-Input	24	18
4-Input	24	18
Flip-Flops		
Delay	8	3
R-S	32	16
*Interrupts	16	16
Control Lines	16	16
Sense Lines	16	16
Digital Switch	10	8

*The XDS Sigma 5 has only 16 Interrupts each of which may be triggered from either the Comcor 5000 or Comcor 500.

5000

T 032
T 033
T 034
T 035
T 036
T 037
T 038
T 039
T 040
T 041
T 042
T 043
T 044
T 045
T 046
T 047
T 048
T 049
T 050
T 051
T 052
T 053
T 054
T 055
T 056
T 057
T 058
T 059
T 060
T 061
T 062
T 063

500

T 040
T 041
T 042
T 043
T 044
T 045
T 046
T 047
T 050
T 051
T 052
T 053
T 054
T 055
T 056
T 057
T 060
T 061
T 062
T 063
T 064
T 065
T 066
T 067
T 070
T 071
T 072
T 073
T 074
T 075
T 076
T 077

TABLE V Trunkline Identification between
Comcor 5000 and Comcor 500

The following is a list of Comcor 5000 power supply addresses and values:

P500	+100	P508	+20
P501	-100	P509	-20
P502	+150	P510	-20
P503	-150	P511	+18
P504	+20	P512	-18
P505	-20	P513	-6
P506	+150	P519	RDAC
P507	-150		

The following is a list of Comcor 500 power supply addresses and values:

P500	+100	P516	-6
P501	-100	P517	-18
P502	+150	P520	+18
P503	-150	P521	-6
P504	+20	P522	+26
P505	-20	P523	-24
P506	+20	P524	-24
P507	-20	P525	RDAC
P510	+150	P526	-18
P511	-150	P527	+18
P512	+20	P530	-6
P513	-20		
P514	+18		
P515	-18		

2.3 Analog/Digital Interfaces

As illustrated in Figure 2.1, the digital computer is interfaced to Astrodata Concor CI-500 and CI-5000 analog computers. Data is transferred through the interfaces via DAC's, ADC's and control and sense (i.e. discrete or binary) lines. Each interface has one or more real-time clocks which are accessible at both sides of the interface to permit time synchronization of analog and digital elements. Inputs to the Sigma 5 external interrupt system are also available at both analog consoles. Table VI lists the data transfer elements in each interface. Note that the special control lines in the CI-5000 interface are for use in controlling simultaneous DAC updating or ADC sampling.

2.3.1 Signal Transfer

Both interfaces are connected to the Sigma 5 DIO channel and all data and logic signals may be transmitted by appropriate 'Write Direct' or 'Read Direct' commands (see Reference 1). However, DAC's and ADC's in the CI-500 interface may also transmit data, under external interrupt control, to and from memory directly by means of the high speed DMA channel. Continuous operation in the DMA mode may be obtained through use of the 8 pre-set registers of the DMA in combination with the 'chain' option.

Discrete signals may be transmitted singly or in groups. Reference 2 contains details of the various commands and interface I/O modes.

2.3.2 DAC and ADC

Digital-analog and analog-digital conversion uses 15 bit wide digital data which represents an analog range of ± 100 volts. Both DAC's and ADC's have optional fixed and floating point modes; in floating point mode DAC inputs and ADC outputs are in standard floating point format in the range ± 1.0 and in this case the data occupies an extra 8 bits to allow space for the floating point exponent. In fixed point mode the digital value from DAC or to ADC occupies a 32 bit word with the most-significant 15 bits representing a binary fraction in the range -1 to $+(1-2^{-14})$ with negative numbers in 2-complement form.

DAC's are of the multiplying type wherein the analog output is formed by multiplying the converted digital signal by an analog signal which is input at the analog patchboard. A unity analog multiplier may be obtained by patching a ± 100 volt signal.

For ADC's which terminate on the CI-5000 (addresses 0-31), a sign inversion occurs on data transfer. Positive voltages input to the ADC are read as negative data in the digital computer. On the CI-500 the DAC's (addresses 32-63) invert so that a positive value output by the digital computer produces a negative voltage if the analog multiplier is positive. ADC conversion on the CI-500 (addresses 32-63) and DAC conversion on the CI-5000 (addresses 0-31) do not involve a sign inversion.

Element	CI-5000		CI-500	
	Number	Address*	Number	Address*
DAC	32	0-31	32	32-63
ADC	32	0-31	32	32-63
Control Line	32	Group 1 0-31	16	Group 3 0-15
Sense Line	32	Group 1 0-31	16	Group 3 0-15
Special DAC Control Line	16	0-15	0	-
Special ADC Control Line	16	0-15	0	-
External Interrupt**	15	1-15	15	1-15
Real-Time Clock	1	0	2	1 and 2

TABLE VI Analog/Digital Interface Elements

* Addresses are given in decimal form

** Interrupts at both consoles are identical

2.3.3 Control and Sense Lines

Control lines may be set, reset or pulsed singly and may be set or reset in a group of 32 or 16 (for CI-5000 and CI-500 respectively). The status of the 32 group 1 (CI-5000) control lines may also be read as a group. Sense lines may be read singly or in the same groups of 32 and 16.

Special control lines are available (on the CI-5000 only) for control of DAC transfer and ADC sample/hold. If the control inputs to DAC and ADC are patched then DAC conversions are only updated when a pulse is sent to the control and ADC sample or hold mode is dependent on the control input being reset or set. If the control inputs have nothing patched to them, then DAC updating and ADC S/H occurs whenever an I/O operation is requested by the digital machine. On the CI-5000 there are 16 DAC and ADC control inputs, each operating on a pair of DAC's or ADC's. The CI-500 has 8 DAC and 8 ADC control inputs, so that each input controls 4 DAC's or 4 ADC's; the CI-500 DAC and ADC control inputs are not accessible directly through the A/D interface, but only through logic signals patched at the logic patchboard.

2.3.4 Interrupts

The Sigma 5 digital computer contains 16 external interrupts, numbered 0 to 15 external group 2. Interrupt 0 is permanently connected to the DVM's (Digital Voltmeters) of the two analog computers and is not available at the logic patchboards. External interrupts 1 to 15 are available at both analog consoles and are paralleled into the Sigma 5 but interrupt 1 is wired to the DMA channel and should not be used by a foreground program. Interrupt response processing in the digital machine has no method of identifying the source of an external interrupt, and thus care should be exercised to avoid confusion in the use of the external interrupts. In addition, the graphics display unit interface (7611) is wired into interrupts 6 and 7 and these interrupts should not be used at the analog consoles if the graphics terminal is in use by any (foreground or background) program. Furthermore, the HRBM monitor Control Task uses external interrupt 15 and thus, this interrupt should not be used.

2.3.5 Clocks (Interval Timers)

The CI-5000 interface contains an interval timer (Clock 0) with the following characteristics:

- (a) Selectable internal and external modes
- (b) Single or repeating countdown (internal mode)

- (c) Internal count rate of 1 megahertz (1 microsecond).
- (d) Maximum internal cycle count 65,537.
- (e) External mode input at CI-5000 logic patchboard.
- (f) Control from digital computer DIO channel only.

The CI-500 interface contains two interval timers (Clocks 1 and 2) with the following characteristics:

- (a) Internal mode only.
- (b) Repeating countdown only.
- (c) Count rate 100 kilohertz (10 microseconds).
- (d) Maximum cycle count 65,537.
- (e) Control from digital computer DIO channel or CI-500 logic patchboard.

Control operations consist of resetting, setting, initializing, running, stopping and reading the clocks. In addition, the status of Clocks 1 and 2 may be determined at the CI-500 logic patchboard.

2.4 Angular Motion Simulator

2.4.1 Hardware Characteristics and Axes Systems

The angular motion simulator consists of a hydraulic, servo-driven platform mounting with three gimbaled rotational degrees of freedom about mutually perpendicular axes and associated control and hydraulic systems. Dynamic inputs to the control system, originating from a real-time dynamic simulation, enable hardware components mounted on the platform to experience real in-service rotational rates and positions. Table VII shows nominal performance specifications of the motion simulator.

Gimbal order is roll, yaw, pitch with roll the inner axis. Thus, a convenient Euler angle representation for defining orientation of body axes of the hardware components relative to an inertial reference is the non-orthogonal ordered set ϕ, ψ, θ where the angles are conventionally defined.

Inputs to the simulator consist of three voltages which determine the angular positions of the three axes. Pickoff output voltages are available which yield the actual position of each axis at any instant.

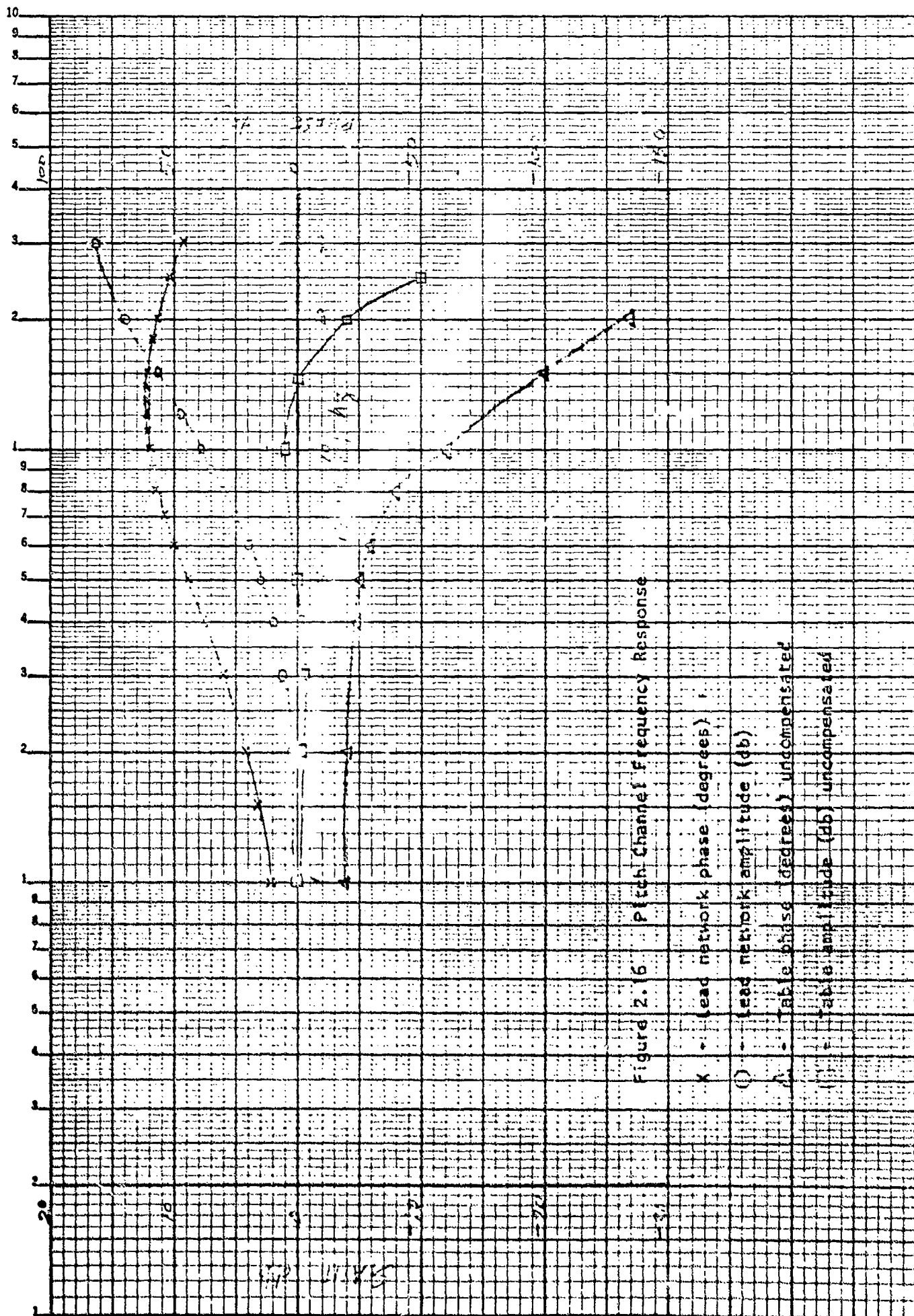
TABLE VII
Angular Motion Simulator
Performance Specifications

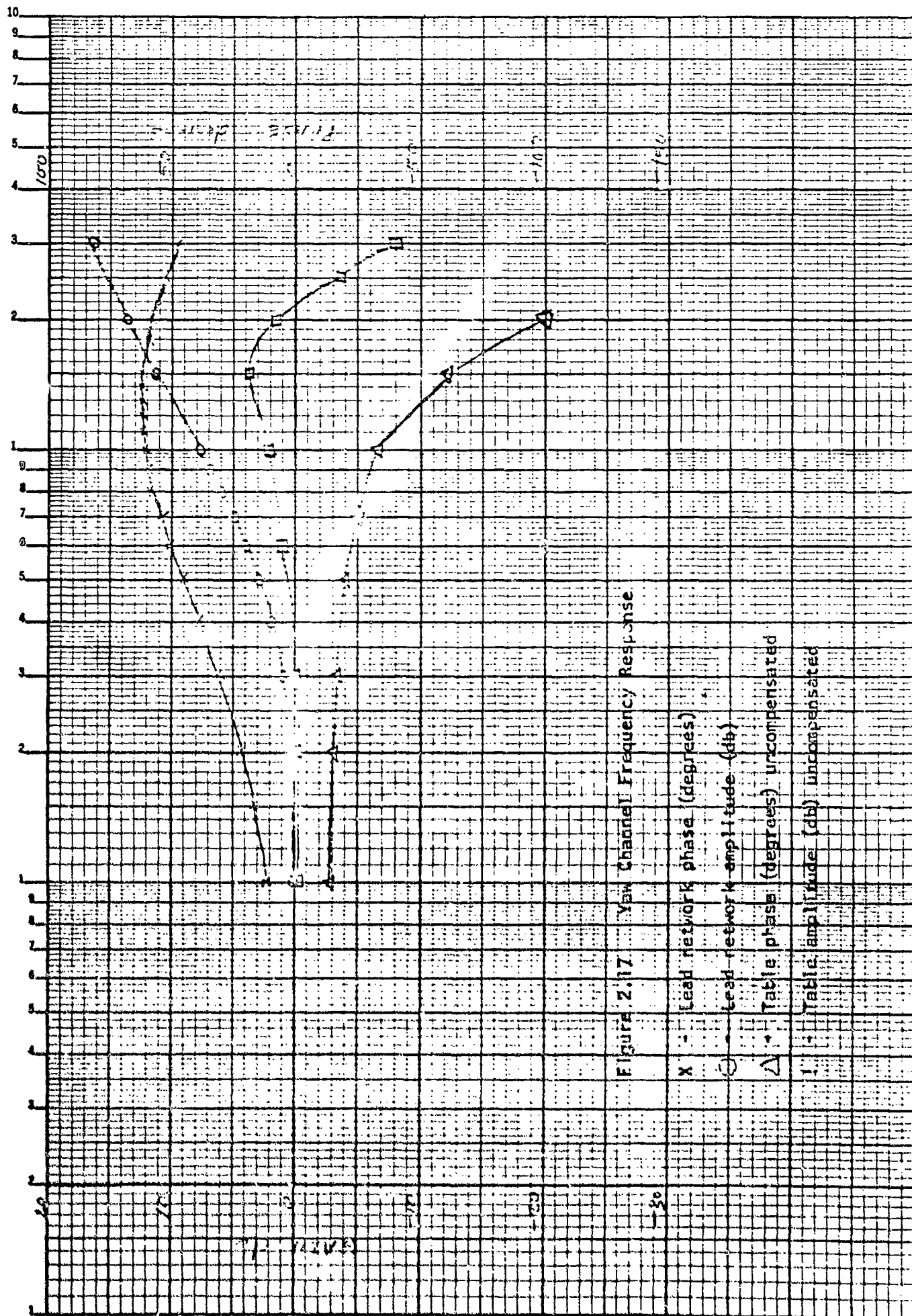
with 5.0 lb-in-sec² load
 (1930 lb-in²)

	Roll (red)	Yaw (yellow)	Pitch (blue)
Max. Acceleration, exceeds	50,000°/sec ²	25,000°/sec ²	12,000°/sec ²
Max. Velocity, exceeds	700°/sec	400°/sec	200°/sec
Min. Velocity, less than	0.0004°/sec	0.0004°/sec	0.0004°/sec
Displacement	±120°	±120°	±120°
Position Accuracy (Maximum)	±0.053°	±0.053°	±0.053°
Repeatability, less than	±0.005°	±0.005°	±0.005°
Readout accuracy	±0.12°	±0.12°	±0.12°
Command Signal Scaling (typical)	1.80°/volt	1.80°/volt	1.80°/volt
Frequency Response	27.5 cps	17.6 cps	12.8 cps

Pl. 1

K-E SEMI-LOGARITHMIC 46 5493
3 CYCLES X 70 DIVISIONS MARK IN U.S.A.
KEUFFEL & ESSER CO.





Nominal scalings between angular position and voltage are shown in Table VII. These may easily be changed to produce increased sensitivity at the cost of reduced displacement ranges by varying the amplifier input resistance. Maximum input voltages are ± 100 volts.

To define the sign conventions of the gimbal angles ϕ , ψ , θ (roll, yaw, pitch) consider the set of Cartesian axes X, Y, Z with origin at the center of gimbaled rotation. The X axis is horizontal, positive defined geographically as pointing to the south wall of the laboratory, the Y axis is positive vertically upwards and the Z axis to the right when facing in the +X direction. Positive voltages into the table control system produce gimbal angles corresponding to an ordered set of Euler angles in the above order (ϕ , ψ , θ) which locate a set of axes, fixed in the inner gimbal, relative to X, Y, Z.

A common set of axes used in aerospace simulations is that obtained by rotating the X, Y, Z set $\pi/2$ about X (i.e. Z vertically downwards). In this case the ψ Euler angle input voltage to the simulator requires to be inverted in sign for correct response of the yaw gimbal. θ and ϕ are unchanged.

2.4.2 Phase Compensation

Figures 2.16 and 2.17 illustrate the frequency response of the pitch and yaw channels of the motion simulator. Occasionally the phase lag exhibited is not acceptable and the application of a lead filter to the input signals is necessary. A network with a frequency domain response given as:

$$f(s) = \left\{ \frac{\tau_1 s + 1}{\tau_2 s + 1} \right\}^2$$

where $\tau_1 = .0222$ and $\tau_2 = .0071$, has been found to give adequate compensation to signals which contain frequencies to about 4-5 Hz. The response of the above filter is included in Figures 2.16 and 2.17 and an analog computer patching diagram which implements the filter is shown in Figure 2.18. Since this filter is essentially a differentiating network, input signals should not be derived from DAC's without some intervening smoothing to give a continuous input signal with continuous first derivative.

2.4.3 Simulation Hardware Protection

When running hardware-in-the-loop simulations which include gyro-mounted seekers or similar units on the Angular Motion Simulator, it is usually wise to protect the hardware components by an automatic run-abort mechanism based on gyro precision angles approaching a maximum value. This is achieved by feeding the gyro angles, which are usually available

$$\frac{E_o}{E_{IN}} = \left(\frac{\tau_1 s + 1}{\tau_2 s + 1} \right)^2$$

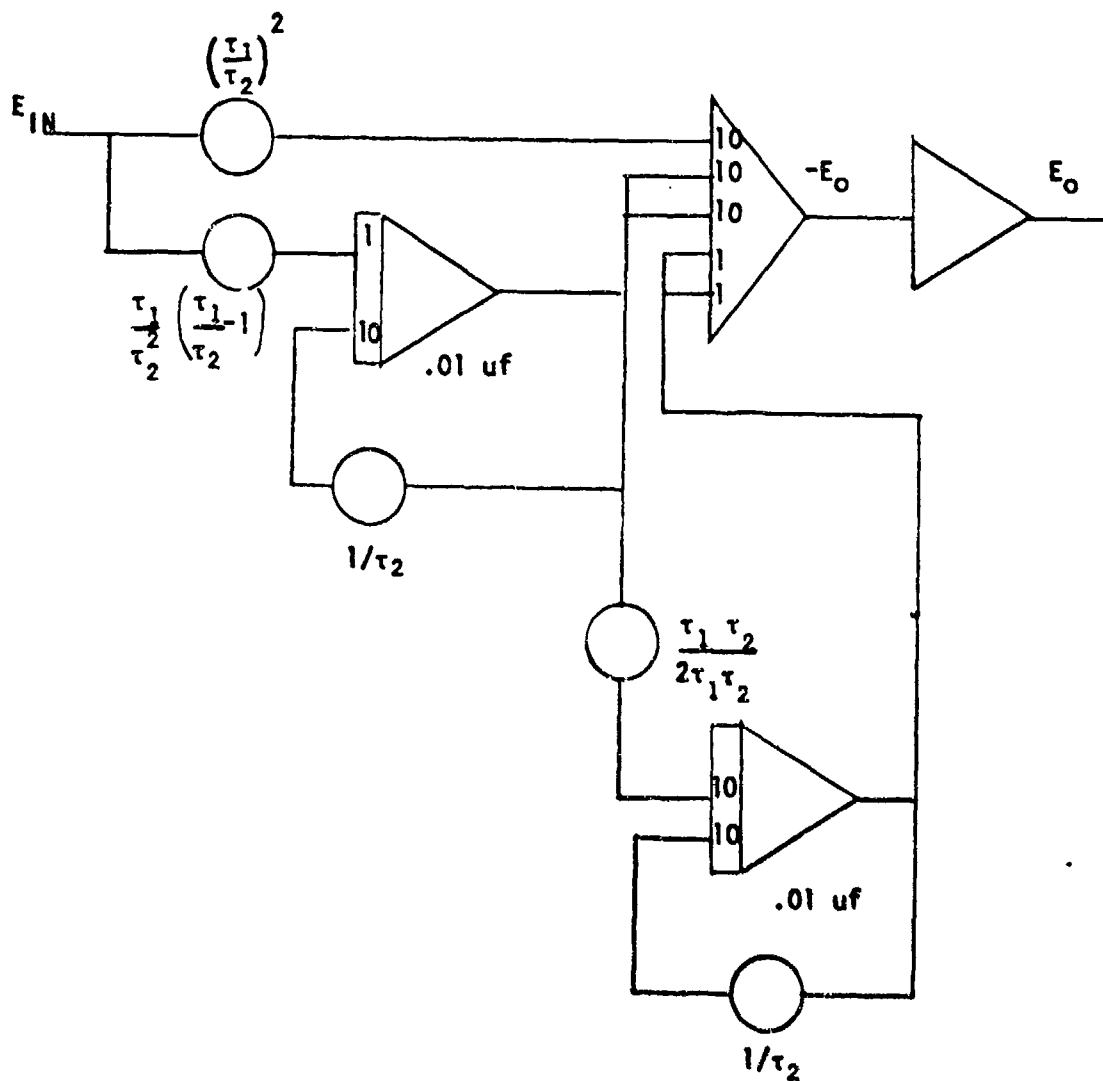


Figure 2.18 Analog Computer Mechanization of Lead Filter

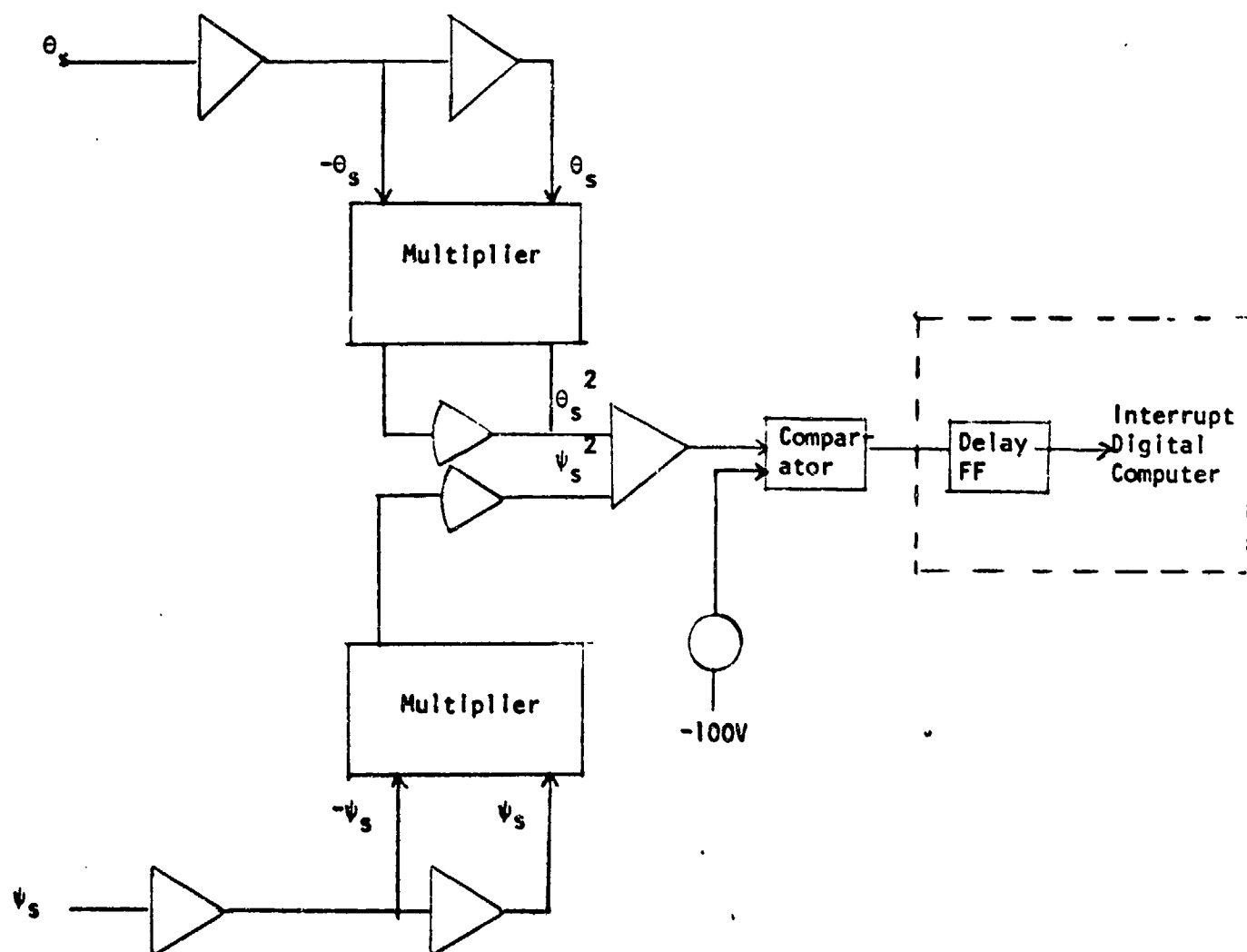


Figure 2.19 Analog Mechanization of Gyro Angle Run-Absort Control

at test points, to one of the analog computers, forming the total angle and comparing this with a pre-determined limit. If the limit is reached, a relay is used to set a logic true pulse which generates an interrupt signal to the digital control program enabling the simulation to be aborted before the gyro reaches its mechanical stops. The analog mechanization of this circuit is shown in Figure 2.19, assuming gyro angle components θ_s, ψ_s .

2.5 Target Simulators

The target simulators consist of laser and optical beam generators which project either a laser spot or slide view of a target respectively onto the rear of a translucent screen. The ground glass translucent screen causes a diffused image to be visible to hardware mounted on the Angular Motion Simulator. The laser simulates the reflection of a laser designator from a target, and the optical contrast generator simulates the view of a target as seen by an on-board video receiver. Position of the simulated target on the screen is controlled by deflecting the beam by means of one (optical simulator) or two (laser simulator) mirrors which rotate about axes in the plane of the mirror with two degrees of freedom. The position of the simulated target on the screen can be controlled to simulate the actual line of sight angles between missile and target when the missile seeker is mounted on the angular motion simulator. Target-missile closure is simulated by increasing the spot size in the laser simulator and zooming the slide display in the optical simulator. The simulator configuration in the laboratory is shown diagrammatically in Figure 2.20. Voltages to control mirror deflections, laser beam aperture position and zoom lens position may be set manually, derived from computer outputs (analog) or other external sources.

2.5.1 Angular Relationships

Displacement of the simulated target display to represent the relative motion of missile and target is governed only by the relative displacement coordinates between the missile and the target. The mathematics underlying the derivation of mirror deflection angles are given in reference 3, but will be summarized here.

Axes systems used in the derivation are as follows:

- (a) The Cartesian inertial system translating (or, instantaneously coincident) with the missile, with the CG as origin. Missile angular orientation is defined relative to these axes.
- (b) The Cartesian axis system defined for the Angular Motion Simulator (Z vertically downwards and Y horizontal).

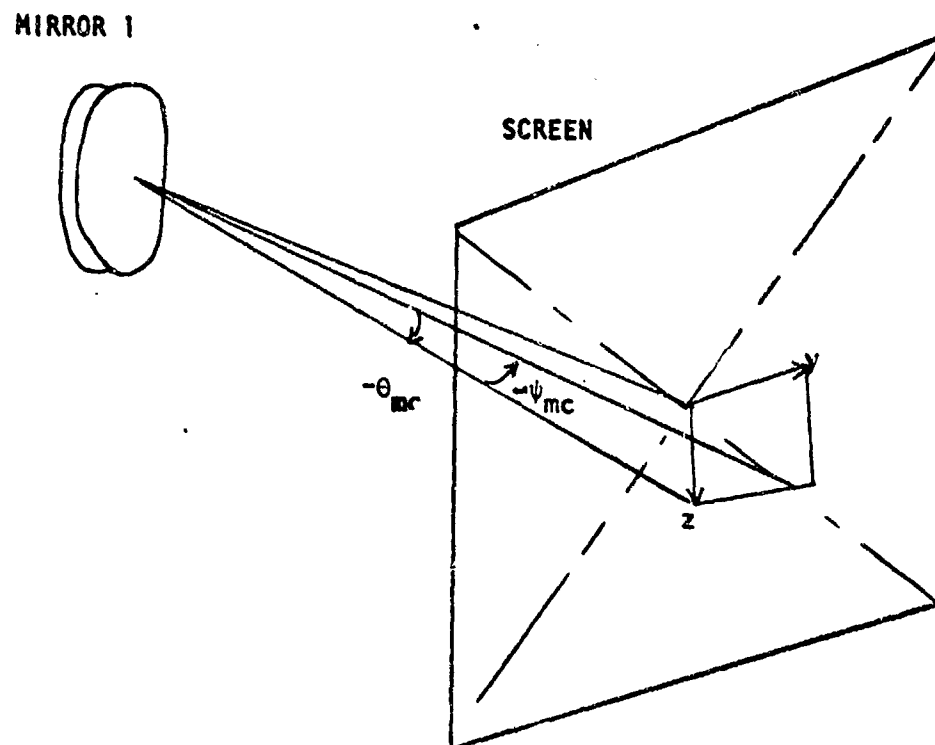


Figure 2.20 Mirror 1 Angles and Screen Axes

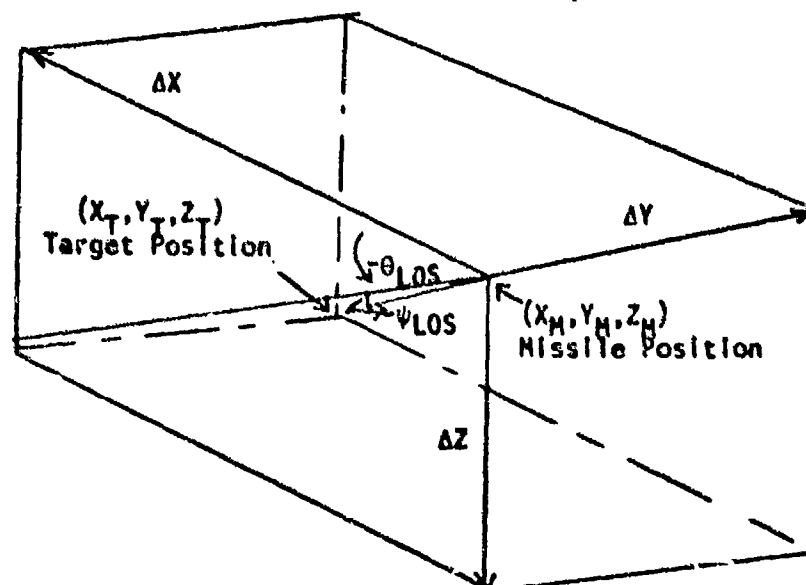


Figure 2.21 Target-Missile Line of Sight Angles

- (c) Two-dimensional axes y, z defining target display displacement on the screen, as shown in Figure 2.21.

Since the screen is approximately 5 feet by 5 feet, the range of line of sight angles which can be simulated are determined by the requirement of keeping the display on the screen without reaching the limits of mirror deflection. It is particularly important that the mirror does not reach its limit of travel if the seeker uses target line of sight rate. In order to select and control the range of line of sight angles which are to be displayed during a simulation, it is assumed that the axis system (a) is inclined to that of (b) at an angle which may be expressed as a component θ_1 about Y in (b) and ψ_1 about the Z in (a). Actual target missile relationships are illustrated in figure 2.22, where the line of sight angular components are functions of the missile-target separation distances ΔX , ΔY , ΔZ and are given by:

$$\tan \theta_{LOS} = \frac{-\Delta Z}{\Delta X}$$

$$\tan \psi_{LOS} = \frac{\Delta Y}{\sqrt{\Delta X^2 + \Delta Z^2}}$$

From these definitions, the displacement of the target display on the screen is (from Reference 3):

$$y = \frac{L_{ST} \cos(\theta_1 \cos \psi_1) \cos(\psi_1 \cos \theta_{LOS}) \tan(\psi_1 \cos \theta_{LOS} + \psi_{LOS})}{\cos \theta_1 \cos \psi_1 \cos(\theta_1 \cos \psi_1 + \theta_{LOS})}$$

$$z = \frac{-L_{ST} \cos(\theta_1 \cos \psi_1) \tan(\theta_1 \cos \psi_1 + \theta_{LOS})}{\cos \theta_1 \cos \psi_1}$$

where L_{ST} is the distance from the display screen to the Angular Motion Simulator center of rotation.

The angles of deflection of the laser or optical beam onto the screen are (Figure 2.21):

$$\theta_{mc} = \tan^{-1} \left\{ \frac{-z}{L_{SH}} \right\}$$

$$\psi_{mc} = \tan^{-1} \left\{ \frac{-y}{\sqrt{L_{SH}^2 + z^2}} \right\}$$

where L_{SM} is the distance from mirror number 1 to the screen.

For the particular case of $\psi_1 = 0$ in the above equations for y and z , we have

$$y|_{\psi_1=0} = L_{ST} \frac{\Delta Y}{\Delta X \cos \theta_1 + \Delta Z \sin \theta_1}$$

$$z|_{\psi_1=0} = L_{ST} \frac{\Delta X \tan \theta_1 - \Delta Z}{\Delta X + \Delta Z \tan \theta_1}$$

In the case where $\psi_1 = 0$ we have a common configuration wherein the actual inertial axes coincident with the missile are related to the laboratory axes (whose X axis is the horizontal line from Motion Simulator to screen) by a rotation θ_1 about their common Y axis. By varying θ_1 , a range of θ_{LOS} is selected for display on the target simulator screen.

For the general case of θ_1, ψ_1 values the inputs to the Angular Motion Simulator θ_T, ψ_T, ϕ_T are (from Reference 3):

$$\theta_T = \tan^{-1} \left\{ \frac{-b_{11} \cos \psi_1 \tan \theta_1 - b_{12} \sin \psi_1 \tan \theta_1 - b_{22}}{b_{11} \cos \psi_1 + b_{12} \sin \psi_1 + b_{22} \tan \theta_1} \right\}$$

$$\psi_T = \tan^{-1} \left\{ \frac{b_{11} \sin \psi_1 + b_{12} \cos \psi_1}{\sqrt{1 - (b_{11} \sin \psi_1 + b_{12} \cos \psi_1)^2}} \right\}$$

$$\phi_T = \tan^{-1} \left\{ \frac{-b_{31} \tan \theta_1 - b_{32}}{b_{21} \tan \theta_1 + b_{22}} \right\}$$

where b_{ij} are the elements of the transformation matrix from inertial to missile body axes through Euler angles θ_E, ψ_E, ϕ_E . Thus,

$$b_{11} = \cos \psi_E \cos \theta_E$$

$$b_{21} = \sin \phi_E \sin \theta_E - \cos \phi_E \sin \psi_E \cos \theta_E$$

$$b_{31} = \cos \phi_E \sin \theta_E + \sin \phi_E \sin \psi_E \cos \theta_E$$

$$b_{12} = \sin \psi_E$$

$$b_{22} = \cos \phi_E \cos \psi_E$$

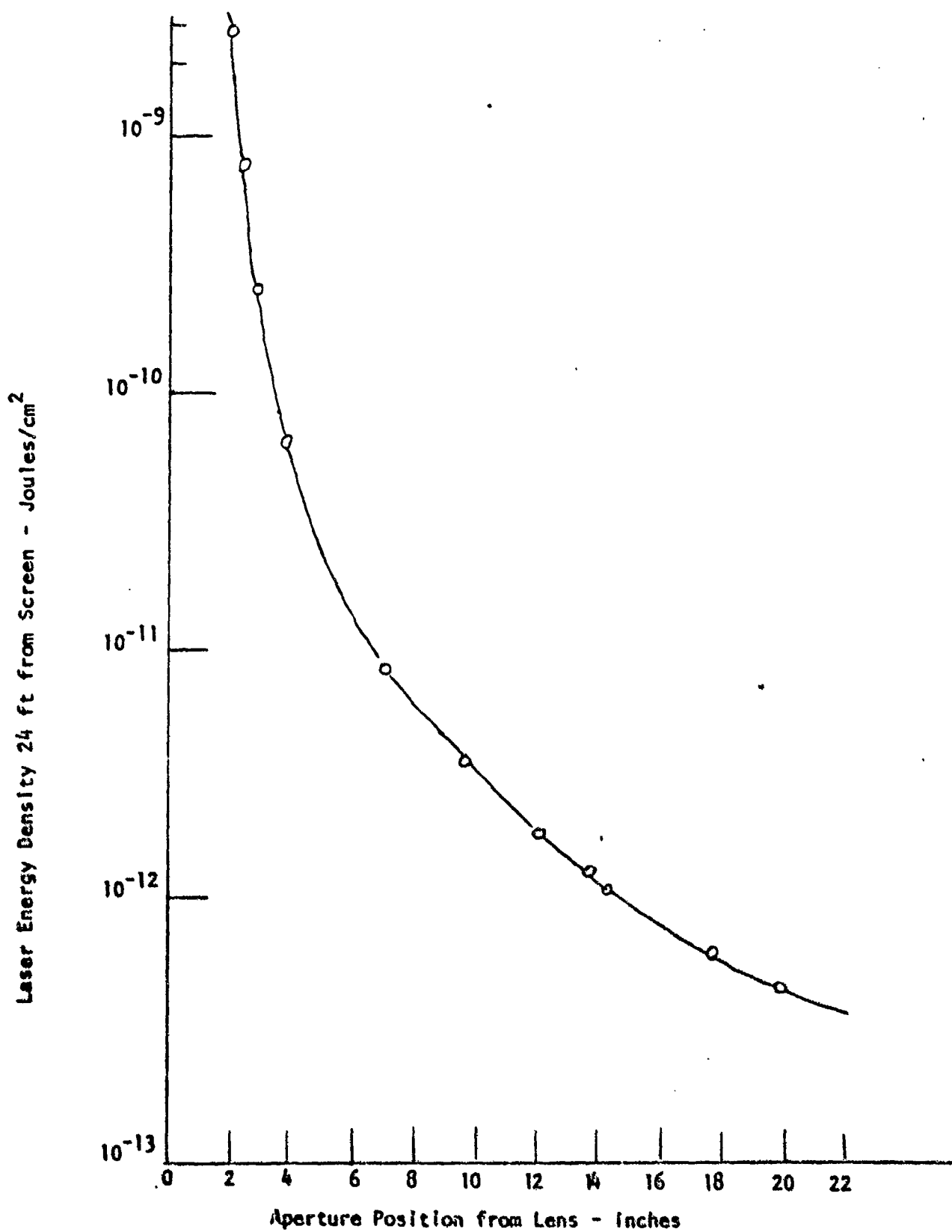


Figure 2.22 Laser Energy Density Variation with Aperture Position

$$b_{32} = -\sin\phi_E \cos\psi_E$$

$$b_{13} = -\cos\psi_E \sin\theta_E$$

$$b_{23} = \sin\phi_E \cos\theta_E + \cos\phi_E \sin\psi_E \sin\theta_E$$

$$b_{33} = \cos\phi_E \cos\theta_E - \sin\phi_E \sin\psi_E \sin\theta_E$$

For the case where $\psi_1 = 0$, the Angular Motion Simulator angles take the form:

$$\theta_T \Big|_{\psi_1=0} = \theta_E + \theta_1$$

$$\psi_T \Big|_{\psi_1=0} = \psi_E$$

$$\phi_T \Big|_{\psi_1=0} = \phi_E$$

2.5.2 Simulation of Target Closure

Closure of the range between target and missile is simulated for laser seekers by increasing the reflected spot size, and hence the reflected laser energy, proportionally according to the inverse of the slant range between target and missile. Laser spot size is varied by passing the laser beam through a convergent lens and then through a moveable knife-edge circular aperture, which produces a spot diameter ratio of 80:1 (38 db) between its closest and furthest positions from the convergent lens. A further 30 db of energy attenuation is available in 10 db steps by means of filter attenuators placed ahead of the convergent lens. Figure 2.22 shows the variation of laser energy density at the screen with aperture position which can vary from 2.25 inches to 22.0 inches from the lens.

Aperture position A_{PC} as a function of slant range from missile to target is given by the following:

$$R_{SR} = \sqrt{\Delta X^2 + \Delta Y^2 + \Delta Z^2}$$

then aperture input voltage A_{PC} is

$$A_{PC} = K_{AF} \text{ when } R_{SR} \geq R_{STFT}$$

$$A_{PC} = K_{AH} \text{ when } R_{SR} \leq \frac{R_{STFT}}{D_R}$$

otherwise $A_{PC} = M_a R_{SR} + B_a$

where $M_a = \frac{-D_R(K_{AN} - K_{AF})}{R_{STRT}(D_R - 1)}$

$$B_a = \frac{D_R K_{AN} - K_{AF}}{D_R - 1}$$

substituting for M_a and B_a gives

$$A_{PC} = \frac{D_R}{D_R - 1} \frac{R_{SR}}{R_{STRT}} (K_{AF} - K_{AN}) - \frac{K_{AF}}{D_R} + K_{AN}$$

where $K_{AF} = 62.4$ is the maximum voltage and $K_{AN} = -94.0$ is the minimum voltage. D_R is the ratio of largest to smallest spot size taking the smallest as unity, i.e. $D_R = 80$ and R_{STRT} is the slant range at which the spot size is assumed to start increasing. Note that the above expressions do not depend on the absolute values of R_{SR} and R_{STRT} but only their ratio. Also, the values of K_{AF} and K_{AN} are given for +100 volt analog range. For a digital program using the floating point option on a DAC these values are divided by 100.

In the case of the optical contrast simulator, target range closure is simulated by the use of a high-speed zoom lens with a magnification ratio of 50:1 and the capability of simulating closing speeds of up to 1600 ft/sec. Input voltage to the zoom control unit varies from zero to 100 volts and is given by the following polynomial fit to measured data:

$$Z_v = -25.7146 + 31.43604X_M - 5.395794X_M^2 + 0.3486925X_M^3$$

when $X_M < 7.075$

$$Z_v = 18.115 + 5.917363X_M - 0.2292095X_M^2 + 4.77775 \cdot 10^{-3}X_M^3$$

$$-3.812966 \cdot 10^{-5}X_M^4 \quad \text{when } X_M \geq 7.075$$

where Z_v is the zoom control voltage and

$$X_M = \frac{R_{SZ}}{R_{SR}} = \frac{R_{SZ}}{\sqrt{\Delta X^2 + \Delta Y^2 + \Delta Z^2}}$$

and R_{SZ} is the range at which the magnification is 1.0.

TABLE VIII

Mirror Correction Voltages* from Zoom Position

Zoom Voltage Z_v	Screen Magnification	$f\theta_c$ Volts	$f\psi_c$ Volts
0	4	0	0
5	4.82	-.08	-.08
10	5.83	-.14	-.08
20	8.18	-.29	-.23
30	12.7	-.40	-.44
40	19.0	-.43	-.52
50	28.3	-.43	-.63
60	41.5	-.24	-.69
70	61.8	-.02	-.62
80	90.0	-.02	-.60
90	131.0	-.03	-.33
100	197.0	-.03	.05

* All voltages are described as for computer input to the simulator
(+100 volts)

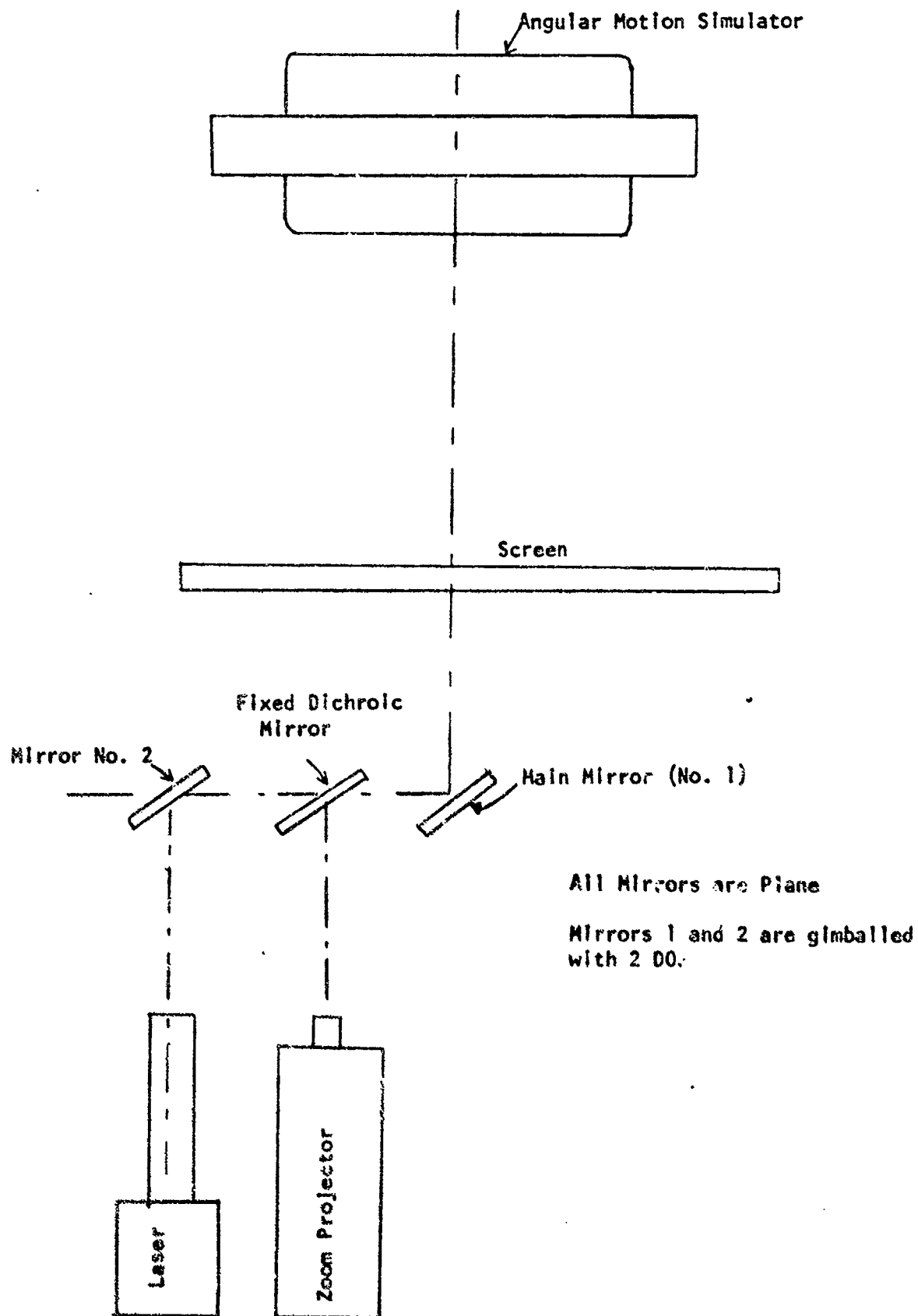


Figure 2.23 Simulator Layout (Horizontal Plane)

The process of zooming the optical contrast lens results in a rotation of the image on the display screen. The rotation is compensated for by adding a correction term to mirror angles (both mirrors, if both are being used) obtained by interpolation in tables for the θ_{MC} correction $f_{\theta C}$ and the ψ_{MC} correction $f_{\psi C}$ as functions of the zoom voltage Z_V . These data are shown in Table VIII. Addition of the corrections to both mirrors is necessary because of the opposite polarity of the mirrors; the object is to correct the zoom distortion by modifying mirror 1, but to eliminate the corrections from the laser deflection by inserting the inverse corrections in mirror 2.

Note that both optical and laser simulators will only be used simultaneously when using 'dual-mode seekers'. Target-missile motion will normally be simulated using mirror 1, and mirror 2, which has a range of motion of $\pm 3^\circ$ in each channel, will be used for second-order effects such as system noise and spot jitter.

2.5.3 Hardware Characteristics

The laser target simulator employs a pulsed Q-switched, neodymium-YAG laser operating at either 10 or 20 pulses/second. Physical layout of the simulator is shown in Figure 2.24. The circular gimballed mirrors are identical; each mirror has a range of deflection between mechanical stops of $\pm 15^\circ$ in both degrees of freedom. However, for the physical layout shown, the laser or light beam is deflected to the edge of the screen by a $\pm 9^\circ$ deflection of mirror number 1 and for the laser beam to impinge on mirror 1, mirror number 2 is limited to deflections of $\pm 3^\circ$. As a result, main control of the spot deflection is exercised through mirror number 1 and mirror number 2 is used to simulate laser spot positional noise and jitter and to oppose corrections to mirror 1 required by the zoom control of the optical contrast simulator.

The energy density of the laser spot on the screen may range from 1.2×10^{-15} to 4.3×10^{-9} joules/cm². Further details of laser performance, calibration and controls are given in Reference 4.

The optical contrast target simulator consists of a photographic slide projector equipped with a high-speed zoom lens which has a dynamic range of 50:1 in image magnification. Magnification at the smallest zoom setting is 4X, giving about 200X at the maximum zoom position. Image size is maintained at 17 inches x 17 inches on the screen. Screen illumination is greater than 4 ft-candles at all zoom positions for an input power to the projector lamp of about 2 kilowatts (lamp rated at 2.5 kilowatts). The time required to zoom the lens over its full range of travel is less than 3.9 seconds permitting the simulation of closing speeds up to about 1600 ft/second.

The projector employs a 2.5 Kilowatt Xenon lamp with an emission spectrum covering wavelengths from 0.2 microns to 1.1 microns. A series of optical filters is inserted between the lamp and slide to reject the heat producing near infra-red radiation at wavelengths above 0.72 microns. The heat is carried away by cooling fans. An alternative filter is available which permits radiation to 0.81 microns to pass through to the screen. Care should be exercised to limit the time at high magnifications to avoid damage to the slide when using this alternative filter. Neutral density filters are available for mounting on the zoom lens aperture to simulate various levels of atmospheric haze. Further details of set-up, use and calibration of the optical contrast simulator are given in Reference 5.

Voltage inputs to Mirrors 1 and 2, laser spot size controller and zoom position have been expressed as being in the range ± 100 volts. This is for analog computer compatibility purposes; a buffer amplifier with a gain of 0.1 is inserted in the input circuitry to the hardware since these components operate on ± 10 volts. The position of any of the above-mentioned hardware components may be obtained at any time by reading the output voltage of the desired unit. This voltage is in the range ± 10 volts. Since the mirrors have a restricted frequency response, as shown in Figures 2.25 and 2.26, it is often necessary to monitor the output of Mirror 1 to ensure that it does not diverge too greatly from the input waveform. Too great a divergence may result in unacceptably large errors in spot motion. Voltage-angle scale factors are non-linear for Mirrors 1 and 2; their variation is shown in Figures 2.27 and 2.28. It should be noted that a positive voltage in the mirror pitch control axis (θ_{mc}) produces a negative mirror angle, which is a downward deflection of the screen image. A positive yaw voltage produces a positive yaw mirror angle (ψ_{mc}) which is a leftward movement of the screen image when viewed from the Angular Motion Simulator.

2.3.4 Control and Patch Panels

The control console for the target simulators contains the following:

- (I) Laser simulator panel
- (II) Optical contrast simulator panel
- (III) Computer remote control panel
- (IV) Interconnection patch panel
- (V) Missile test panel

The laser and optical contrast panels each contain an input selection switch which permits input voltages to originate from analog computer

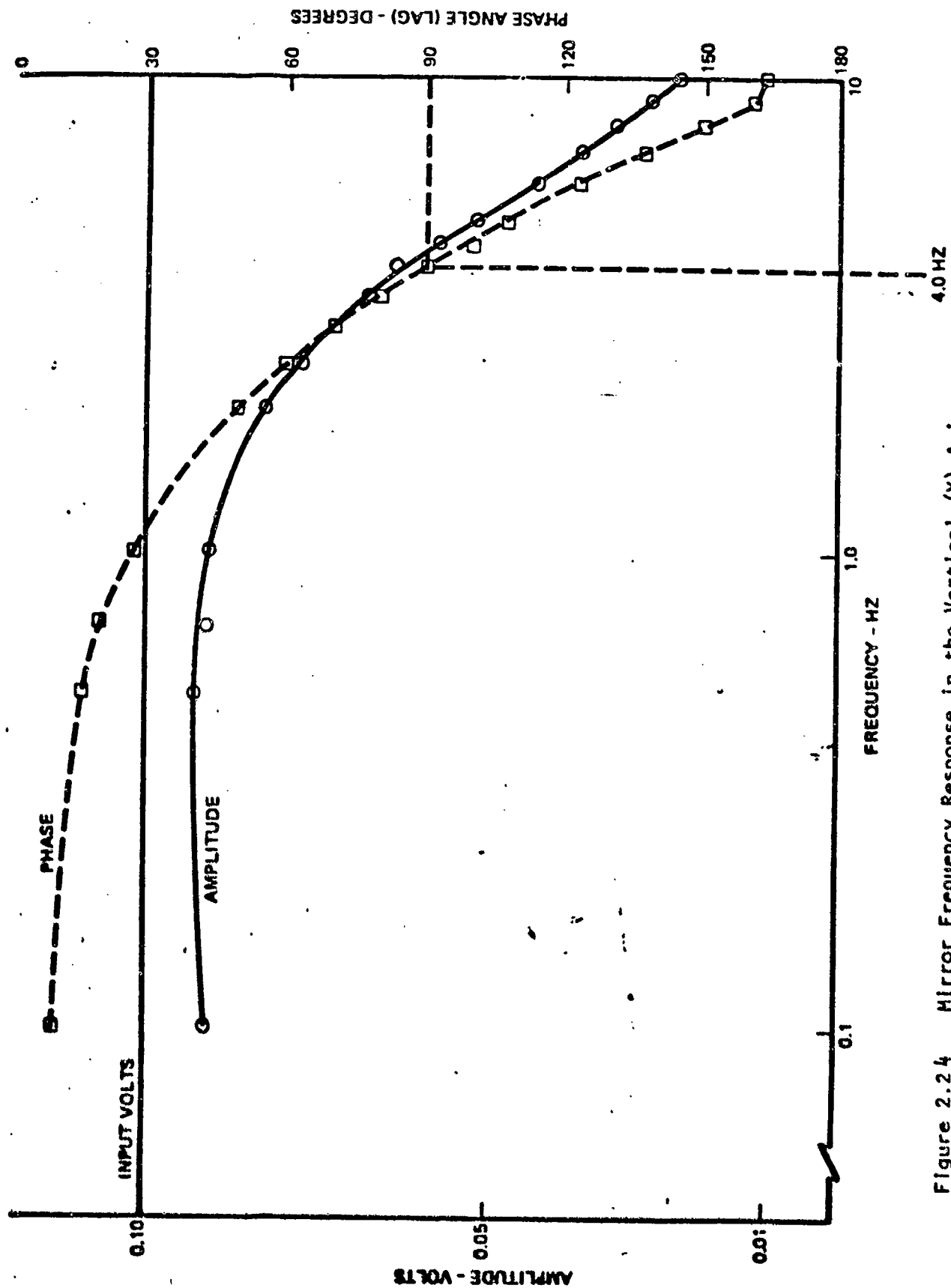


Figure 2.24 Mirror Frequency Response in the Vertical (Y) Axis

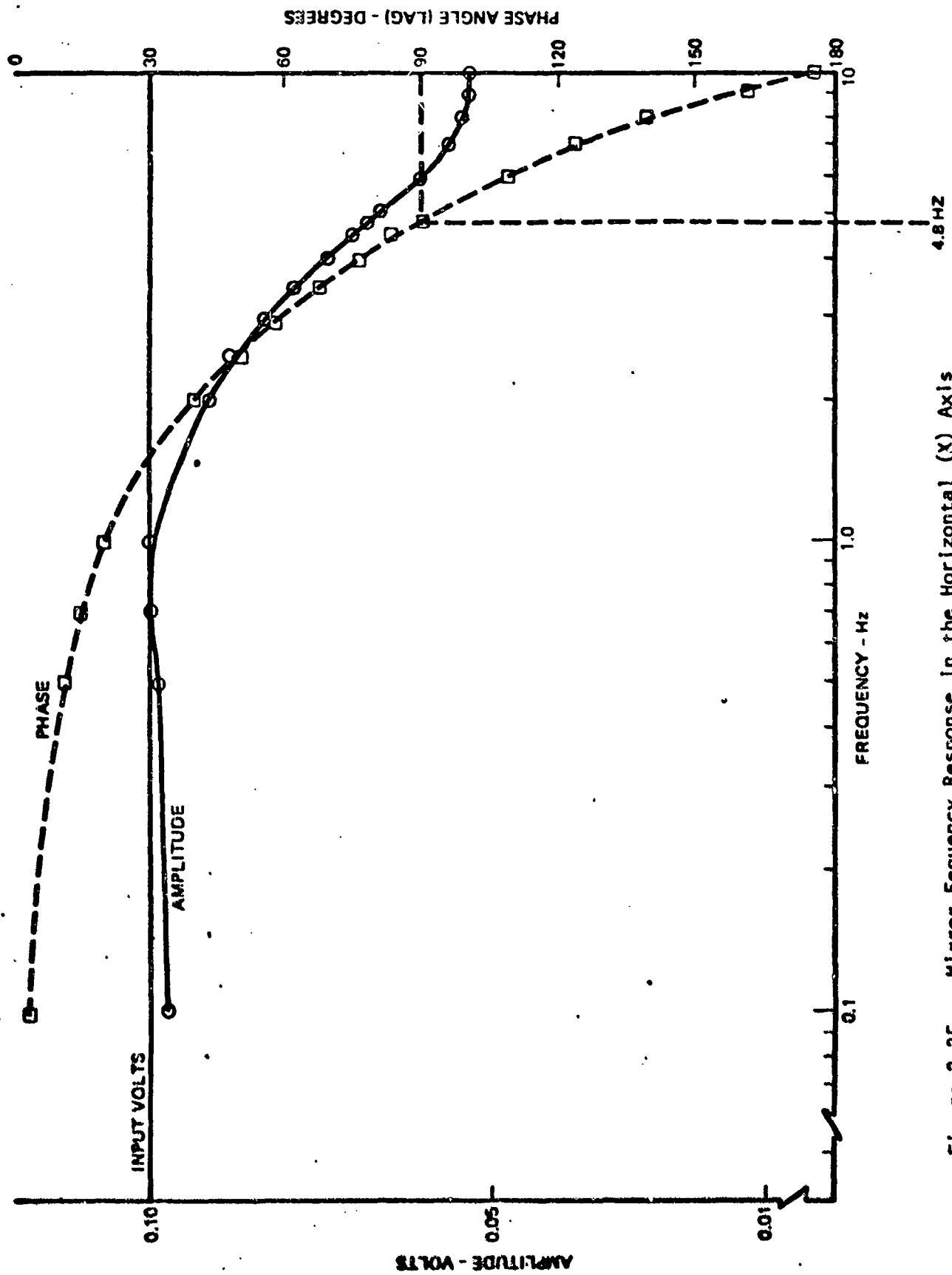


Figure 2.25 Mirror Frequency Response in the Horizontal (X) Axis

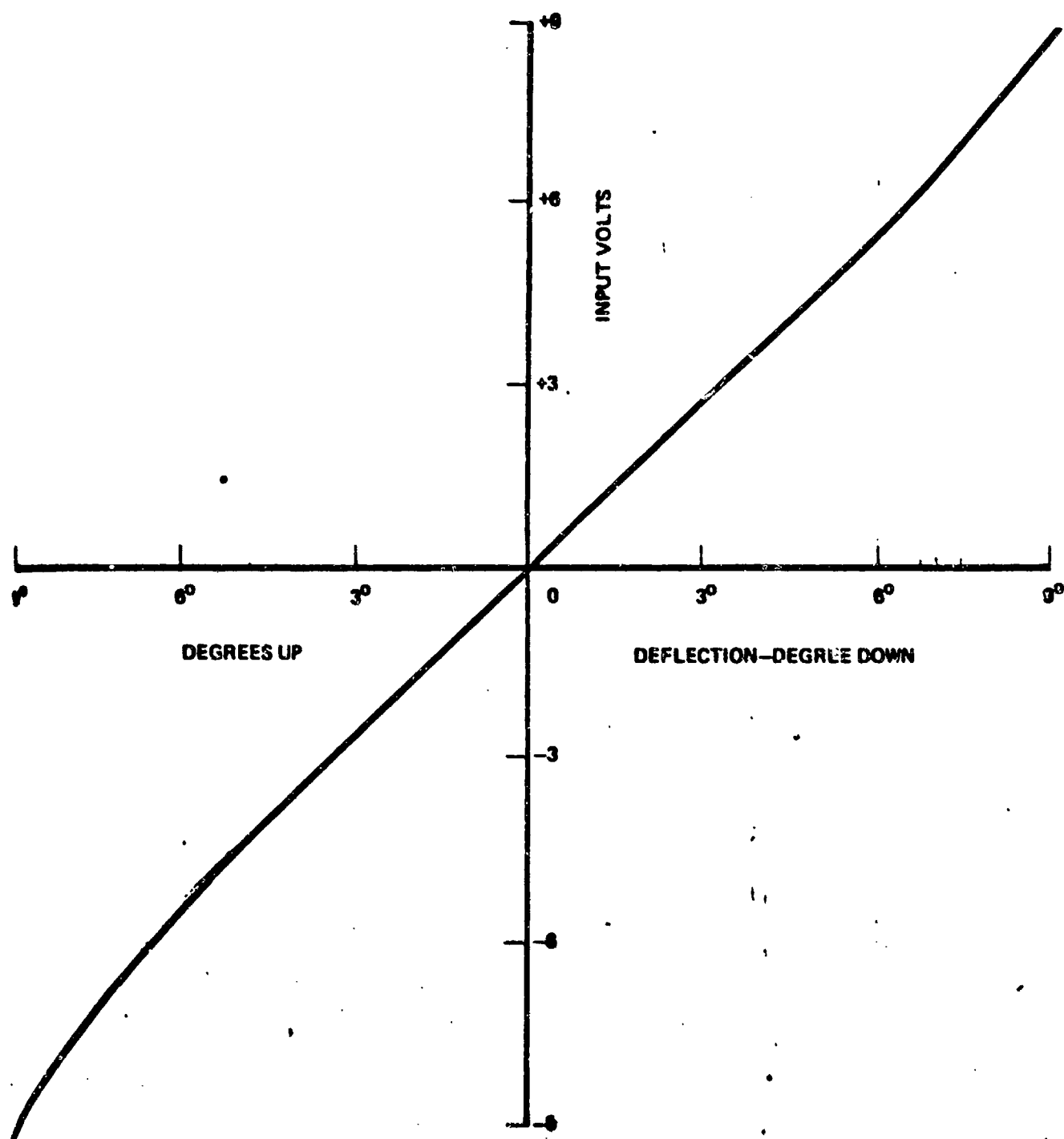


Figure 2.26 Dual Gimbal Mirror Calibration - Up/Down

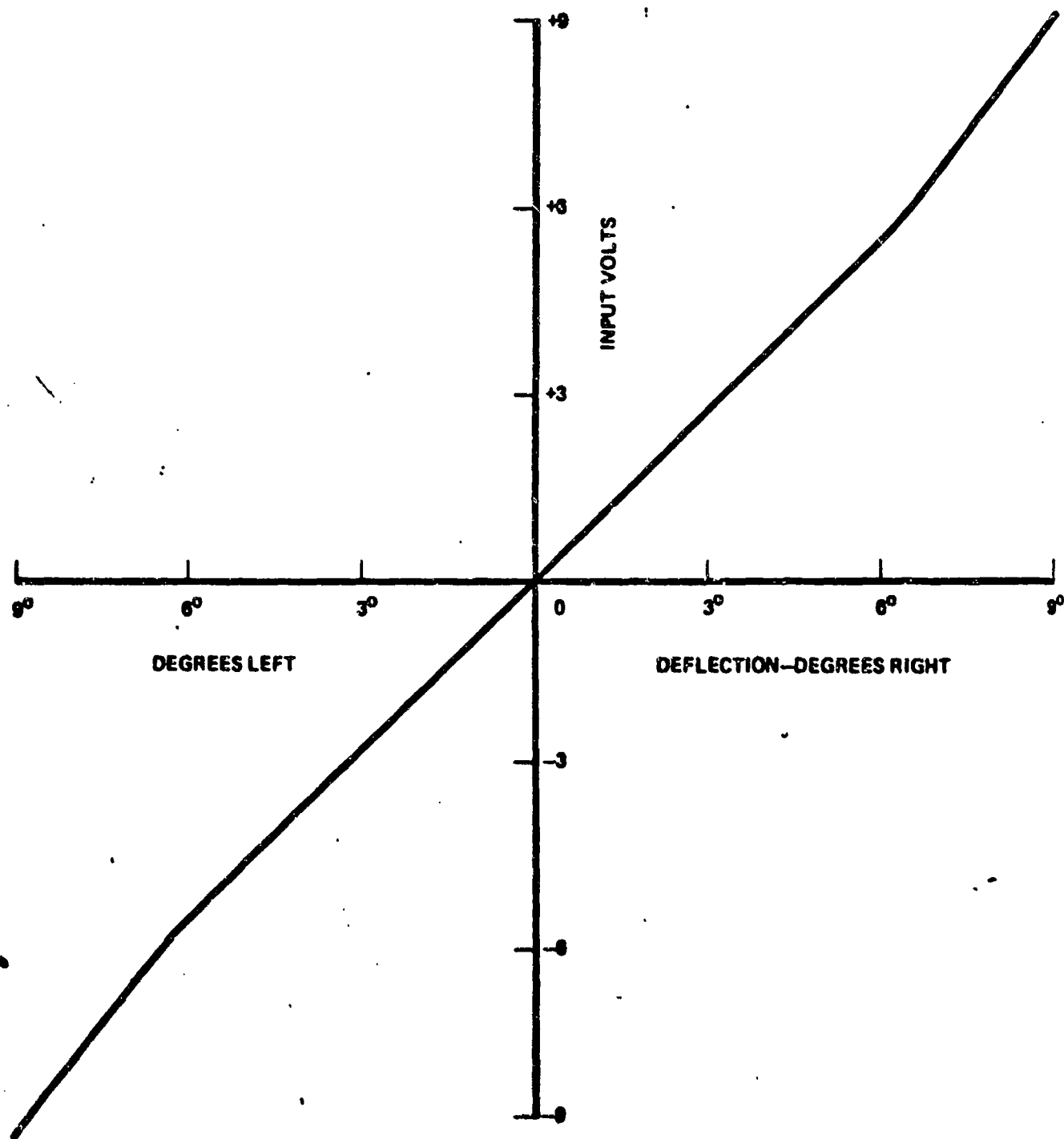


Figure 2.27 Dual Gimbal Mirror Calibration - Left/Right

output, manual control or external source. When manual control is selected three potentiometers on each panel control the value of input voltages to the two axes for each mirror and for zoom and spot size controller respectively in laser and optical simulators. In manual or external control voltage inputs are in the range ± 10 volts.

The computer remote control consists of a series of momentary push-button switches which permit the transmission of logic signals to the CI-5000 logic patchboard; these logic signals may then be used to generate an interrupt or sense line input to the digital computer. Each button may be lit from the rear under computer control to indicate the simulation status.

The interconnection patch panel is a 37-34 hole interchangeable patchboard for cross-connecting various elements. The holes are labelled horizontally A to Z, AA to MM and vertically 1 to 34 (from the top down). Included in the control console are sixteen operational amplifiers for use as summers/inverters. Inputs and outputs for these appear on the patchboard. Tables IX and X indicate the allocation of some of the patchboard connections.

The control console includes a missile test panel which contains test point outputs and control input monitoring points for a particular set of missile hardware. These monitoring and input points are connected to the patchboard as shown in Table X.

TABLE IX PATCHBOARD CONTENTS

Connection Element	Patchboard Coordinates	Connection Element	Patchboard Coordinates
<u>CI-5000 Trunklines</u>		<u>Laser Simulator Control Panel</u>	
T066	A-1	Mirror Pitch (computer)	C-14
T067	A-2	Mirror yaw (computer)	C-15
T068	A-3	Spot size (computer)	C-16
T069	A-4	Mirror pitch (External)	C-17
T070	A-5	Mirror yaw out	C-18
T071	A-6	Spot size out	C-19
T072	A-7	Mirror pitch (monitor)	C-11
T073	A-8	Mirror yaw (monitor)	C-12
T074	A-9	Spot size (monitor)	C-13
T075	A-10	Ground	C-20
T076	A-11	<u>Optical Contrast Control Panel</u>	
T077	A-12	Mirror pitch (computer)	C-21
T078	A-13	Mirror yaw (computer)	C-22
T079	A-14	Zoom (computer)	C-23
T080	A-15	Mirror pitch (external)	G-14
T081	A-16	Mirror yaw (external)	G-15
T082	A-17	Zoom (external)	G-16
T083	A-18	Mirror pitch out	C-17
T084	A-19	Mirror yaw out	C-18
T085	A-20	Zoom out	C-19
T086	A-21	Mirror pitch (monitor)	C-24
T087	A-22	Mirror yaw (monitor)	C-25
T088	A-23	Zoom (monitor)	C-26
T089	A-24	Ground	G-20
T090	A-25	<u>Missile Test Panel</u>	
T091	A-26	Test Point 12	G-1
T092	A-27	Test Point 13	G-2
T093	A-28	Test Point 14	G-3
<u>CI-500 Trunklines</u>		Test Point 15	G-4
T140	C-1	Test Point 16	G-5
T141	C-2	Test Point 17	G-6
T142	C-3	Test Point 26	G-7
T143	C-4	Test Point 27	G-8
T144	C-5	Ground	G-10
T145	C-6	Vane 1	E-1
T146	C-7		
T147	C-8		
T150	C-9		
T151	C-10		

Cont/...

TABLE IX PATCHBOARD CONTENTS cont.

Connection Element	Patchboard Coordinates	Connection Element	Patchboard Coordinates
<u>Missile Test Panel cont.</u>			
Vane 2	E-2		
Vane 3	E-3		
Vane 4	E-4		
Vane Ground	E-5		
Driv 1	E-6		
Driv 2	E-7		
Driv 3	E-8		
Driv 4	E-9		
Driv Ground	E-10		
Roll Monitor	E-11		
Yaw rate	E-12		
Pitch rate	E-13		
Roll gyro	E-14		
Seeker yaw	E-15		
Seeker pitch	E-16		
Launch timer	E-17		
Forward Loop ²	E-18		
Forward Loop ²	E-19		
Ground	E-20		

Patching Coordinates	Connection	Patching Coordinates	Patching Connection	Coordinates	Patching Connection	Coordinates	Patching Connection
U-1	1 gain	W-1	1 gain	U-7	1 gain	W-7	1 gain
U-2	1 gain	W-2	1 gain	U-8	1 gain	W-8	1 gain
U-3	10 gain	W-3	10 gain	U-9	10 gain	W-9	10 gain
U-4	SJ	W-4	SJ	U-10	SJ	W-10	SJ
U-5	Output	W-5	Output	U-11	Output	W-11	Output
U-6	Output	W-6	Output	U-12	Output	W-12	Output
V-5	Output	Z-5	Output	V-11	Output	X-11	Output
V-6	Output	Z-6	Output	V-12	Output	X-12	Output
Y-1	1 gain	AA-1	1 gain	Y-7	.1 gain	AA-7	1 gain
Y-2	1 gain	AA-2	1 gain	Y-8	.1 gain	AA-8	1 gain
Y-3	10 gain	AA-3	10 gain	Y-9	1 gain	AA-9	1 gain
Y-4	SJ	AA-4	SJ	Y-10	SJ	AA-10	SJ
Y-5	Output	AA-5	Output	Y-11	Output	AA-11	Output
Y-6	Output	AA-6	Output	Y-12	Output	AA-12	Output
Z-5	Output	BB-5	Output	Z-11	Output	BB-11	Output
Z-6	Output	BB-6	Output	Z-12	Output	BB-12	Output
CC-1	1 gain	EE-1	1 gain	CC-7	.1 gain	EE-7	.1 gain
CC-2	1 gain	EE-2	1 gain	CC-8	.1 gain	EE-8	.1 gain
CC-3	10 gain	EE-3	10 gain	CC-9	1 gain	EE-9	1 gain
CC-4	SJ	EE-4	SJ	CC-10	SJ	EE-10	SJ
CC-5	Output	EE-5	Output	CC-11	Output	EE-11	Output
CC-6	Output	EE-6	Output	DD-11	Output	EE-12	Output
DD-5	Output	FF-5	Output	DD-12	Output	FF-11	Output
DD-6	Output	FF-6	Output	CC-12	Output	FF-12	Output
GG-1	1 gain	JJ-1	1 gain	GG-7	.1 gain	JJ-7	.1 gain
GG-2	1 gain	JJ-2	1 gain	GG-8	.1 gain	JJ-8	.1 gain
GG-3	10 gain	JJ-3	10 gain	GG-9	1 gain	JJ-9	1 gain
GG-4	SJ	JJ-4	SJ	GG-10	SJ	JJ-10	SJ
GG-5	Output	JJ-5	Output	GG-11	Output	JJ-11	Output
GG-6	Output	JJ-6	Output	GG-12	Output	JJ-12	Output
HH-5	Output	KK-5	Output	HH-11	Output	KK-11	Output
HH-6	Output	KK-6	Output	HH-12	Output	KK-12	Output

TABLE X Patchboard Operational Amplifier Connections

3.0 SOFTWARE

3.1 Monitor/Operating System

All digital and hybrid (except specially written 'stand-alone') programs execute on the Sigma 5 under the control of an operating system program (also called a monitor system). The monitor currently used is an adaptation of a standard Xerox software product called RBM and is known as HRBM (for Hybrid Real-Time Batch Monitor). HRBM handles all input/output through the MIOP, handles job start and termination, controls the allocation of core memory and use of disk storage space, provides interrupt connection/release/processing including loading and executing real-time tasks. A detailed description of RBM is given in References 6, 7 and 8 and of HRBM in Reference 9. An overview of HRBM will be given here and the modifications to RBM in forming HRBM will be indicated.

Core memory is divided into three regions described as follows in order of increasing addresses:

- (i) Core-resident HRBM
- (ii) Background area
- (iii) Foreground area

The division of memory not used by (i) into regions (ii) and (iii) is determined initially at the time the monitor system is generated, but is alterable by operator key-in at the teletype (RBM) and by a program control card (HRBM only); if the foreground area is to be truncated but contains a program the alteration is not accepted by the monitor and is ignored; a message to this effect is typed on the operator console (teletype).

Two batch job streams are permitted in HRBM (one in RBM), which execute (not simultaneously) in the background area. The two streams are labelled 'local' and 'remote' with all local jobs having priority over remote jobs. A card reader and line printer is allocated to each stream and readers and printers may be switched by operator key-in. If a remote job is executing and a local job is entered at the local card reader then the remote job is suspended (known as 'checkpointed' in References 6, 7, 8 and 9), the memory contents and CPU context saved on a RAD file, and the local jobs processed. The remote job is restored from the RAD file and execution continued after all local jobs have been completed. Since there is no protection of the I/O devices which can be accessed by both local and remote jobs, care must be exercised to avoid a checkpointed job being jeopardized by interference with its uncompleted use of an I/O device. I/O devices in this category are the magnetic tape drive and the graphics display unit.

The foreground region is used by programs which respond to external interrupts and thus may interrupt execution of background processing (which may be local or remote). Foreground programs may occupy any part of the combined background/foreground areas; when a foreground program uses a part of the background region an execution background program is checkpointed as described above but on a different RAD file. Foreground programs may be initiated from the background job streams, or from previously compiled and loaded RAD files by operator key-in. All compilers, assemblers and loaders (or link-editors) execute as background jobs.

Card reader input and line printer output in HRBM is handled through what are called 'I/O symbionts'. This means that the card inputs and print outputs are buffered via RAD files; all cards are input via HRBM to the RAD buffer file and all requests to read a card by a background program (foreground programs are not permitted to read cards) are diverted to read the next available card image from the RAD buffer. Similarly, printer output is intercepted in HRBM and stored on a RAD file. Output to a printer thus occurs asynchronously with the program originating the output at a rate dependent on print line availability and printer speed. The symbiont I/O system is driven by internal clock 3 which interrupts processing every 200 milliseconds to initiate card reading and line printing on both readers and both printers if required. Once initiated, reading and printing are continued by the I/O interrupt response routines. If, however, the clock 3 interrupt routine finds an external interrupt of priority higher than 10 active then no initiation action is taken in order to minimize interference with high priority foreground jobs. Local background and foreground printer outputs use different RAD file buffers but share the same printer.

Most components of RBM use re-entrant coding which thus precludes the use of extra register blocks. HRBM uses the extra three register blocks mentioned in Section 2.1.1 as follows:

- (I) Register Block 1 is assigned to the clock 3 interrupt routine where re-entrancy is prevented by disabling the clock 3 counter equals 0 interrupt until just prior to exit from the interrupt routine.
- (II) Register Block 2 is assigned to the clock 4 interrupt routine and the control panel interrupt routine. Clock 4 is used to maintain time of day and the date.
- (III) Register Block 3 is assigned to the DMA channel interrupt response routine.

Extensions to the list of unsolicited key-ins for HRBM are indicated in Table XI. Note that the SS key-ins are only necessary to intervene

TABLE XI HRBM Operator Keyins

Keyin Characters*	Effect
MST [F] [B]	Permit a foreground (F) or background (B) program to select Master mode in the CPU
SLV [F] [B]	Inhibit a foreground (F) or background (B) program from selecting Master mode in the CPU
SDS P R	Exchange the designation of local and remote printers (P) or card readers (R)
SS F B	initiate foreground (F) or background (B) output symbiont printing

* Brackets [] Indicate a character to be selected which must be preceded by a blank.

in the current output to the local line printer in order to switch the output from foreground to background or vice versa. Under normal circumstances, as soon as the RAD output buffer being printed is empty the symbiont I/O control routine switches to the other output buffer. If simultaneously core-resident background and foreground jobs are both producing printer output this may result in the two sets of outputs being intermixed. It may be necessary to direct the foreground output to a RAD file and then print the file by means of a separate background job.

As an added convenience in HRBM the SY, FG and FMEM keyins of RBM have been extended to become control commands with formats:

```
!SY  
!FG  
!FMEM n
```

and are inserted in a job deck at the points where the keyins would normally be required. Note that for the FMEM command the foreground area should not be occupied if it is being allocated to background, otherwise the command is ignored with an operator message to this effect. These control card commands have exactly the same effect as the corresponding keyin commands.

A further extension included in HRBM are CAL instruction responses to meet the requirements of the input/output control from/to the graphics display terminal; this control includes central connection of and response to external interrupts 6 and 7 for character receipt and transmission. Further details are given in Section 3.5.

3.2 Compilers and Assemblers

The system software contains two FORTRAN compilers and an assembler, briefly described as follows:

- (I) Extended H level FORTRAN (referenced by file name FORTRANH in the SP area of RAD storage). This compiler is described in Reference 10 and is a superset of standard FORTRAN IV. It is a one-pass compiler designed to be compatible with IBM 360 H level FORTRAN; however, in-line assembly language coding may be introduced if required.
- (II) Extended FORTRAN IV (referenced by file name FORTRAN in the SP area of RAD storage) is also a superset of standard FORTRAN IV but contains many more extended features than H FORTRAN. Additionally, it is designed for rapid compilation and produces highly efficient compiled code. Reference 11 contains details of the compiler and its use. This FORTRAN compiler requires a larger amount of core space than does the H level compiler, and also makes greater demands on the temporary background (BT)

RAD files. For a program which contains one or more relatively large subroutines it is necessary to increase the allocated size of the X1 scratch file with a card which precedes the !FORTRAN card and is punched as follows:

!ALLOBT (FILE,X1),(FSIZE,200),(RSIZE,202)

For very large routines it is sometimes necessary to insert a similar card for the X2 file.

- (iii) Assembler (referenced by file name MACRSYM in the SP area of RAD storage). This is a two pass assembler with extensive macro facilities. Input (i.e. assembly language programs) may be in the form of source records (card images) or of compressed symbolic records. In the latter case an updating facility is provided and output of a compressed symbolic deck may be specified. Further details are given in Reference 12. The language is named MACROSYMBOL; a useful feature is the capability of automatically including within a program the contents of a source file on the RAD by means of a SYSTEM directive and a control card entry.

3.3 Hybrid Computer Control and I/O Routines

A set of library routines is provided which enables a FORTRAN or MACROSYMBOL program to communicate with the analog computers via the A/D Interfaces. These library routines are formed into a pre-compiled and linked Public Library which is stored in a RAD file named HYBLIB in the FP area. To use this Public Library a field containing (PUBLIB, HYBLIB) must be included in the users !LOAD card (see Reference 6, Section 6). The Public Library is capable of being shared by foreground and background programs. A description of the method of use of these routines is given in Reference 2.

The hybrid library routines permit communication with both analog computers (CI-500 and CI-5000) and provide control of the analog-digital interfaces and the high-speed DMA channel. A brief list of the operations performed is as follows:

- i) read any addressable analog component
- ii) set potentiometers
- iii) read analog status lines
- iv) read the digital voltmeter

- v) set the analog reference dividers
- vi) control the analog modes
- vii) send analog codes
- viii) write MDAC's
- ix) read ADC's
- x) control the real-time clocks
- xi) read sense lines (discretes)
- xii) write control lines (discretes)
- xiii) control external interrupts

For real-time operation the concept of "patterned" execution is implemented. In this usage, patterns of hybrid I/O and control operations are defined prior to real-time execution as a series of computer instructions in the user program temp stack. Then, during real-time operation, the pattern is invoked which causes execution of the pre-stored stack of instructions resulting in minimum overhead time during real-time operation.

The hybrid computer library routines use the CAL3 and CAL4 instructions and therefore an assembly language user program should not use these instructions for any other purpose. Hybrid library routines contain extensive error checking and diagnostic outputs to cover illegal use of their entry points. Construction of the library permits the concurrent use of the routines by various foreground and background programs. However, connection of interrupts already assigned to a program is prevented, as is the arming and enabling of interrupts which have not been connected through the interrupt connecting routine.

The task control block (TCB) for each connected interrupt is also allocated to the user program temp stack. Thus, the use of patterns and interrupts should be considered when allocating a user program temp stack size.

3.4 CSSL

CSSL (Continuous Systems Simulation Language) is a digital computer problem-oriented language for the representation of continuous dynamic systems which can be modeled by sets of ordinary differential equations. Language definitions and methods of use are given in Reference 13. A program formed from CSSL statements is converted by the CSSL translator into a FORTRAN program which is then compiled, loaded and executed in the usual manner.

CSSL permits both equation based and block oriented representation statements and is thus suitable for a wide range of users.

The language translator implemented for the Sigma 5 requires three scratch files in the background temporary area of the RAO. These files are X4, X5 and X6 with blocked format, record size 128 words and file size 400 records for an average program. A large program will require increased file sizes. The translated output is a series of FORTRAN statements and the output device is usually a magnetic tape; the translated program is also a foreground program with the first subroutine connected to interrupt 14 (interrupt address 6E₁₆). However, if real-time execution is not required the first three statements of the output may be deleted, in which case the FORTRAN output may be executed as a background job, as determined by the !OLOAD card.

The control card set-up for a CSSL program is as follows:

```
!JOB 082274,ACJ
!REW 9TA80
!ALLOBT (FILE,X4),(FORMAT,B),(RSIZE,128),(FSIZE,400)
!ALLOBT (FILE,X5),(FORMAT,B),(RSIZE,128),(FSIZE,400)
!ALLOBT (FILE,X6),(FORMAT,B),(RSIZE,128),(FSIZE,400)
!CSSL
```

CSSL Source Cards

```
!WEOF 9TA80
!REW 9TA80
!PREC 9TA80,3
!ASSIGN (M:SI,9TA80)
!FORTRAN SI,GO,BC,NS,LS
!OLOAD GO,(TEMP,400),(LIB,USER,SYSTEM)
!ROV
!FIN
```

Note that the !PREC card skips the first 3 statements to avoid connecting the program to an interrupt. If foreground operation is required the (FORE) field is required on the !OLOAD and an !FG card is required before the !ROV card.

3.5 Graphics Software

As mentioned in Table 1, the graphics terminal is a Tektronix Model 4002A. The terminal comprises a storage tube display, keyboard input,

hardcopier and a "joystick" driven cross-hair cursor. The 4002A is connected to the Sigma 5 by means of a Xerox 7611 character oriented communications controller which transmits and receives ASCII 8 bit characters. External interrupts 6 and 7 (interrupt locations 66₁₆ and 67₁₆) are used by the character receive and transmit circuitry respectively to indicate completion status to the Sigma 5. Character transmission rate is currently set at 9600 Baud; the 7611 has a capacity of 64 lines of which only one (line no. 0) is used by the 4002A.

Standard software which facilitates the use of the graphics terminal is provided in three packages, which are briefly described as follows:

1) Hardware I/O Control and Utility Routines.

This package comprises a set of FORTRAN callable assembly language routines (MACROSYMBOL) which perform the software interfacing function between the Sigma 5 and the 4002A. Entry point names, parameter definitions and operating functions may be described as follows:

CALL ST7611 - this entry point initializes the software package by resetting the 7611 and checking that it is switched on. Interrupts 6 and 7 are connected centrally and various counters initialized. This entry point should be called once before use of the following entry points.

CALL WT7611(I) - transmits one ASCII character contained in the rightmost byte (i.e. byte 3) of I to the display screen. The character may be of 7 bit form without the parity bit; this is calculated and inserted by the routine. Note that on return from this call the character has not necessarily been transmitted to the screen since the output is buffered.

CALL RD7611(N, ICHAR, IWT) - reads N characters from the terminal and stores them in consecutive bytes in array ICHAR. Parameter IWT is used to indicate whether it is desired to return control to the calling routine before or after the read operation is complete. If IWT is non-zero control is not returned until the requested number of characters have been received, and if IWT is zero control is returned immediately after the input has been initiated. In the latter case the NSTCK entry point may be used to determine the status of the I/O operation.

CALL HT7611 - is the terminating call to the software package which stops I/O activity and disconnects the external interrupts.

The remaining entry points are used as FORTRAN function calls:

NEBCD(I,J) - returns the EBCDIC character corresponding to the 8 bit ASCII character contained in the J'th byte of array I counting from 1 at the left of the first word. Thus, characters read by the RD7611 entry point into I may be conveniently converted into the SIGMA 5 internal code.

NASCII(I,J) - returns the ASCII character corresponding to the 8 bit EBCDIC character contained in the J'th byte of array I counting from 1 at the left of the first word.

NDIG(I,J,K) - when K is positive the J'th byte taken from array I is converted to the equivalent binary integer if it is an ASCII numeric character but otherwise returns a zero. If K is zero the least significant 5 bits of the J'th character are returned as a binary integer and if K is negative the least significant 7 bits are returned.

NSTCK(I) - is used to check the I/O status where I is a dummy parameter. Values returned are integers with the following meanings:

- 0 The software is uninitialized
- 1 Software has been initialized but no I/O is in progress
- 3 Characters are being output to the terminal
- 5 Input from the terminal has been requested but the required number of characters have not been received
- 7 Both input from and output to the terminal are currently in progress

Any other code means the same as zero.

11) Tektronix Terminal Control System - 4002A

This package consists of a set of FORTRAN routines which facilitate the generating of displays and provides convenient control of the screen contents. Reference 14 contains full details of the operation of use of this package.

The package contains routines for direct and relative vector drawing, controlling and positioning alphanumerics including

tabs and margins and provides support for the scratchpad and edit modes. The concept of virtual graphics is also implemented which permits the generation of displays on a virtual screen larger than the actual screen. Portions of the virtual display may then be viewed through one or more screen "windows" which may rotate or scale the actual display relative to the virtual window.

III) Tektronix Advanced Graphing II

This package consists of a set of FORTRAN routines which are useful in generating graphs. A larger number of options are included in an attempt to provide a flexible means of drawing all the types of graph normally encountered. Full details are provided in Reference 15.

All the software described in this subsection is stored on RAD files in the D6 area in source and relocatable binary form. File names are given in the following table:

Software Package	File Names	
	Source	Relocatable Binary
COC I/O Control	MOCPACK	MOCAG2BO
Terminal Control System	TCSGRAP	GRAPHICS
Advanced Graphing II	AG2FILE	AG2FILBO

TABLE XII GRAPHICS SOFTWARE FILE STORAGE

3.6 Utility Routines

Software under this heading consists of a background processor which performs various editing and transmission functions on any symbolic source card images. The processor is written in MACROSYMBOL assembly language and is held in the system programs area of the RAD in core image format in a file named UPDATE. It may thus be loaded into core memory for execution by an !UPDATE control card.

The operations of this editing processor are controlled by a series of commands provided by the user and read by UPDATE from the C device. Input and output source records may be in compressed or uncompressed form. The compressed symbolic form is that used by the MACROSYMBOL assembler.

Source files are considered conceptually to consist of a set of 'decks', the delimiters of a deck being records containing END (except for the first deck in a file, where the first record is the initial delimiter). Thus, UPDATE is designed to handle editing of source programs which may contain FORTRAN and MACROSYMBOL decks intermixed. However, because of the different requirements for END cards in FORTRAN and MACROSYMBOL a command is provided to indicate which type of END card is used as a delimiter for the current deck.

UPDATE control commands permit the following operations:

- . copy decks from an input to an output device
- . during a deck copying operation insert and delete records in the deck
- . control the selection of input, output, command or listing devices during deck transferral and thus permit the deletion or addition of decks to a file
- . convert source code from IBM 026 code to the 029 code if required
- . write end of file on the output device
- . rewind selected devices
- . assign DCB's corresponding to operational labels to RAD files or hardware devices

The control commands to perform these operations are described as follows. Note that each control command format is :X followed by a single blank where X is one of the commands described below. Punching always starts in card column 1.

I) :C xx

Specifies the command input device where xx is its operational label and may be C, OC, CI or SI. The default is C

II) :I xx

Specifies the input device where xx is its operational label and may be SI or CI. The default is SI and if CI is specified the source must be in standard MACROSYMBOL compressed symbolic form.

iii) :O xx

Specifies the output device where xx is its operational label and may be S0, C0 or D0. D0 is deleted output, in which case the decks being copied are deleted until the output device is changed. C0 produces standard compressed symbolic output. The default is S0.

iv) :L xx

Specifies whether or not the output file listing is required where xx is L0 for listing and N0 for no listing. The default is N0. The listing will appear on the system L0 device and will have the input source line numbers with all insertions flagged.

v) :A xx,yy,nnnnnnnn

Assigns the DCB corresponding to operational label xx to device yy (TY for TYA01, CR for CRA03, GT for GTA80) or to RAD area yy (SP, FP, BP, BT, DI, ...,DF). If the assignment is to a RAD file area nnnnnnnn specifies the file name but if yy is a RAD area and the file name is omitted the DCB is assigned to the entire RAD area. xx may be SI, CI, OC, C, S0 or C0.

vi) :FOR

Specifies that decks are to be delimited by FORTRAN END cards. The default is MACROSYMBOL.

vii) :MAC

Specifies that decks are to be delimited by MACROSYMBOL END cards.

viii) :EOF

Writes an end of file mark on the output device.

ix) :REW xx

Rewinds the device assigned to operational label xx where xx may be SI, CI, S0 or C0. If xx is unspecified the output device will be rewound.

x) :B

Specifies that 026 to 029 conversion "switch" is to be toggled.

The default is no conversion. The first time this card is encountered conversion of the input begins and the next time a card of this type is encountered the conversion is suppressed, and so on.

x1) :END

Returns control from UPDATE to the monitor.

The remaining control cards refer to the control of copying decks within a file. These cards are indicated by a + or - in column 1 of each card.

a) +END m

Specifies that decks are copied from the input to the output devices until m decks (i.e. m END cards) have been copied. A missing m defaults to the value 1.

b) +EOF m

Specifies that decks are copied from the input to the output devices until m end of file marks have been copied. A missing m defaults to the value 1; the file marks are copied to the output file.

c) -EOF m

Identical to b) above except that the end of file marks are not copied to the output file.

d) +m,n

Specifies that line numbers m through n in the current deck are to be deleted during copying to the output file.

e) +m

Specifies that the current deck is to be copied to the output file up to and including line number m.

Any record read from the command device which cannot be interpreted as an UPDATE command is written to the output device at the current line position. For this reason insertion and deletion commands d) and e) must appear in the set of control commands grouped for each deck in ascending numerical line number order. Insertion records should appear immediately following the type d) or e) record to which they refer.

4.0 EXAMPLE HYBRID SIMULATION PROBLEM

As a typical example of the application of the analog/digital hybrid computer described in the foregoing sections of this report, a simulation of a simplified missile intercepting a target in two dimensional space is given in this section. Motion is assumed to occur in a vertical plane, thus requiring two translational and one rotational degrees of freedom to describe the motion of the missile. The target is assumed to be stationary.

The simplified mathematical model of the missile system is described, followed by a discussion of the partitioning of the problem into analog and digital sections. The design of the digital program and its interaction with the analog computer is described in detail and finally a set of results is included.

4.1 Mathematical Model Description

4.1.1 Vehicle Geometry, Mass and Inertia

Reference length	.584 ft
Reference area	.267 ft ²
Initial mass M_0	4.5942 slugs

Initial pitch axis moment of inertia I_{yy0} 6.9 slugs ft²

Mass variation

$$\begin{aligned} \frac{dm}{dt} &= \frac{-\text{Thrust}}{g SI} & 0 < t \leq t_{\text{burn}} \\ &= 0 & t > t_{\text{burn}} \end{aligned} \quad (4.1)$$

where g is gravity acceleration and SI is the engine specific impulse.

Inertia variation

$$\frac{d I_{yy}}{dt} = \frac{dm}{dt} \frac{I_{yy0}}{M_0} \quad (4.2)$$

4.1.2 Engine Thrust

Engine thrust is 3000 lb along the missile centerline for a duration of 3 seconds (i.e. $t_{\text{burn}} = 3$ secs).

Engine specific impulse $SI = 250$ secs.

4.1.3 Axis Systems and Reference Frames

Reference frames used in the simulation include an inertial frame XOZ with the origin O at the point on the earth surface given by the launch point; a frame fixed in the body xoz with origin at the missile CG and a frame fixed in the space referenced seeker. The inertial X axis is horizontal pointing towards the target and the inertial Z axis is vertically downwards.

The angle between the body axes and inertial axes and inertial axes is θ_E measured positively when Z is rotated towards X. The angle between the seeker frame x axis and the body x axis is θ_S are both zero all the reference frames coincide.

4.1.4 Aerodynamic Forces and Moments

Aerodynamic forces and moments are simplified by assuming linearity with angle of attack α , and control vane angle δ , and are expressed in the form of tabulated functions. The functions required are axial force coefficient C_X , normal force derivative $C_{N\alpha}$, pitching moment derivative $C_{m\alpha}$, pitch damping derivative C_{mQ} , control vane force derivative $C_{N\delta}$. All these are functions of the Mach number, and given in Table XIII. In this table all derivatives are per radian, except for C_{mQ} which is per radian/sec. Moment derivatives are referred to the missile CG.

The variation of air mass density and speed of sound with altitude is approximated by the following expression:

Table XIII Aerodynamic Functions

Mach No.	C_X	C_N	$C_{m\alpha}$	C_{mQ}	$C_{N\delta}$	$C_{m\delta}$
.6	.305	8.94	-7.16	-348	1.6125	9.57
.8	.305	14.9	-10.3	-405	2.15	9.57
1.0	.54	16.6	-9.78	-410	2.15	12.4
1.2	.755	14.75	-6.94	-406	2.29	13.37
1.4	.832	12.89	-2.29	-336	2.25	8.08

$$\frac{1}{2} \rho = .0011869 + 3.3922 \times 10^{-8} Z \text{ slugs/ft}^3 \quad (4.3)$$

$$V_s = 1116.08 + .003629Z \text{ ft/sec} \quad (4.4)$$

where ρ is the air density, V_s is the local speed of sound and Z is the missile Z coordinate which is normally negative (for positive altitude).

Aerodynamic forces in the body frame X and Z directions are:

$$F_{XA} = -qS C \quad (4.5)$$

$$F_{ZA} = -qS(C_{N\alpha}\alpha + C_{N\delta}\delta) \quad (4.6)$$

where q is the aerodynamic pressure $\frac{1}{2}\rho V^2$ and δ is the control vane angle, positive trailing edge down. The pitching moment about the CG is given by

$$M_y = qS\bar{c}(C_{m\alpha}\alpha + C_{m\delta}\delta + \frac{\bar{c}}{2V} C_{mq}Q) \quad (4.7)$$

where Q is the missile pitch rate.

4.1.5 Translation and Rotation Equations

The total force components acting on the missile are:

$$F_{XB} = F_{XA} - mg\sin\theta_E + F_{TX} \quad (4.8)$$

$$F_{ZB} = F_{ZA} + mg\cos\theta_E + F_{TZ} \quad (4.9)$$

where F_{TX} and F_{TZ} are thrust components along the missile x and z axes respectively. Relative to the inertial frame, the translational equations of motion are:

$$\ddot{X} = \frac{F_{XB} \cos\theta_E + F_{ZB} \sin\theta_E}{m} \quad (4.10)$$

$$\ddot{Z} = \frac{F_{XB} \sin\theta_E + F_{ZB} \cos\theta_E}{m} \quad (4.11)$$

$$m = M_0 - \frac{dm}{dt} t$$

and the magnitude of the missile velocity is

$$V = \sqrt{\dot{X}^2 + \dot{Z}^2} \quad (4.12)$$

Rotational equations of motion are

$$\dot{Q} = \frac{M_y}{I_{yy}} \quad (4.13)$$

$$\dot{\theta}_E = Q \quad (4.14)$$

$$I_{yy} = I_{yy0} + \frac{dI_{yy}}{dt} t \quad (4.15)$$

The aerodynamic angle of attack α may be defined as

$$\tan \alpha = \frac{W_B}{U_B} \quad (4.16)$$

where U_B and W_B are velocity components along the x and z body axes. Alternatively, the angle of attack may be introduced as a state equation given by

$$\dot{\alpha} = \frac{F_{ZB} \cos \alpha - F_{XB} \sin \alpha}{mV} + Q \quad (4.17)$$

Equation (4.17) has some advantages in a hybrid application over (4.16) and will be used in the simulation program.

4.1.6 Guidance and Control

The target is assumed to be designated by a pulsed laser operating at 20 pulses per second. A laser detector is mounted on a gimballed platform (one degree of freedom) in the nose of the missile; the detector receives the laser pulses reflected by the target and generates guidance signals from them using a form of proportional navigation.

The pulsed laser effect is equivalent to providing a sampled data seeker as illustrated in block diagram form in Figure 4.1.

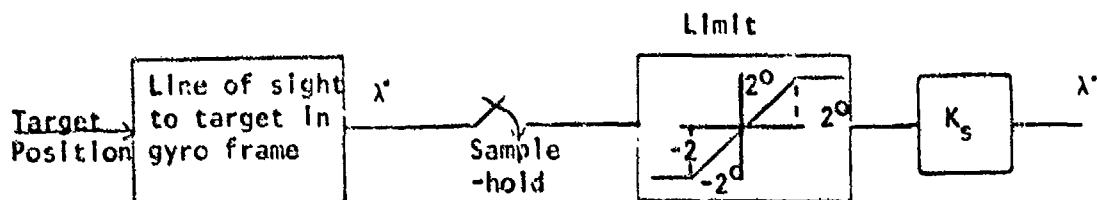


Figure 4.1 Seeker Block Diagram

The line of sight from missile to target is defined by a line of sight angle θ_{LOS} in the reference frame measured positively in the same direction as θ_E . These angles are illustrated in Figure 4.2. If the missile position is (X_m, Z_m) and the target position is (X_T, Z_T) then

$$\theta_{LOS} = \tan^{-1} \left\{ \frac{-(Z_T - Z_m)}{X_T - X_m} \right\} \quad (4.18)$$

and, in the seeker frame,

$$\lambda = \theta_{LOS} - \theta_E - \theta_s \quad (4.19)$$

λ is converted to an error signal by the limiting process illustrated in Figure 4.1 and is then multiplied by K_s , the effective gyro precession rate, to give a signal λ' which drives the gyro platform and the rear-mounted missile control fins to reduce λ to zero and to align the missile along the seeker axis. The gyro equation is simplified to

$$\dot{\theta}_s = \lambda' - \dot{\lambda} \quad (4.20)$$

and the autopilot block diagram in Laplace plane form is given in Figure 4.2. Constant terms in Figures 4.1 and 4.2 have the following values:

$$K_A = 2, K_G = .1, K_N = 60, K_D = .3703, K_S = 3$$

$$\tau_1 = .025, \tau_2 = .005, \tau_3 = .01, \tau_4 = 0,$$

$$\xi = .6, \omega_n = 60$$

The θ_s input to the autopilot in Figure 4.3 is a pitch damping loop where θ_s is differentiated and smoothed to approximate the body pitch rate $\dot{\theta}$ and the lead-lag network filters the signal to avoid exciting the missile natural frequency in pitch.

4.1.7 Target Intercept

Intercept occurs during an integration step in which either of the missile position coordinates exceeds the corresponding target coordinate. Thus, at the end of each integration step the missile position is extrapolated over the next step, using the current velocity to test whether intercept will occur, i.e. if

$$X_m + \dot{X}_m \Delta t > X_T \quad (4.21)$$

$$\text{or } Z_m + \dot{Z}_m \Delta t > Z_T \quad (4.22)$$

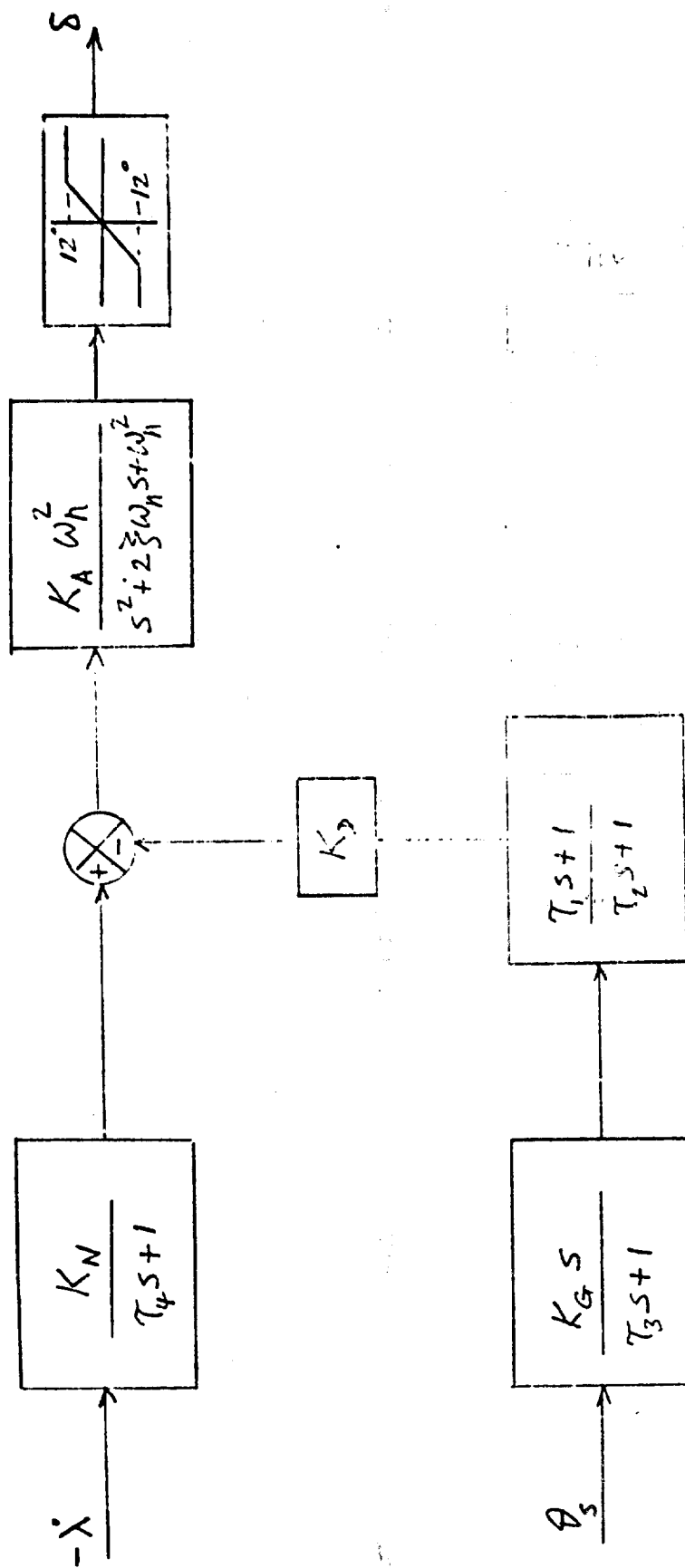


Figure 4.2 Pitch Autopilot and Actuator

Best Available Copy

If either of the above inequalities is satisfied the simulation is terminated at the current step (time T). The miss distance over the next time interval is

$$R_{\text{miss}} = \sqrt{\{x_T - x_m - \dot{x}_m (T-t)\}^2 + \{z_T - z_m - \dot{z}_m (T-t)\}^2} \quad (4.23)$$

where t is the time within the next interval T+δt. R_{miss} is a minimum at the point of closest approach of missile to target; differentiating (4.23) and equating to zero gives a value of t at intercept of

$$t = \frac{\dot{x}_m (x_T - x_m + \dot{x}_m T) + \dot{z}_m (z_T - z_m + \dot{z}_m T)}{\dot{x}_m^2 + \dot{z}_m^2} \quad (4.24)$$

where the subscript T implies evaluation at time T. Substituting t given by (4.24) into (4.23) gives the approximate miss distance, neglecting acceleration over the last partial time step.

4.2 Partitioning of the Problem

Partitioning of the mathematical model between the digital and analog computers is constrained by the requirement that the simulation execute in real-time. This constraint implies that if all the equations are implemented on the digital computer all the 12 states (4 translational, 2 rotational and 6 in the seeker, autopilot and actuator combination) must be integrated in real-time. On the Sigma 5 described in the foregoing sections this would only be possible using large integration steps and a simple integration process. However, the time constants in the autopilot are such as to require an integration step of the order of a millisecond. Thus, it is necessary to allocate part of the mathematical model for simulation on the analog computer to take advantage of its high speed parallel computation properties.

The parts of the mathematical model most appropriate for assignment to the analog computer are those containing the highest natural frequencies and short time constants. The actuator, with $\omega_n = 60$ rad/sec is an obvious candidate, as are the functional blocks in Figure 4.2 which operate on the gyro angle θ_s and line of sight rate inputs to the autopilot. The time constants of 25, 5 and 10 milliseconds there represent break frequencies of 40, 200 and 100 rad/sec respectively.

The seeker sample and hold process, however, is suitable for implementation on the digital computer since this has a period of 50 milliseconds (20 pulses/second) and a DAC output is ideally suited to providing the

sampled values. An objection to this can arise in the sampling of the gyro angle θ_s which will be small over most of the flight but must be scaled to the maximum limits on θ_s (normally around 20°). To overcome this disadvantage the gyro is implemented on both digital and analog computers. On the analog computer θ_s is calculated according to equation (4.21) and used as an input to the autopilot. On the digital computer the gyro angle is calculated relative to inertial axes to give

$$\theta_{SI} = \Delta t \sum \lambda' \quad (4.25)$$

where Δt is the 50 millisecond sample-hold interval and λ is the line of sight sample converted to a rate. From equation (4.25) the line of sight angle in the seeker frame becomes

$$\lambda = \theta_{LOS} - \theta_{SI} \quad (4.26)$$

The missile translation equations are assigned to the digital computer because of the low frequencies contained in the velocity and displacements and because of the need for a high dynamic range required to give an accurate miss distance at the end of a flight. There remains to be assigned then the missile rotational equations for Q and θ_E . Q will respond approximately as a damped second order system with a natural frequency given by (see reference 16):

$$\omega_Q = \sqrt{\frac{mg\bar{c}}{C_{L_0} I_{yy}} \left(-C_{m\lambda} - \frac{g\bar{c}}{2V^2} \frac{C_{N\lambda} C_{mq}}{C_{L_0}} \right)} \quad (4.27)$$

$$\text{where } C_{L_0} = \frac{mg}{\frac{1}{2}\rho V^2 S} \quad (4.28)$$

Assuming conditions of $V = 1000$ ft/sec at an altitude of 1000 ft and substituting the appropriate values after engine burnout into equations (4.27) and (4.28) gives ω_Q of 17.1 rad/sec which is just less than 3 Hz. A simple integration process using one derivative evaluation per step would require an integration interval of less than 10 milliseconds which would probably permit Q and θ_E to be assigned to the digital computer in this simple 3 degree of freedom problem. For a full dof case, however, the time requirement would prevent this, so to make the present problem a realistic example the equations for Q , θ_E and θ_s are assigned to the analog computer. In addition, λ is evaluated according to equation (4.17) on the analog computer and λ , δ and Q are used as analog DAC multipliers in the evaluation of the aerodynamic moment input to equation (4.13) with the aerodynamic derivatives $C_{m\lambda}$, $C_{m\delta}$ and C_{mq} obtained by table look-up on the digital computer. Evaluating M_y in this manner reduces the "staircase" effect of the DAC outputs.

Four translation states are integrated on the digital computer and all body force and moment terms are evaluated there. The digital integration method is a 2 point Adams-Bashforth process with projection of derivatives from the previous time frame, which may be represented by the following equations for the X inertial direction:

$$\dot{x}_{C_{i+1}} = x_{C_i} + \frac{h}{2} (\ddot{x}_i + \ddot{x}_{i-1}) \quad (4.29)$$

$$\dot{x}_i = x_{C_{i+1}} + \frac{h}{2} (3\ddot{x}_i - \ddot{x}_{i-1}) \quad (4.30)$$

$$x_{i+1} = x_i + \frac{h}{2} (\dot{x}_{C_{i+1}} + \dot{x}_i) \quad (4.31)$$

where the step length is h, i refers to the current time and the suffix C implies a "corrected" value.

Termination of the simulation and the calculation of miss distance employs equations (4.21) through (4.23).

A block diagram of the partitioned problem is given in Figure 4.3.

4.3 Design of the Digital Program

The design of the digital section of the hybrid program centers around the timing and interrupt sequencing. The real-time section of the simulation is driven by a series of clock interrupts which synchronize the program with real-time and provide control of the digital integration process. A convenient reference cycle within the simulation is the 50 millisecond seeker sample-hold interval and the digital integration interval is chosen to be an integral multiple of this overall time frame.

A digital integration step of 25 milliseconds is adequate for the translation equations which is also the interval for output of the aerodynamic moment update terms and for the output to the angle of attack equation. A diagram illustrating time usage during the real-time section of the program is given in Figure 4.4. All the program sections are written in FORTRAN.

In addition to the real-time section the program contains initialization, pre-real time and post-real time sections. The initialization section is the Main Program and is executed only once at the time the program is loaded into the foreground region of memory. Initialization consists of connecting the Sigma 5 external interrupts to various routines in the program and defining the hybrid I/O patterns. Thereafter, for all subsequent simulation runs the program is driven by the external interrupts.

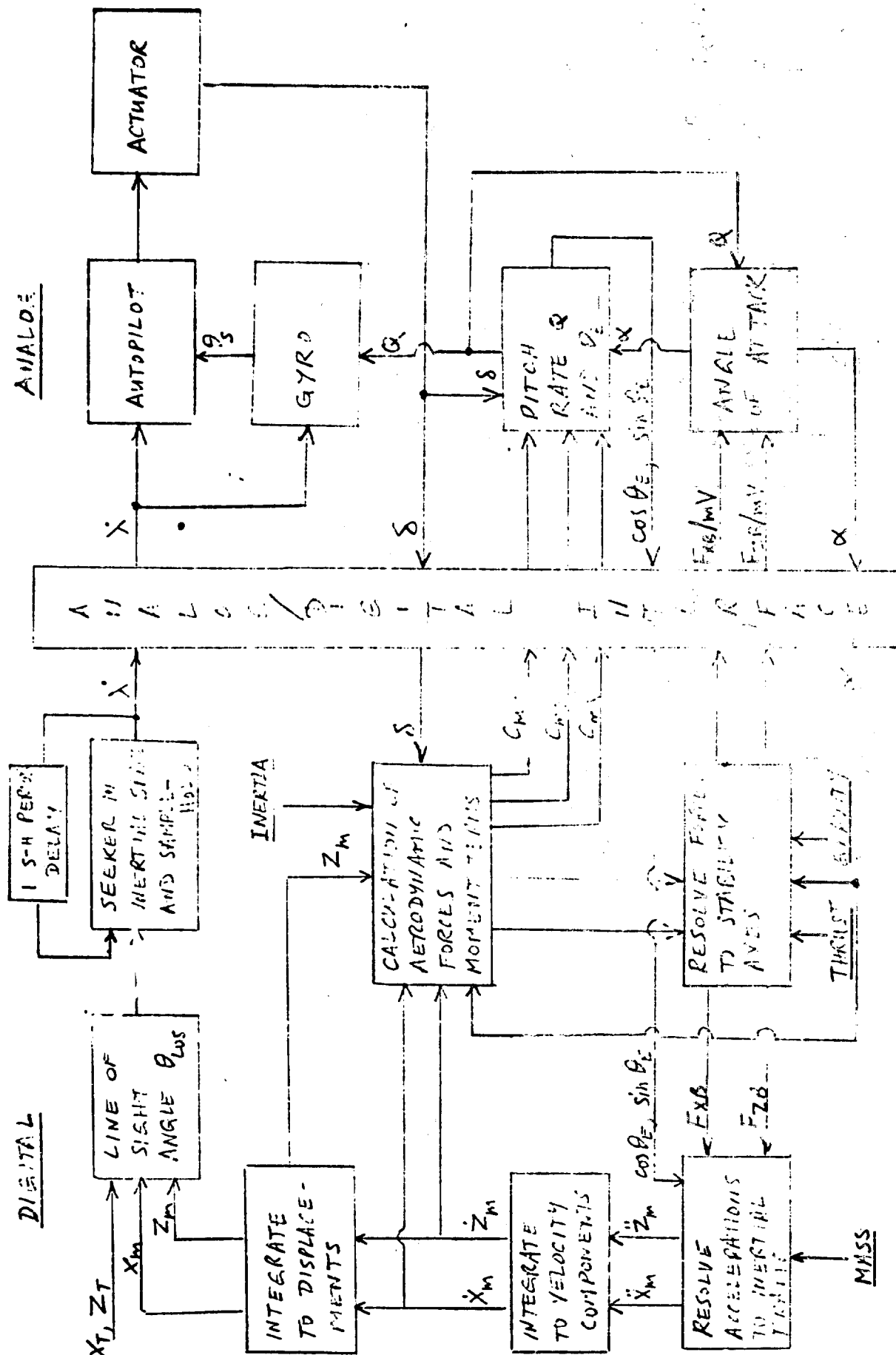
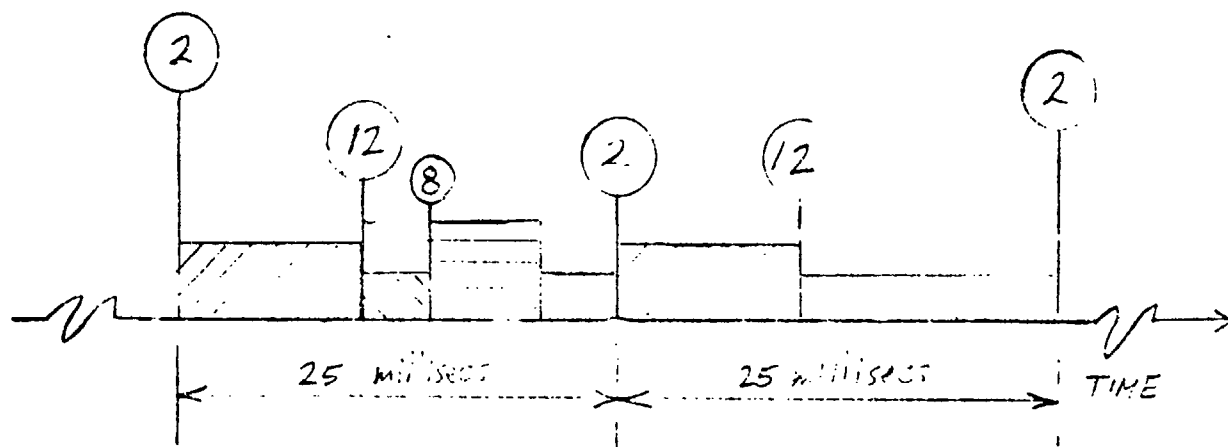


Figure 4.3 Partitioned Problem



2

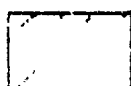
Clock Interrupt followed by ADC Input and digital Integration

12

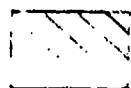
Software trigger of subroutine SCHEDULE

8

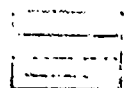
Software trigger of subroutine SEEKER



Digital Integration, calculation and output of DAC values (except sample-hold)



Routine SCHEDULE (plus time unused)



Seeker sample-hold calculation and DAC output

Figure 4.4 Frame Time Usage

The pre-real time section consists of subroutine STARTRUN which is connected to interrupt 14. This subroutine initializes various digital and analog variables and reads various run control parameters from a RAD file. Run control parameters are read from file SAMPLE in the D7 RAD area and are punched as for a NAMELIST input (see references 10 and 13) as given in Table XIV. Variables which are included in Table X V, but which are not read from file SAMPLE have the default values indicated in the table. At the end of subroutine STARTRUN an initial integration step is taken by triggering the appropriate interrupt, the real-time clock is then set running and the analog computer is set to the compute mode which thus commences the simulation.

Subroutine CLOCK responds to the real-time clock interrupts (no.2) at 25 millisecond intervals, reads analog inputs through ADCs and performs the calculation required for integration and analog output. At its end it triggers subroutine SCHEDULE (interrupt 12) to determine whether the run is to terminate or whether a seeker sample-hold update is required. If the latter is required, interrupt 8 is triggered to activate subroutine SEEKER to perform the sample-hold operation on the line of sight rate. Subroutine SCHEDULE also writes to magnetic tape a record of selected variables at time intervals determined by the input data. It should be noted that all event times within the real-time section are converted in routine STARTRUN to an integral number of 25 millisecond time frames. In this way, the current time is maintained by incrementing a counter by 1 each time subroutine CLOCK is entered.

Finally, the last part of subroutine SCHEDULE performs the post-realtime operations of reading back the magnetic tape, printing the results and calculating and printing the intercept miss distance.

A block diagram of the interlinking of the subroutines which comprise the program is given in Figure 4.5. Subroutine ABORT is triggered from the analog console and is used to terminate a simulation run externally; ABORT is connected to interrupt number 3. External interrupt 13 is connected to an entry point in the hybrid library named FG\$RLS which has the effect, when triggered, of releasing foreground program OV (which will usually be the name under which this program is executed) from the system.

A listing of each routine in the program is given in Appendix A and in Figure 4.6 a flowchart of the program execution is outlined to indicate the flow of control during program execution. Table XV contains the list of ADC's and DAC's used in the program and their variable definitions.

TABLE XIV NAMELIST Input to the Digital Program

FORTRAN NAME	SYMBOL	DEFINITION	DEFAULT VALUE
TIME	αt	Initial value of independent variable	0
DT	δt	Integration and Clock Interval	.025 secs
TSTOP	-	Termination time (unless intercept occurs)	25 secs
TACQ	-	Time of target acquisition	0
TPSTRT	-	Time to start tape output	0
TAPETIME	-	Time between tape output records	1 sec
IPULSE	-	Seeker sample-hold rate	20/second
THEMAX	-	Seeker S-H input angle limit	2 deg
PRMAX	-	Seeker S-H output angle limit	2 deg
XT	X_T	Target initial X position	8000 ft
ZT	Z_T	Target initial Z position	0
X	X_m	Missile initial X position	0
Z	Z_m	Missile initial Z position	-500 ft
VELOCITY	V	Missile initial total velocity	0
ALFASCAL	-	α scale factor	20°
DELTSCAL	-	δ scale factor	20°
THESCALE	-	θ_E scale factor	-
BURNOUT	t_{BO}	Time of engine burnout	3 secs
STMASS	M_0	Initial mass of missile	4.6 slugs
ENDMASS	m	Final mass (at burnout)	3.9 slugs
STIYY	I_{yy0}	Initial moment of inertia	6.1 s.ft ²
ENDIYY	I_{yy}	Final moment of inertia	4.9 s.ft ²
S	S	Aerodynamic reference area	.267 ft ²
CBAR	$\bar{c}$	Aerodynamic reference length	.584 ft
QIC	-	Initial value of Q	0
THETAIC	-	Initial value of θ_E	0
THRUST	F_{TX}	Thrust	1500 lb.
GAIN	K_S	Gyro precession rate gain	10/sec

Figure 4.5 Subroutine Linkage in the Program

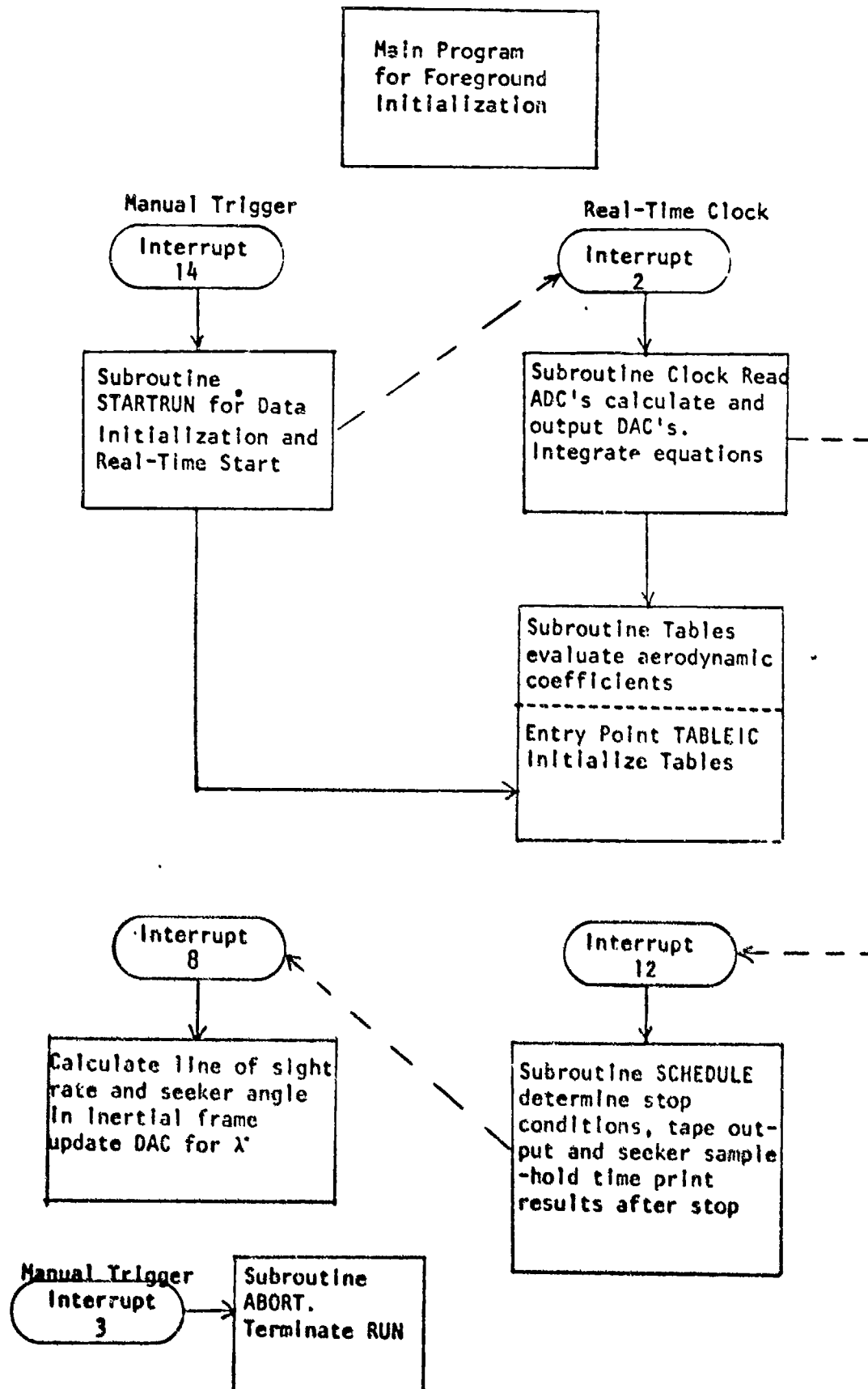


TABLE XV DAC and ADC Usage

DAC or DC	Variable Definition	Scale Factor	Analog Multiplier
DAC 02	$\frac{qS\bar{c}^2}{2I_{yy}V} C_{mQ}$ Aerodynamic pitch damping	932.8	Q
DAC 03	$\frac{qS\bar{c}}{I_{yy}} C_{m\dot{\alpha}}$ Pitch moment derivative	373.1	$\dot{\alpha}$
DAC 05	$\frac{-qS\bar{c}}{I_{yy}} C_{m\dot{\delta}}$ Vane moment derivative	3.73	$\dot{\delta}$
DAC 10	$\frac{-F_{XB}}{mV}$ Wind axis rate term	0.2	$\dot{\alpha}$
DAC 12	$\frac{F_{ZB}}{mV}$ Wind axis rate term	0.1	-1.0
DAC 22	$-\theta_{EIC}$ Euler angle initial value	180°	1.0
DAC 26	$-Q_{IC}$ Initial pitch rate	150°/sec	1.0
DAC 28	$\dot{\chi}/K_s$ Line of sight rate	2°	1.0
ADC 00	$\cos\theta_E$	1.0	-
ADC 01	$\sin\theta_E$	1.0	-
ADC 06	$\dot{\alpha}$ Angle of attack	.35	-
ADC 08	$\dot{\delta}$ Vane angle	.35	-

Figure 4.6 Program Overall Flow Chart

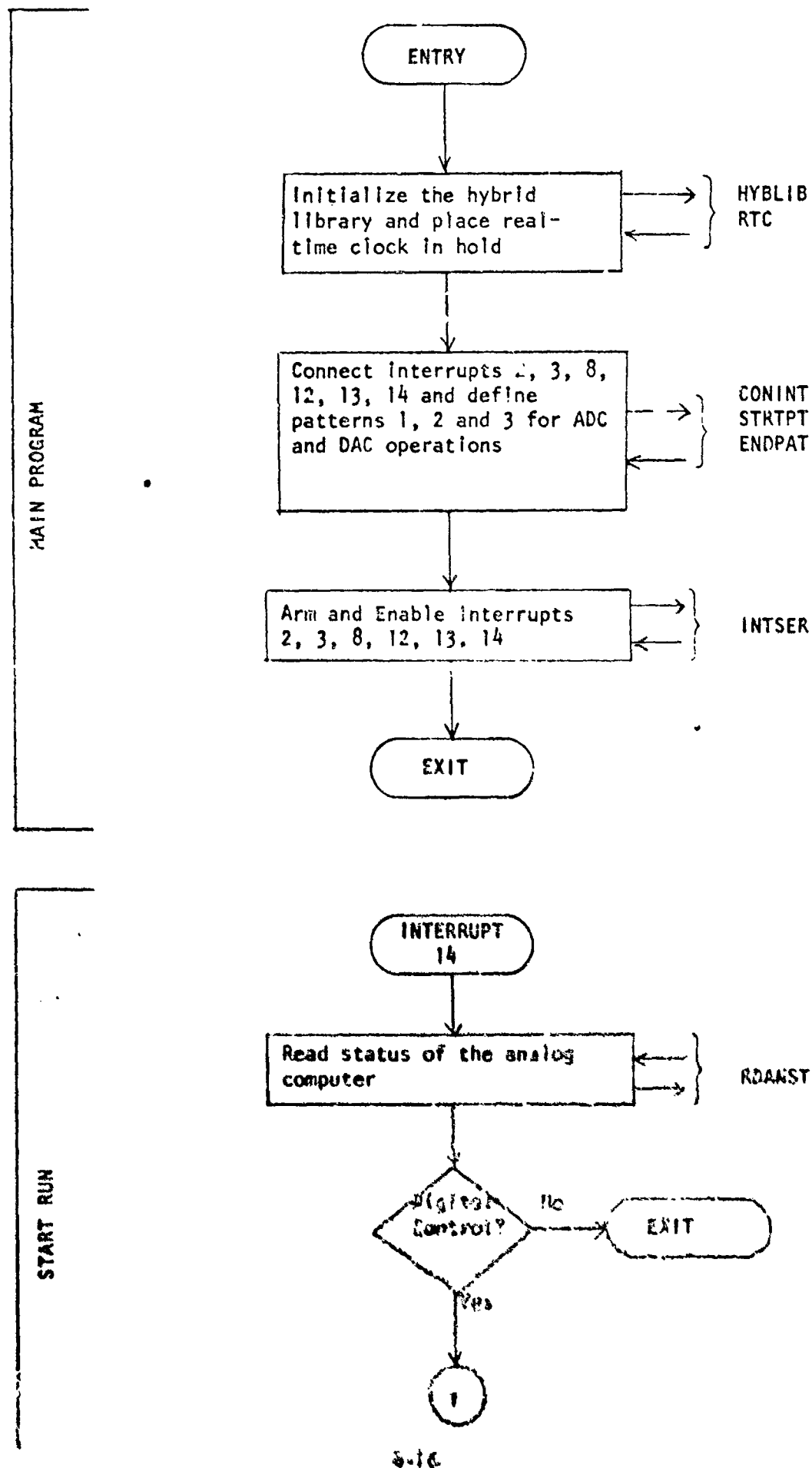


Figure 4.6 Cont.

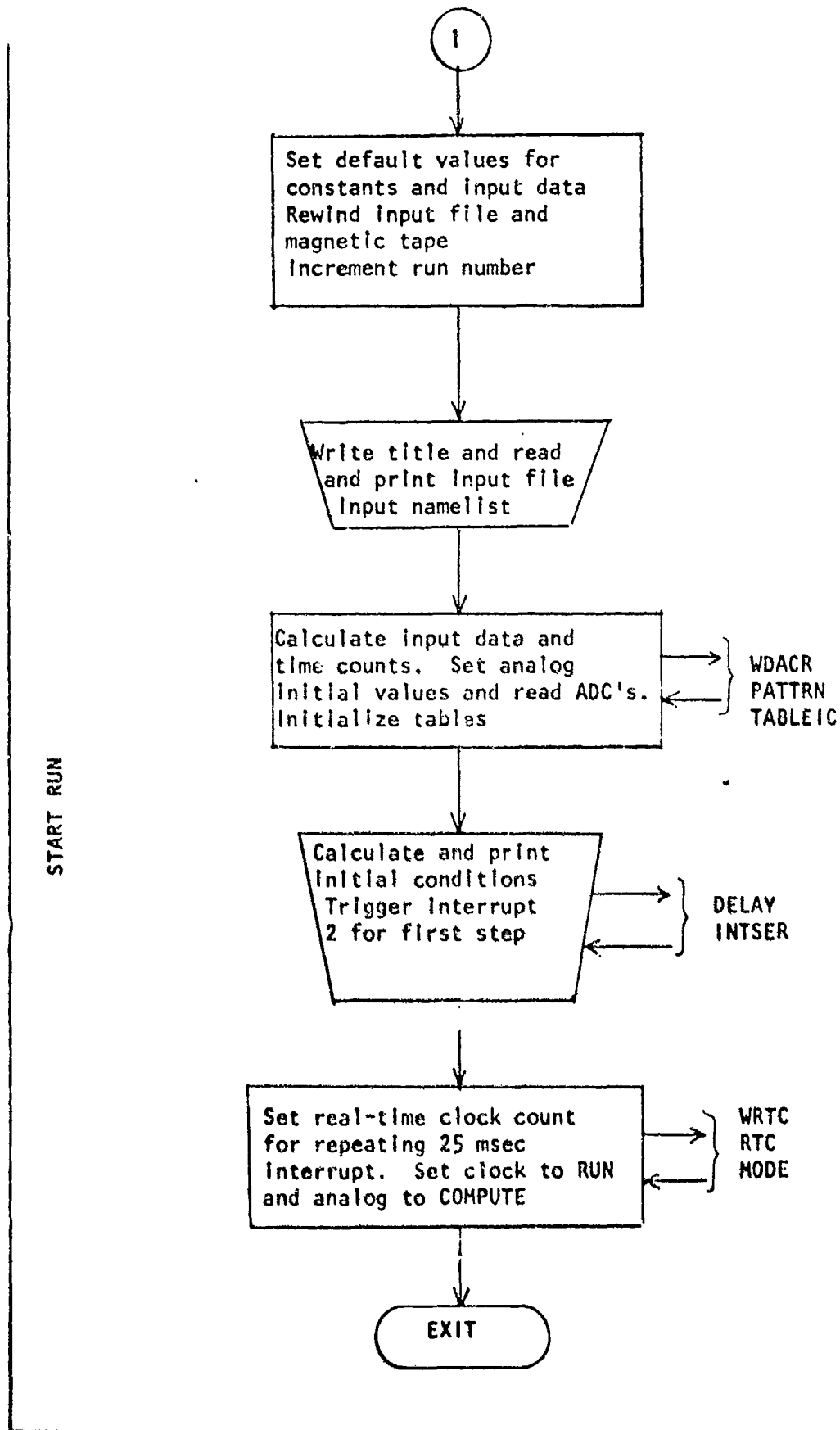


Figure 4.6 Cont.

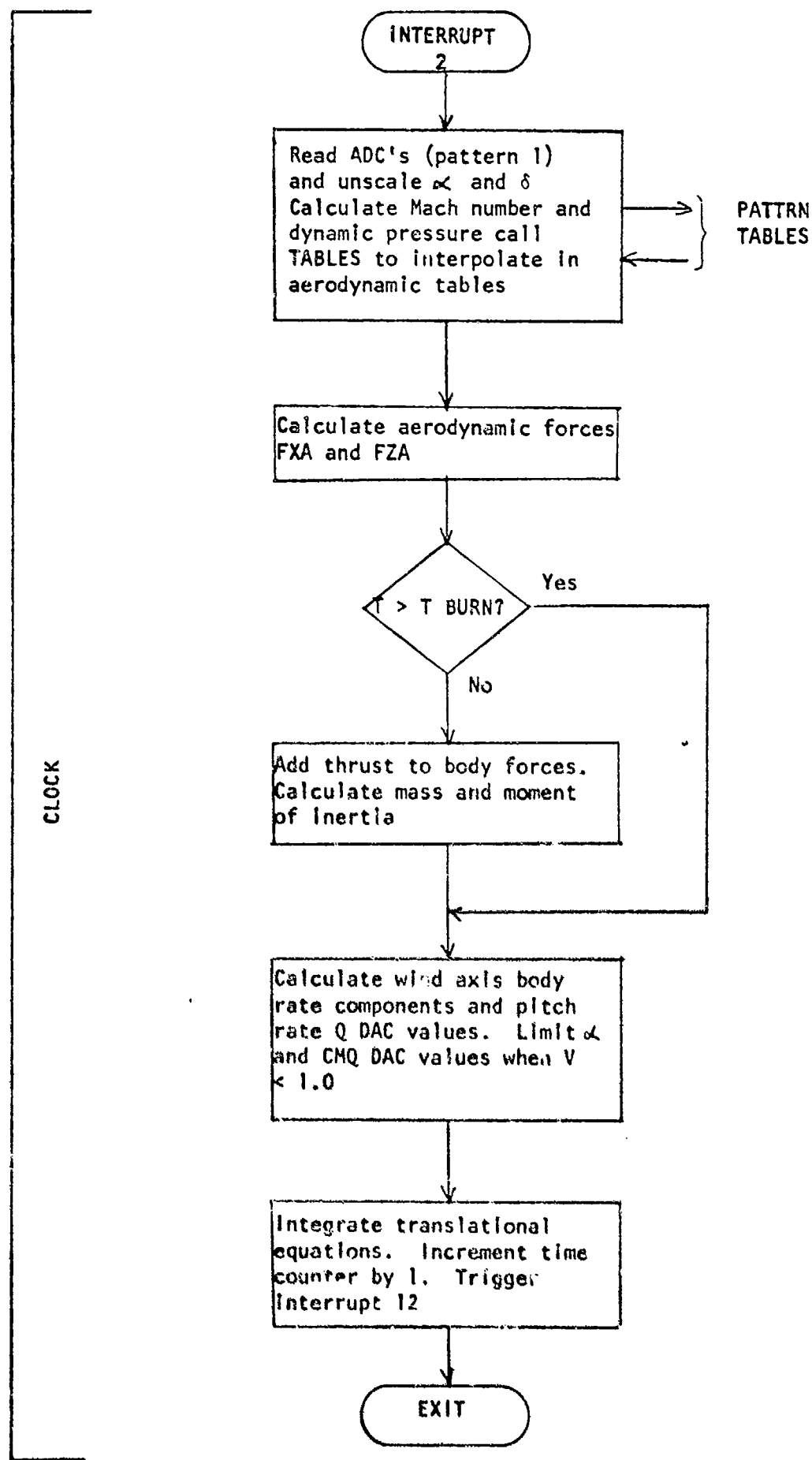


Figure 4.6 Cont.

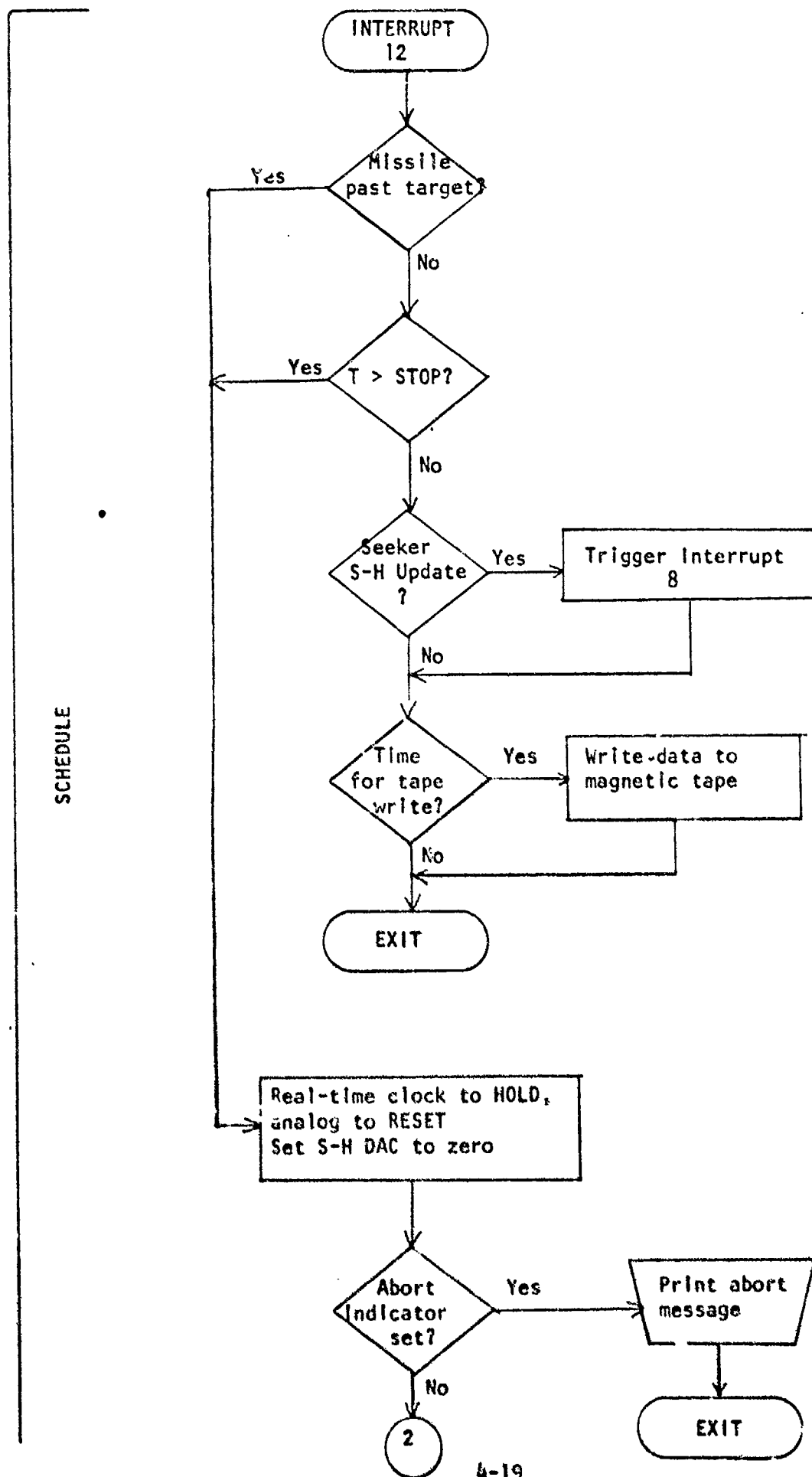
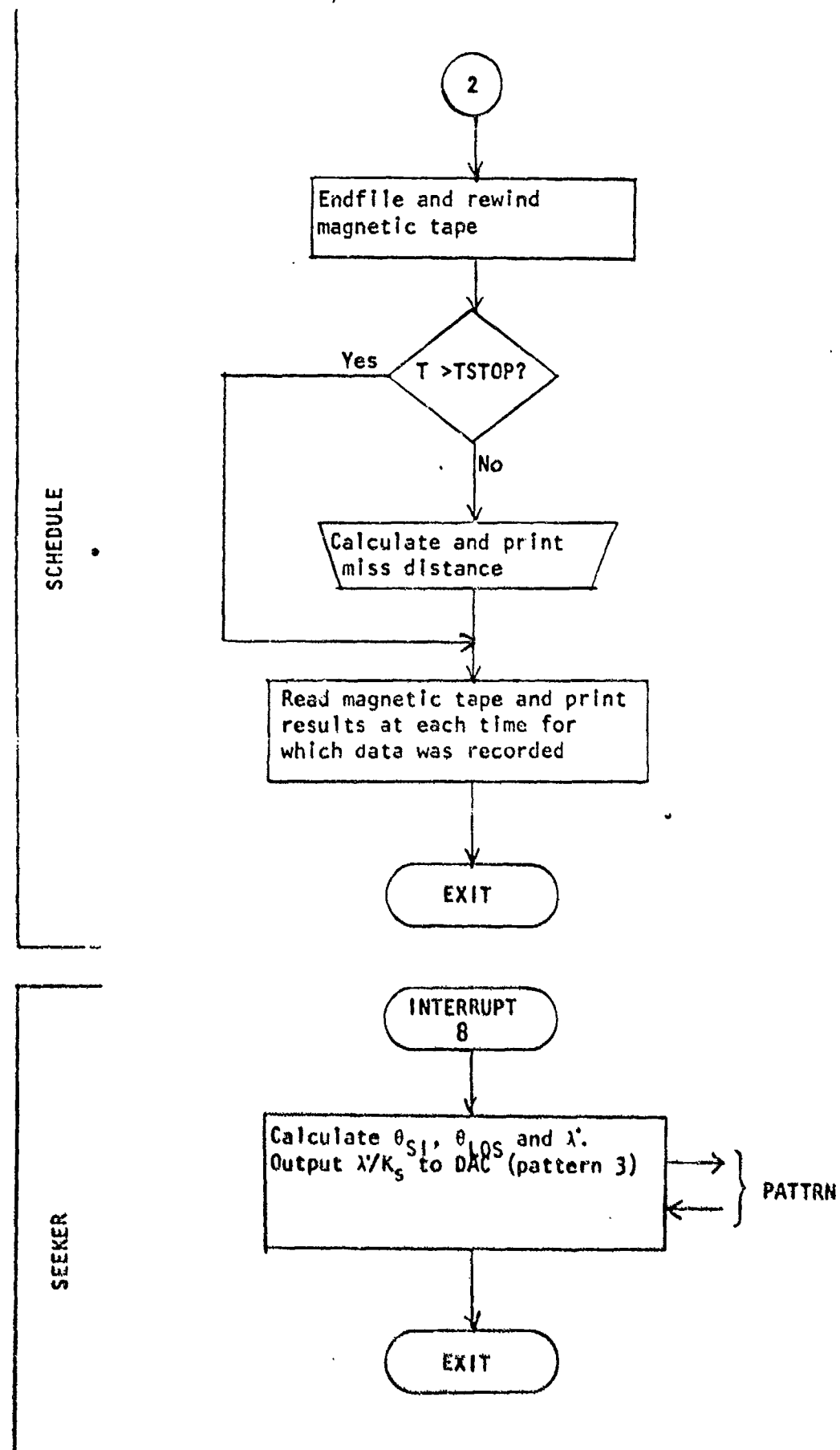


Figure 4.6 Cont.



4.4 Analog Programming and Patching Diagrams

This section contains the analog patching diagrams and potentiometer settings for the mathematical model sections assigned to the analog computer. Component numbers refer to the CI-5000 analog computer.

TABLE XVI Potentiometer Settings

Potentiometer Number	Setting	Potentiometer Number	Setting
001	.2000	048	.5250
003	.2151	049	.5250
022	.8571	050	.6852
025	.4000	051	.7194
027	.1237	068	.2674
028	.2000	078	.6000
029	.5200	105	.1874
031	.8571	107	.9550
036	.1000	115	.6272
047	.9999	129	.3096
		137	.5714

4.5 Results

The simulation described in the preceding sections was checked out and run with the following conditions:

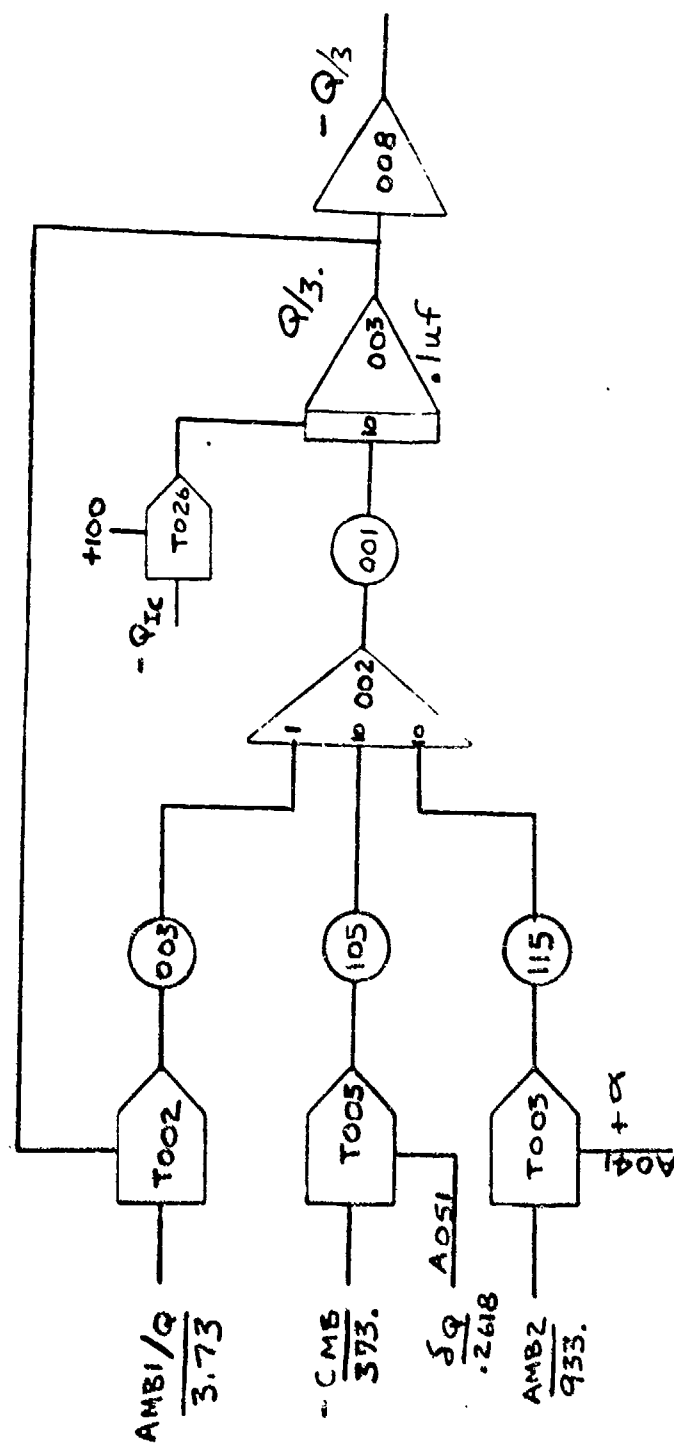
Target stationary at $X_T = 8000$ ft, $Z_T = 0$.

Missile Initial position $X_m = 0$, $Z_m = -500$ ft.

Missile Initial velocity 0, Initial $\theta_E = 0$, Initial $Q = 0$.

Thrust = 3000 lb, burnout at 3 seconds, and the remaining data given by the default values in Table XIV.

Results printed by the digital program are shown in Table XVII and selected analog variables as functions of time, reproduced from strip chart recordings are given in Figures 4.14 and 4.15.



$$\dot{Q} = AMB1\phi Q + Q - CMB\delta Q + AMB2\alpha$$

Figure 4.7 Rotational Pitch Rate Analog Diagram

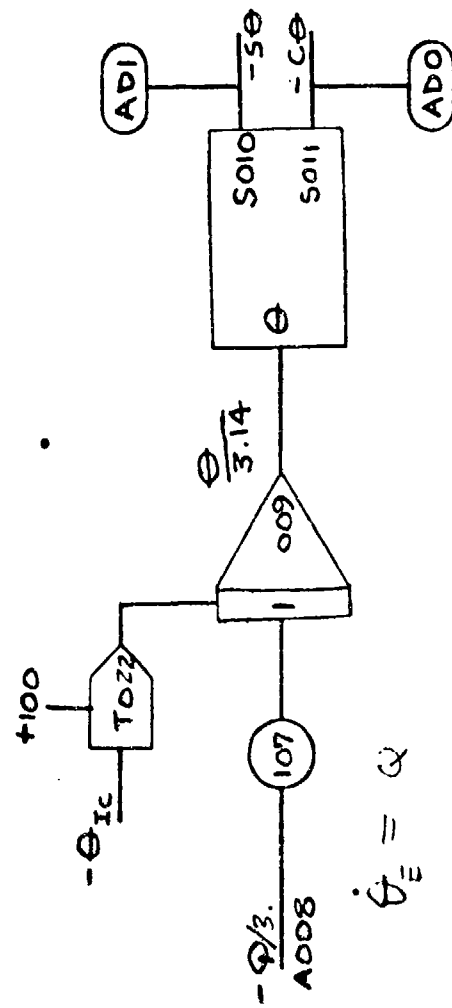
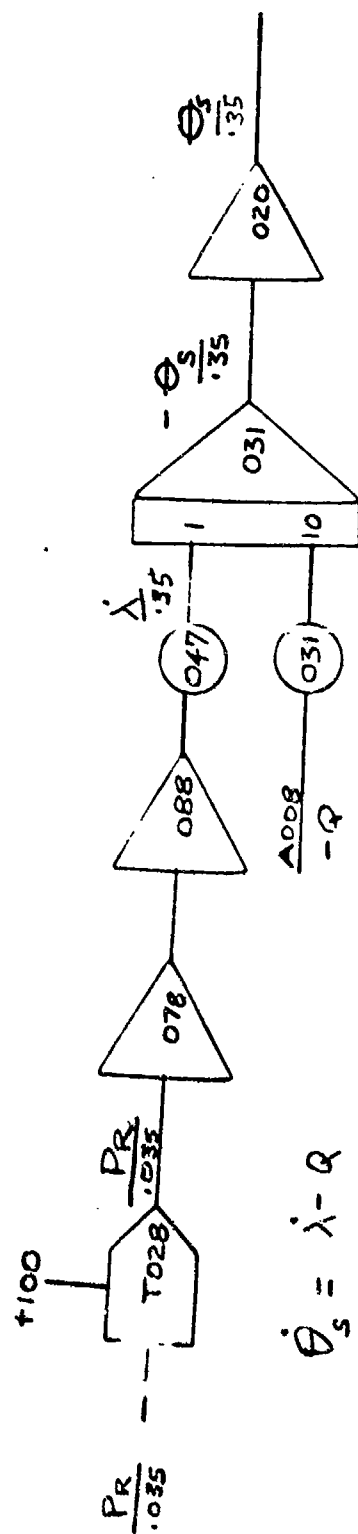
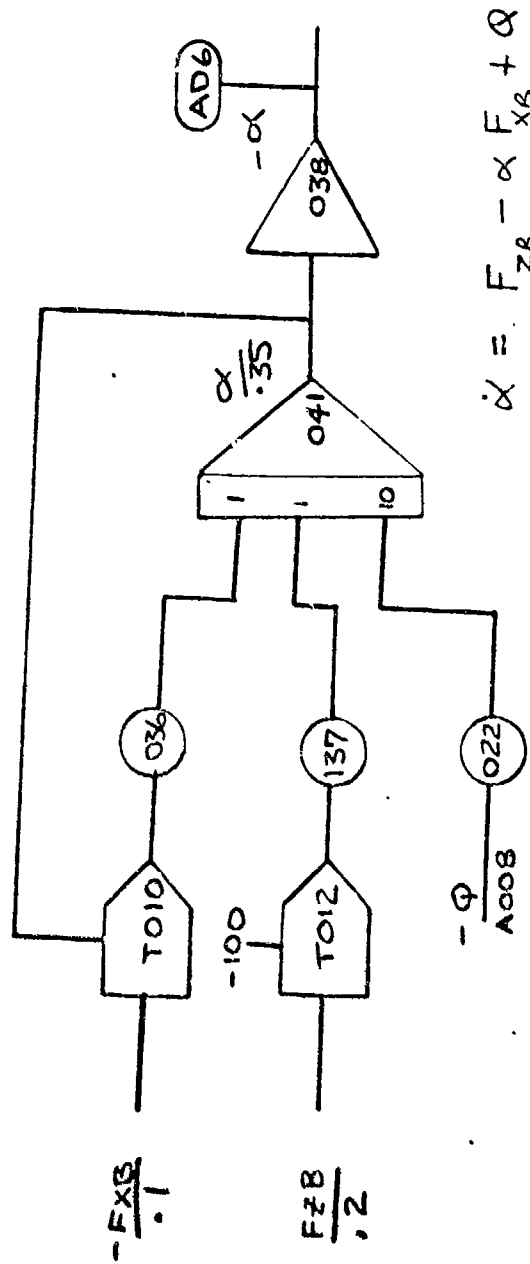


Figure 4.8 Pitch Euler Angle



$$\dot{\theta}_s = \dot{\lambda} - Q$$

Figure 4.9 Gyro Angle



$$\dot{\alpha} = F_{zB} - \alpha F_{xB} + Q$$

Figure 4.10 Angle of Attack

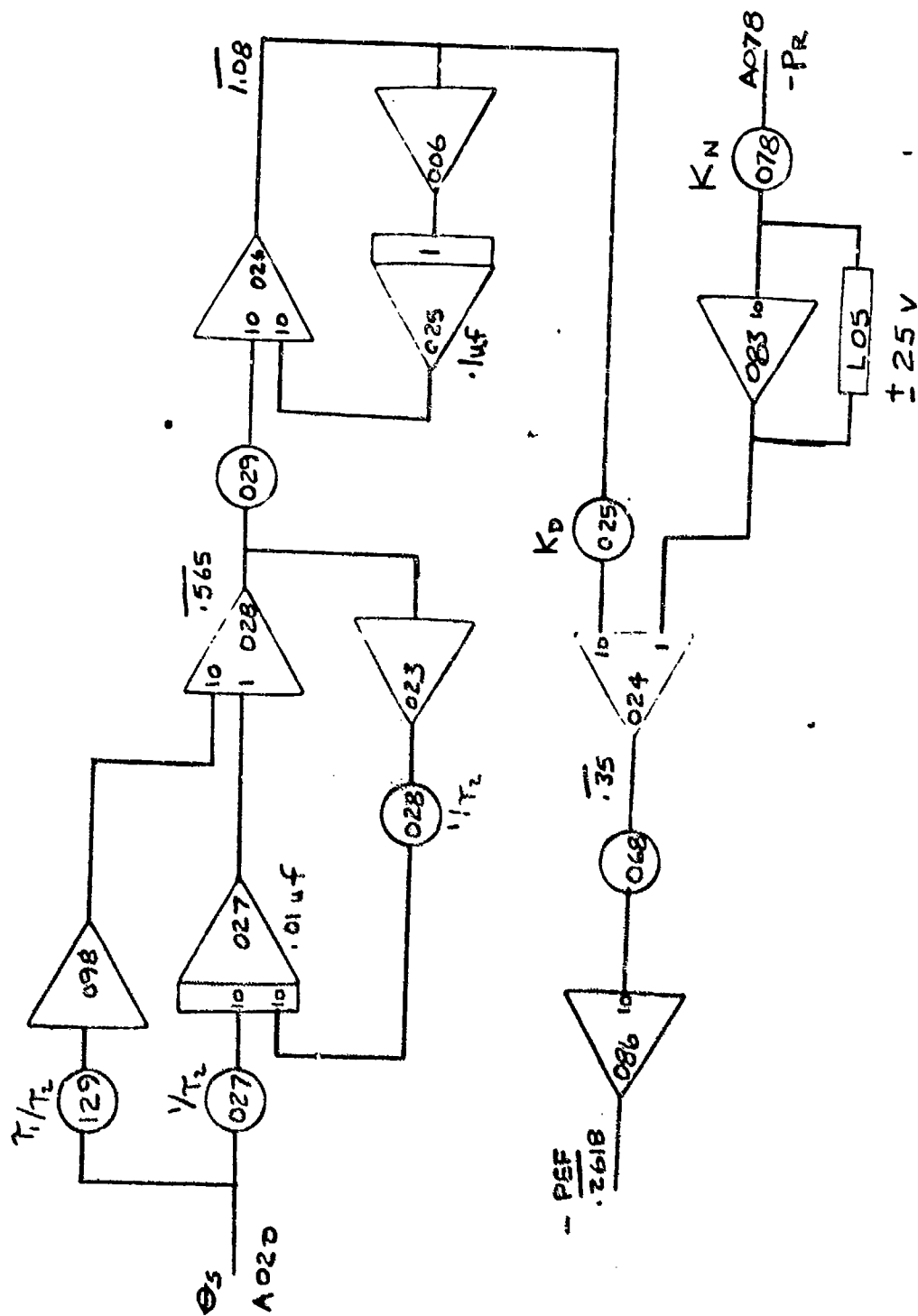
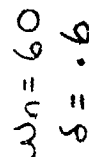


Figure 4.11 Pitch Autopilot



$$\ddot{E}_0 = -72 \dot{E}_0 - 3600 (E_0 - E_{11})$$

4.12 Pitch Actuator

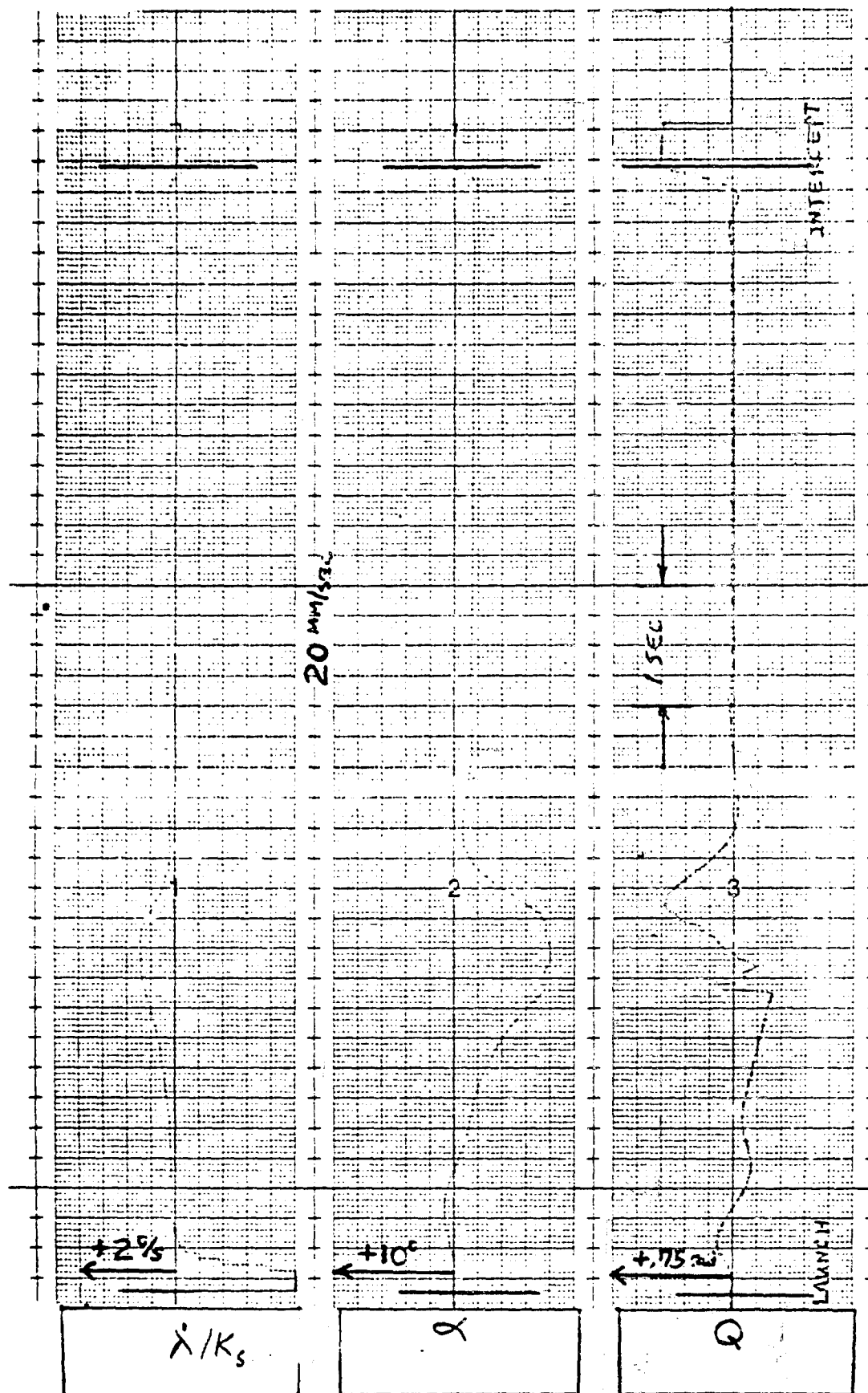


Figure 4.13 Analog Variables as Functions of Time (1)

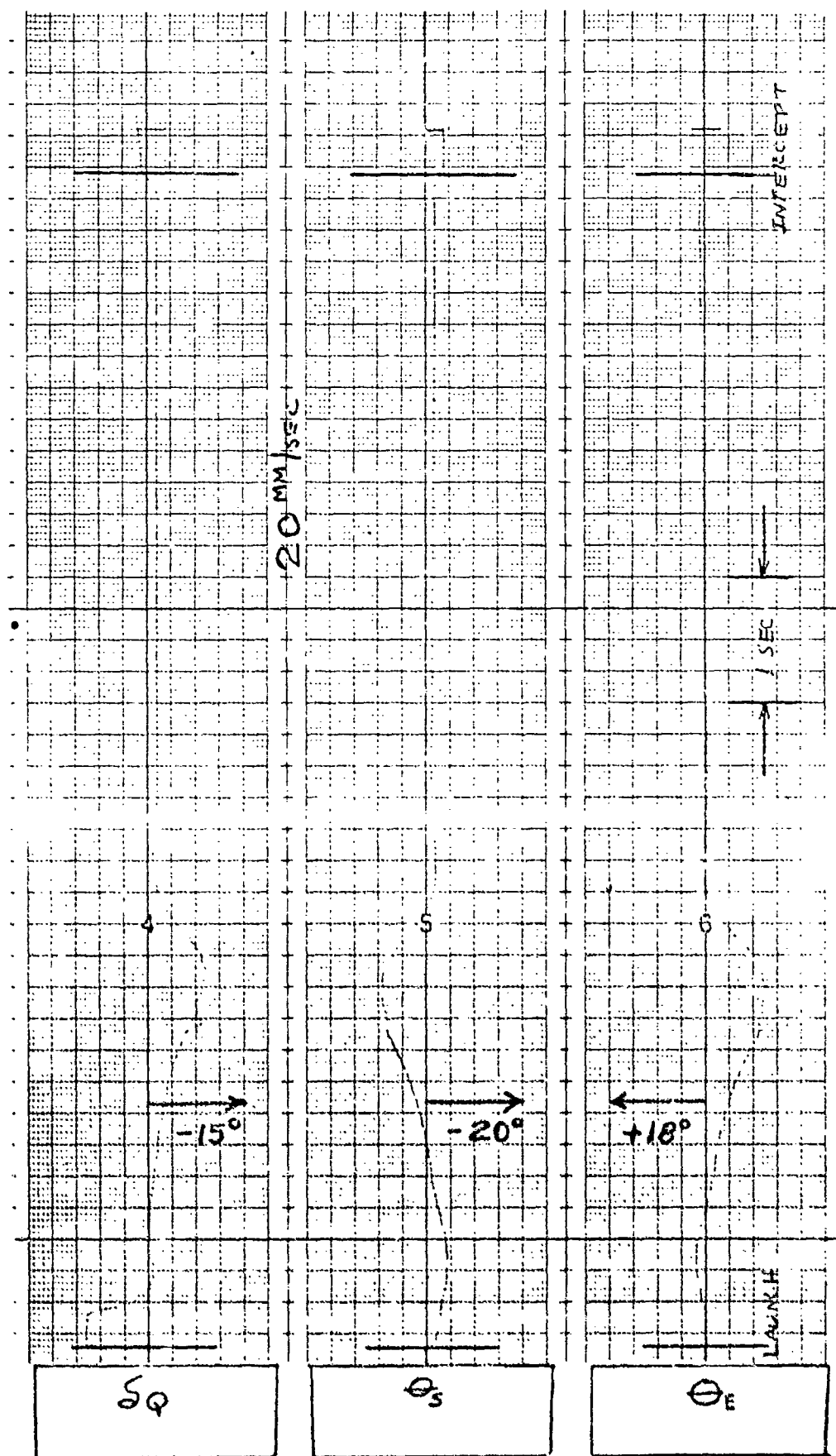


Figure 4.14 Analog Variables as Functions of Time (2)

1999

END
25. AS FRAME TIME. STOP AT 25. SEC. ACQUISITION AT 0. SEC. SERVER UPDATES 20 TIMES PER SECOND. FT.
TRACK 0000. LB. LIFT BURST AT 0. SEC. REFERENCE AREA = .967 SQ. FT. REFERENCE LENGTH = .588. FT.
TRACK 0001. LIFTED POSITION, VELOCITY AND ALTITUDE: (0.0 SEC.) FT. (0.0 FT/SEC AND 0.0 DEGREES.
0002. LIFTED AT 7.0000. 0.0 FEET.

REPT AT TIME • 3.11.6 • 9.42 SECONDS • 10.11.6

2 ACC.
TLETA
DACS
PASS
MACH
CMALP

Z ACC.
THETA
DACS
PASS
PACH
CPALP
3.2794E C1
-3.4997E-03
1.0000E C0
4.3667E C0
3.0095E-C1
-7.1600E C0

Z ACC.
THETA
DACS
PASS
PACH
CPALP

Z ACC
THETA
DACS
PASS
PACW
CMALP

Z ACC
THEYA
DACS
PASS
MACH
CUAL



CDFLT	1.133E-01	THETAS	-2.608E-00	THETALS	-2.509E-00	PR	9.5921E-03		
TIME	5.000E-00	X ACC.	2.7783E-01	X VEL.	1.051E-03	X	3.6419E-03	Z ACC.	-5.7354E-00
Z VEL.	5.300E-01	Z	-1.879E-02	ALPHA	-1.464E-00	DELTA	-1.5230E-00	THETA	-3.4785E-01
DAC1	-1.012E-01	DAC2	1.2701E-02	DAC3	-1.274E-00	DAC4	-2.6449E-02	DACS	-3.6299E-01
FVA	1.445E-03	FVA	-1.445E-02	FVB	-1.445E-02	F7B	-1.445E-02	MACH	3.8941E-00
IVV	1.445E-03	CLF	1.445E-02	VELOCITY	1.445E-02	SS	1.115E-03	CHALP	9.2033E-01
RFL	2.341E-03	CV	4.444E-01	CHALP	1.445E-01	CDELTA	2.1500E-00		-1.0030E-01
CDFLT	1.133E-01	THETAS	-2.608E-00	THETALS	-2.509E-00	PR	1.2442E-02		
TIME	5.000E-00	X ACC.	2.7783E-01	X VEL.	1.051E-03	X	4.5574E-03	Z ACC.	-5.7354E-00
Z VEL.	5.300E-01	Z	-1.879E-02	ALPHA	-1.464E-00	DELTA	-1.5230E-00	THETA	-3.4785E-01
DAC1	-1.012E-01	DAC2	1.2701E-02	DAC3	-1.274E-00	DAC4	-2.6449E-02	DACS	-3.6299E-01
FVA	1.445E-03	FVA	-1.445E-02	FVB	-1.445E-02	F7B	-1.445E-02	MACH	3.8941E-00
IVV	1.445E-03	CLF	1.445E-02	VELOCITY	1.445E-02	SS	1.115E-03	CHALP	9.2033E-01
RFL	2.341E-03	CV	4.444E-01	CHALP	1.445E-01	CDELTA	2.1500E-00		-1.0030E-01
CDFLT	1.133E-01	THETAS	-2.608E-00	THETALS	-2.509E-00	PR	1.2442E-02		
TIME	5.000E-00	X ACC.	2.7783E-01	X VEL.	1.051E-03	X	5.6560E-03	Z ACC.	-5.7354E-00
Z VEL.	5.300E-01	Z	-1.879E-02	ALPHA	-1.464E-00	DELTA	-1.5230E-00	THETA	-3.4785E-01
DAC1	-1.012E-01	DAC2	1.2701E-02	DAC3	-1.274E-00	DAC4	-2.6449E-02	DACS	-3.6299E-01
FVA	1.445E-03	FVA	-1.445E-02	FVB	-1.445E-02	F7B	-1.445E-02	MACH	3.8941E-00
IVV	1.445E-03	CLF	1.445E-02	VELOCITY	1.445E-02	SS	1.115E-03	CHALP	9.2033E-01
RFL	2.341E-03	CV	4.444E-01	CHALP	1.445E-01	CDELTA	2.1500E-00		-1.0030E-01
CDFLT	1.133E-01	THETAS	-2.608E-00	THETALS	-2.509E-00	PR	1.2442E-02		
TIME	5.000E-00	X ACC.	2.7783E-01	X VEL.	1.051E-03	X	6.6332E-03	Z ACC.	-5.7354E-00
Z VEL.	5.300E-01	Z	-1.879E-02	ALPHA	-1.464E-00	DELTA	-1.5230E-00	THETA	-3.4785E-01
DAC1	-1.012E-01	DAC2	1.2701E-02	DAC3	-1.274E-00	DAC4	-2.6449E-02	DACS	-3.6299E-01
FVA	1.445E-03	FVA	-1.445E-02	FVB	-1.445E-02	F7B	-1.445E-02	MACH	3.8941E-00
IVV	1.445E-03	CLF	1.445E-02	VELOCITY	1.445E-02	SS	1.115E-03	CHALP	9.2033E-01
RFL	2.341E-03	CV	4.444E-01	CHALP	1.445E-01	CDELTA	2.1500E-00		-1.0030E-01
CDFLT	1.133E-01	THETAS	-2.608E-00	THETALS	-2.509E-00	PR	1.2442E-02		
TIME	5.000E-00	X ACC.	2.7783E-01	X VEL.	1.051E-03	X	7.6104E-03	Z ACC.	-5.7354E-00
Z VEL.	5.300E-01	Z	-1.879E-02	ALPHA	-1.464E-00	DELTA	-1.5230E-00	THETA	-3.4785E-01
DAC1	-1.012E-01	DAC2	1.2701E-02	DAC3	-1.274E-00	DAC4	-2.6449E-02	DACS	-3.6299E-01
FVA	1.445E-03	FVA	-1.445E-02	FVB	-1.445E-02	F7B	-1.445E-02	MACH	3.8941E-00
IVV	1.445E-03	CLF	1.445E-02	VELOCITY	1.445E-02	SS	1.115E-03	CHALP	9.2033E-01
RFL	2.341E-03	CV	4.444E-01	CHALP	1.445E-01	CDELTA	2.1500E-00		-1.0030E-01
CDFLT	1.133E-01	THETAS	-2.608E-00	THETALS	-2.509E-00	PR	1.2442E-02		
TIME	5.000E-00	X ACC.	2.7783E-01	X VEL.	1.051E-03	X	8.5876E-03	Z ACC.	-5.7354E-00
Z VEL.	5.300E-01	Z	-1.879E-02	ALPHA	-1.464E-00	DELTA	-1.5230E-00	THETA	-3.4785E-01
DAC1	-1.012E-01	DAC2	1.2701E-02	DAC3	-1.274E-00	DAC4	-2.6449E-02	DACS	-3.6299E-01
FVA	1.445E-03	FVA	-1.445E-02	FVB	-1.445E-02	F7B	-1.445E-02	MACH	3.8941E-00
IVV	1.445E-03	CLF	1.445E-02	VELOCITY	1.445E-02	SS	1.115E-03	CHALP	9.2033E-01
RFL	2.341E-03	CV	4.444E-01	CHALP	1.445E-01	CDELTA	2.1500E-00		-1.0030E-01
CDFLT	1.133E-01	THETAS	-2.608E-00	THETALS	-2.509E-00	PR	1.2442E-02		

Reproduced from
best available copy.

5.0 REFERENCES

1. Xerox Sigma 5 Computer Reference Manual, Document 90 09 59E, October 1971.
2. Hazel, L.H. "Hybrid Library Reference Manual for XDS Sigma 5, CI-5000, CI-500 Hybrid Computer System", CODE Research Corporation Report No. 74/1, January 1974.
3. Jolly, A.C. "Missile Target Angular Relationships for Target and Rotational Motion Simulators", CODE Research Corporation Report No. 74/2, June 1974.
4. Betz, P., Cunningham, D., Strand, R., Wismer, K., "Laser Spot Homing Simulator", Boeing Aerospace Company, Research Division, Document D180-12926-1, June 1971.
5. Betz, P., Strand, R., Wismer, K., "Optical Contrast Homing Simulator", Boeing Aerospace Company, Research Division, Document D180-15374-1, April 1973.
6. Xerox Real-Time Batch Monitor RBM Reference Manual, Document 90 15 810, April 1971.
7. Xerox Real-Time Batch Monitor RBM Operations Reference Manual, Document 90 16 470, October 1972.
8. Xerox Real-Time Batch Monitor RBM User's Guide, Document 90 16 53B, October 1972.
9. CODE Research Corporation "Hybrid Remote Batch Monitor Operator/ User Manual for Redstone Arsenal", July 1971.
10. Xerox Extended FORTRAN IV-H for Sigma 5/7 Computers Reference Manual, Document 90 09 66E, March 1970.
11. Astrodata, Inc., "Continuous System Simulation Language (CSSL)" Document 121600.
12. Xerox Macro-Symbol for Sigma 5-9 Computers Language and Operations Reference Manual, Document 90 15 78B, June 1972.
13. Xerox Extended FORTRAN IV Reference Manual for Sigma 5-9 Computers, Document 90 09 560, April 1970.

14. Tektronix, Inc., "PLOT-10 Terminal Control System - 4002A User's Manual", Document No. 062-1464-00.
15. Tektronix, Inc., "PLOT-10 Advanced Graphing II User's Manual", Document No. 062-1530-00.
16. Spearman, F.R.J. "The Derivation and Use of Aerodynamic Transfer Functions of Airframes", Royal Aeronautical Society Journal, V.59 no. 539, November 1955, pp 743-761.

APPENDIX A
Program Listing

A-1

```

0 1: C MAIN PROGRAM - SAMPLE PITCH PLANE HYBRID SIMULATION.
2: C
3: C THE FUNCTION OF THE MAIN PROGRAM IS FOREGROUND INITIALIZATION.
4: C
5: COMMON /VARIABLE/ WTF,X2DOT,XDOT,X,Z2DOT,ZDOT,Z,CSTHA,
6: 1 SNTHA,ALPHA,DELTA,THETA,DAC1,DAC2,DAC3,
7: 2 DAC4,DAC5,DAC6,FYA,FZA,FXB,FZB,MASS,
8: 3 IYY,QUE,VELOCITY,SS,MACH,HALERHO,CX,
9: 4 CNALP,CNDEL,T,CNALP,SM'ELT,SMQ,THE TASI,
10: 5 THETA,BS,PR,X2DOTM1,Z2DOTM1,XDOTC,ZDOTC
11: C
12: COMMON /CONSTANT/ IWAIT,(30RT,DT,DT02,SKRT,THEMAX,PRMAX,
13: 1 XT,7T,NTACC,NTBUTY,NTDTP,SKRNT,ISKR,
14: 2 NTAP,INCRTAP,ALFASCAL,DELTSCAL,THESCALE,
15: 3 SFDAC1,SFDAC2,SFDAC3,SFDAC4,SFDAC5,
16: 4 DELMASS,DELIYY,GBAR,G,THRUST,RTD,RJNUM
17: 5 ICOUNT
18: REAL MASS, IYY, MACH
19: INTEGER RJNUM, SKRNT
20: DATA RJNUM/0/
21: C ALL SUBROUTINES ARE CONNECTED TO AN EXTERNAL INTERRUPT
22: C MUST BE LISTED IN AN EXTERNAL STATEMENT.
23: C
24: EXTERNAL CLOCK, ABRT, SEEKER, SCHEDULE,
25: 1 PRGRS, START RUN
26: C
27: DATA 1,575 /30.174,57.2257735/
28: DATA DAC1, SFDAC1, SFDAC2, SFDAC3, SFDAC4, SFDAC5
29: 1 /3.0, 3.0, 3.75, .0, .1/
30: C
31: C III HYBRID LIBRARY.
32: C
33: C
34: C
35: C MAKE SURE THE REAL-TIME CLOCK IS NOW RUNNING.
36: C
37: C CALL RTC(14)
38: C
39: C CONNECT INTERRUPTS.
40: C
41: CALL CHINT( 2, 3, CLOCK,
42: 1 3, ABRT,
43: 2 8, SEEKER,
44: 3 12, SCHEDULE,
45: 4 13, PRGRS,
46: 5 14, START RUN)
47: C
48: C DEFINE HYBRID INPUT/OUTPUT PATTERNS.
49: C
50: C CALL STRPT(1)
51: C CALL RACCP(2,2,CSTHA)
52: C CALL RACCP(3,ALPHA,DELTA)
53: C CALL ENDPAT
54: C
55: C CALL STRPT(2)
56: C CALL RACCP(2,DAC3,3,DAC1,5,DAC2,10,DAC5,12,DAC4)
57: C CALL ENDPAT
58: C
59: C CALL STRPT(3)
60: C CALL RACCP(2,DAC6)
61: C CALL ENDPAT
62: C

```

Reproduced from
best available copy.

63: C ARM AND ENABLE INTERRUPTS.
64: C
65: CALL INTSER(2,2,3,8,12,13,14)
66: C
67: CALL EXIT
68: END

A-3

Best Available Copy

```

1: SUBROUTINE START RUN
2: C
3: C THIS SUBROUTINE IS CONNECTED TO EXTERNAL INTERRUPT 14 WHICH
4: C IS PATCHED TO A LOGIC SWITCH ON THE DI-500.
5: C
6: COMMON /VARIABLE/ NTF,X2DBT,XDBT,X,Z2DBT,ZDBT,Z,CSTHA,
7: 1 STHA,A_PHA,DELTA,THETA,DAC1,DAC2,DAC3,
8: 2 DAC4,DAC5,DAC6,FXA,FZA,FXB,FZB,FZC,MASS,
9: 3 IYV,CUE,VELOCITY,SS,MACH,HALFR4,CX,
10: 4 CMA_P,CNDE,TICMA_P,CNDELT,CMD,THETAS,
11: 5 THETA_SS,PR,X2DBTM1,Z2DBTM1,XDBTC,ZDBTC
12: C
13: COMMON /CONSTANT/ TAIT,KRRT,DT,DEPR,RKRT,THEMAX,PRMAX,
14: 1 XT,ZT,TACC,NTURN,NTSTP,SKRNT,ISKR,
15: 2 NTPE,INOTPE,A_FASCA,DELTSCAL,THESCALE,
16: 3 SFAC1,SFAC2,SFAC3,SFAC4,SFAC5,
17: 4 DELMASS,DELIYV,S,CBAR,G,THRUST,RTD,RUNUM
18: 5 ICOUNT
19: REAL MASS,
20: INTEGER RUNUM,
21: NAME LIST
22: TIME, DT, TSTP, TACC,
23: 1 TSTRT, TAPTIME, IPJSE, THEMAX,
24: 2 PRMAX, XT, ZT, X,
25: 3 Z, VELOCITY, A_FASCA, DELTSCAL,
26: 4 TAPTIME, SFAC1, SFAC2, SFAC3,
27: 5 SFAC4, SFAC5, NTURN, STMASS,
28: 6 DELMASS, STIYV, ENDIYV, S,
29: 7 CBAR, G, THETA_IC, THRUST
30: 8 GAIN
31: INTEGER CARD1,CARD2
32: 9 DIVA-ENCE (CARD1,CARD2)
33: CALL DUMMY(1,1)
34: IF(1.EQ.0) RETURN
35: TIME = TACC = TSTRT = XT = X = XDBT = ZDBT = ZIC = THETA_IC =
36: 1 XDBT = ZDBT = STHA = THETA = ALPHA = DELTA = VELOCITY =
37: 2 FXA = FZA = FXB = FZB = CX = CMA_P = CNDELT = CMA_P = CNDELT =
38: 3 CMA_P = THETAS = THETA_SS = PR = CUE = MACH = THETA = DAC1 =
39: 4 DAC2 = DAC3 = DAC4 = DAC5 = DAC6 = 0.0
40: GAIN = 10.0
41: CSTHA = 1.0
42: CARD1 = KRRT = 0
43: XT = 3000.0
44: Z = -500.0
45: DT = 0.025
46: TSTP = 25.0
47: TAPTIME = 1.0
48: IPJSE = 20
49: THEMAX = 2.0
50: PRMAX = 2.0
51: A_FASCA = 20.0
52: DELTSCAL = -15.0
53: NTURN = 2.0
54: STMASS = 4.6; ENDMASS = 3.9
55: STIYV = 6.1; ENDIYV = 4.9
56: THRUST = 1500.0
57: S = 0.267
58: CBAR = 0.584
59: REWIND 2
60: RUNUM = RUNUM + 1
61: WRITE(108,1000) RUNUM
62: REPEAT 10, WHILE CARD1.NE.4END

```

Reproduced from
best available copy.

A-4

Best Available Copy

```

53: READ(150,1040)      CARD
54: 12 ASITE(108,1020)  CARD
55: REWIND 15:
56: LPUT(150)
57: THETA = THETA IC/RTD
58: CALL DOAGR(26,-DIO/150.0,22,-THETA IC/180.0)
59: CALL PATTN(3)
60: CALL TABLE IC
61: ALFASCA = ALFASCA/RTD
62: DELISCA = DELISCA/RTD
63: ICOUNT = IFIX(DT*1.E6 + 0.5)
64: DTSP = 0.5*DT
65: NTSPR = IFIX(TSPTR/DT + 0.5)
66: NTF = IFIX(T/DT + 0.5)
67: NTACD = IFIX(TACD/DT + 0.5)
68: NTBURN = IFIX(BURNOUT/DT + 0.5)
69: NTAPR = IFIX(TPRT/DT + 0.5)
70: INCRTAPR = IFIX(TAPRTIME/DT + 0.5)
71: ICKR = 100000/(ICOUNT*IPULSE)
72: SKRT = DT*DT*(ICKR)/RTD
73: SKRT = SKRT*GAIN
74: SKRNT = NTF
75: MASS = CMMASS
76: IYV = STYV
77: DELMASS = (ENDMASS - MASS)*DT/BURNOUT
78: DELIYV = (ENDIYV - IYV)*DT/BURNOUT
79: THETA = THETA/RTD
80: C
81: C WAIT FOR ICS TO CHANGE THEN READ ADDS.
82: C
83: CALL DELAY(1000)
84: CALL PATTN(1)
85: CALL PATTN(2)
86: XDOT = XDOTS = VELOCITY*SENTHA
87: ZDOT = ZDOTS = -VELOCITY*SENTHA
88: THETA = RTD*ATAN2(SENTHA,CSENTHA)
89: WRITE(108,1030) DT,TSPR,TACD,IPULSE,THRIST,BURNOUT,
90: 1 S,CHAR,Y,Z,XDOT,ZDOT,THETA,XI,ZI
91: CALL DELAY(2000)
92: IWAIT = 1
93: C
94: C TRIGGER INTERRUPT 2 TO COMPLETE INITIALIZATION.
95: C
96: CALL INTSER(7,2)
97: C
98: C START RUN.
99: C
100: CALL RTC(111)
101: CALL RTC(ICOUNT,2)
102: CALL MODE(101)
103: CALL RTC(131)
104: RETURN
105: C
106: 1000 FORMAT(11,T20,SAMPLE PITCH PLANE HYBRID SIMULATION *** RUN NUMBER
107: 1'I3//T10'INPUT DATA')
108: 1010 FORMAT(20A4)
109: 1020 FORMAT(T10,20A4)
110: 1030 FORMAT(/T10,3253.0' MS FRAME TIME. STPR AT10PF4.0' SEC. ACQUIST
111: 1T10N AT1F4.0' SEC. SEEKER UPDATES'I3' TIMES PER SECOND.'/T10
112: 2'THRUST156.0' 3'UNILL BURNOUT AT1F4.0' SEC. REFERENCE AREA *I
113: 3F6.3' 30 FT. REFERENCE LENGTH =1F6.3' FT./T10
114: 4'INITIAL MISSILE POSITION, VELOCITY AND ATTITUDE: (1F6.0',1F6.0

```

125: 51) FT. (1F5.01, 1E5.01) FT/SEC AND 1F4.01 DEGREES. 1/110
126: 51) TARGET AT (1F6.01, 1F6.01) FEET. 1/110
127: C
128: E N D

A-6

Best Available Copy

***** 9 J T I N E *****

THIS SUBROUTINE IS CONNECTED TO EXTERNAL INTERRUPT 2 WHICH
IS PATCHED TO REAL TIME CLOCK 1 BY THE CI-500 LOGIC BOARD.

COMMON /VARIABLE/

NTF,XDORT,XDST,X,ZDORT,ZDST,Z,ORTHA,
STHA,ALPHA,DELTA,THETA,DAC1,DAC2,DAC3,
DAC4,DAC5,DAC6,FXA,FZA,FXB,FZB,MASS,
IYY,QUE,VELOCITY,SS,MACH,HALFR49,CX,
CMA_P,CNDELTA,CMA_P,CNDELTA,CMD,THETAS1,
THETA9S,PR,XDORTM1,ZDORTM1,XDSTC,ZDSTC

COMMON /CONSTANT/

IAIT,KBORT,OT,OTPR,SKRT,THEMAX,PRMAX,
XT,ZT,NTACC,NTBURN,NTSTOP,SKRONT,ISKR,
NTAPE,INCRPAPE,ALFASCA,DELTSCL,THESCALE,
SEDAC1,SEDAC2,SEDAC3,SEDAC4,SEDAC5,
DELMASS,DELIYY,SS,CBAR,C,THRUST,RTD,RJUM
ICPJNT
REAL MASS, IYY, MACH
INTEGER RJUM, SKPJNT

READ ACOS (PATTERN 1) AND SCALE RESULTS

CALL PATTN (1)

ALPHA = ALPHA*ALFASCA

DELTA = DELTA*DELTSCL

CALCULATE TOTAL VELOCITY, SPEED OF SOUND, ATMOSPHERIC DENSITY
AND MACH NUMBER.

VSQARE = XDORT*XDORT + ZDORT*ZDORT

VELOCITY = SQRT(VSQARE)

SS = 1116.0*7 + 0.000622*7

HALFR49 = 0.00055-0.7 + 0.011860

MACH = VELOCITY/SS

BACK UP AERODYNAMIC FORCE AND MOMENT COEFFICIENTS

CALL TABLES

CALCULATE DYNAMIC PRESSURE AND FORCES.

QUE = HALFR49*VSQARE

QUES = QUE*SS

FXA = QUES*CX

FZA = FZB = QUES*(CNALP*ALPHA + CNDELTA*DELTA)

FXB = -FXA

NEW THRUST DEPENDENT TERMS.

IF(NTF > NTBURN) GO TO 10

IYY = IYY + DELIYY

MASS = MASS + DELMASS

FXB = FXB + THRUST

CONTINUE

CALCULATE AND OUTPUT DACS NEEDED BY ROTATION EQUATIONS.

FXMASS = FXB/MASS; FZMASS = FZB/MASS

OSCHAR = QUES*CBAR/IYY

DAC1 = OSCHAR*CMA_P/SEDAC1

A-7

Reproduced from
best available copy.

Best Available Copy

```

63:      DACC2 = (SCORAR*DELTA)/SFDACC2
64:      DACC3 = -1.01      DACC4 = 0.01      DACC5 = 1.0
65:      IF (VELOCITY < 1.0) GOTO 20
66:      DACC2 = 0.5*(SCORAR*CRAN*CRAN)/(VELOCITY*SFDACC2)
67:      IF (NTF < NTBJRN) GOTO 20
68:      DACC4 = (FZMASS + 3*GSTHA)/(VELOCITY*SFDACC4)
69:      DACC5 = (FXMASS + 3*SNTHA)/(VELOCITY*SFDACC5)
70: C
71: PD    CALL PATTERN(P)
72: C
73: C    CALCULATE AND INTEGRATE TRANSLATION DERIVATIVES.
74: C
75:      XD9T41 = XD9T;      ZD9T41 = ZD9T
76:      XD9T = FXMASS*GSTHA + FZMASS*SNTHA
77:      ZD9T = FZMASS*GSTHA - FXMASS*SNTHA + 3
78:      XD9TC = XD9TC + DT92*(XD9T + XD9T41)
79:      ZD9TC = ZD9TC + DT92*(ZD9T + ZD9T41)
80:      XD9T = XD9TC + DT92*(3.0*XD9T - XD9T41)
81:      ZD9T = ZD9TC + DT92*(3.0*ZD9T - ZD9T41)
82:      X = X + DT92*(XD9TC + XD9T)
83:      Z = Z + DT92*(ZD9TC + ZD9T)
84: C
85: C    TRIGGER INTERRUPT 12 TO CHECK FOR SCHEDULING OF OTHER TASKS.
86: C
87:      CALL INTSER(7,12)
88: C
89: C    INCREMENT TIME FRAME COUNTER.
90: C
91:      NTF = NTF + 1
92: C
93:      RETURN
94:      END

```

SUBROUTINE TABLE

THIS SUBROUTINE PERFORMS A LOOKUP FOR AERODYNAMIC FORCE AND MOMENT COEFFICIENTS AS FUNCTIONS OF MACH NUMBER.

COMMON /VARIABLE/ NTE,XPDRT,XDRT,X,ZDRT,ZDRT,Z,CSTHA,
 1 SNTHA,A_PHA,DELTA,THETA,DAC1,DACP,DAC3,
 2 DAC4,DAC5,DAC6,FYA,FYA,FYA,FZA,MASS,
 3 IYV,QUE,VELOCITY,SS,MACH,HALFR49,CX,
 4 CMALP,CNDELTA,CMALP,CNDELTA,CM2,THETAS1,
 5 THETA,PS,PR,XPDRTM1,ZPDRTM1,XDRTC,ZDRTC

COMMON /CONSTANT/ IMAIT,KBRRT,DT,DT2,SKRDT,THEMAX,PRMAX,
 1 XT,ZT,NTACD,NTORRN,NTSTRP,SKRONT,ISKR,
 2 NITPE,INCRTAPE,A_FASCAL,DELTSCL,THESCALE,
 3 SFDA1,SFDA2,SFDA3,SFDA4,SFDA5,
 4 DE_MASS,DE_IYV,S,CBAR,B,THRUST,RTD,RUNUM

REAL MASS, IYV, MACH
 INTEGER RUNUM, SKRONT

DATA IS SPECIFIED AT MACH NUMBERS .45, .8, 1.0, 1.2 AND 1.4

DIMENSION CMAT(5), CMT(5), CMAT(5), CMT(5),
 1 CXT(5), CMT(5), ACNA(4), BCNA(4),
 2 ACND(4), BCND(4), ACMA(4), BCMA(4),
 3 ACND(4), BCND(4), ACX(4), BCX(4),
 4 ACMD(4), BCMD(4)

NORMAL FORCE COEFFICIENT DUE TO ANGLE OF ATTACK.

DATA CMAT/.45, 14.9, 16.6, 16.75, 12.89/

NORMAL FORCE DUE TO WAVE ANGLE.

DATA CMT/.45, 2.16, 2.16, 2.24, 2.25/

MOMENT DUE TO ANGLE OF ATTACK.

DATA CMAT/-7.44, -10.08, -9.78, -6.90, -2.25/

MOMENT DUE TO WAVE ANGLE.

DATA CMT/3.57, 3.57, 12.4, 12.37, 8.05/

DAMPING DERIVATIVE.

DATA CMT/-345., -405., -410., -405., -335./

DRAW COEFFICIENT.

DATA CXT/.305, .305, .54, .755, .832/

CALCULATE INTERPOLATION INDEX.

I = IFIX(5.0*MACH) - 2

IF(I > 4) I = 4

IF(I < 1) I = 1

NEW INTERPOLATE USING PRE-CALCULATED COEFFICIENTS.

CX = ACX(I) + BCX(I)*MACH

CMALP = ACNA(I) + BCNA(I)*MACH

CNDELTA = ACND(I) + BCND(I)*MACH

CMALP = ACMA(I) + BCMA(I)*MACH

CNDELTA = ACMD(I) + BCMD(I)*MACH

CM2 = ACMD(I) + BCMD(I)*MACH

RETURN

ENTRY TABLE IS

THIS ENTRY IS CALLED TO CALCULATE INTERPOLATION COEFFICIENTS
 USED IN REAL-TIME BY THE TABLE LOOKUP ROUTINE.

Reproduced from
 best available copy.

```

53:      DO 100 I = 1,4
54:      XMI = 0.1 * AT(2*(I+2))
55:      BCNA(I) = 5.0*(CNAT(I+1) - CNAT(I))
56:      BCND(I) = 5.0*(CNDT(I+1) - CNDT(I))
57:      BCMA(I) = 5.0*(CMAT(I+1) - CMAT(I))
58:      BCMO(I) = 5.0*(CMDT(I+1) - CMDT(I))
59:      BCX(I) = 5.0*(CXT(I+1) - CXT(I))
60:      BCMQ(I) = 5.0*(CMQT(I+1) - CMQT(I))
61:      ACNA(I) = CNAT(I) - XMI*BCNA(I)
62:      ACND(I) = CNDT(I) - XMI*BCND(I)
63:      ACMA(I) = CMAT(I) - XMI*BCMA(I)
64:      ACMO(I) = CMDT(I) - XMI*BCMO(I)
65:      ACX(I) = CXT(I) - XMI*BCX(I)
66:      ACMQ(I) = CMQT(I) - XMI*BCMQ(I)
67:      100
77:      RETJRN
78:      END

```

A-10

Best Available Copy

```

1: SUBROUTINE SCHEDULE
2: C
3: C THIS SUBROUTINE IS CONNECTED TO EXTERNAL INTERRUPT 12 WHICH
4: C IS TRIGGERED BY SUBROUTINE CLOCK.
5: C COMMON /VARIABLE/ NTF,XDOT,XDOT,X,7PDOT,ZDOT,Z,CSTHA,
6: 1 SMTHA,ALPHA,DELTA,THETA,DAC1,DAC2,DAC3,
7: 2 DAC4,DAC5,DAC6,FXA,FZA,FX3,FZB,MASS,
8: 3 IYY,QUE,VELOCITY,SS,MACH,HALFRHO,CX,
9: 4 CNALP,CNDEL,CNALP,CNDEL,CNQ,THETAS1,
10: 5 THETALOS,PP,Y2DOTM1,Z2DOTM1,XDOTC,ZDOTC
11: C
12: C COMMON /CONSTANT/ INAIT,KBPRT,DT,DT2,SKRDT,THEMAX,PRMAX,
13: 1 XT,ZT,NTACC,NTBUEN,NTSPP,SKRCNT,ISKP,
14: 2 NTAPE,INCRTAPE,ALFASCAL,DELTSICAL,THESCALE,
15: 3 SFDAC1,SFDAC2,SFDAC3,SFDAC4,SFDAC5,
16: 4 DELMASS,DELIYY,S,CBAR,G,THRUST,RTD,RUNUM
17: 5 ICPUJT
18: REAL MASS, IYY, MACH
19: INTEGER RUNUM, SKRCNT
20: INTEGER TAPESTAT
21: DATA ISKP, TAPESTAT / -1,2 /
22: C
23: C CHECK FOR RUN TERMINATION.
24: C
25: C IF A PROJECTION OF X OR Z THROUGH ONE MORE INTEGRATION WILL
26: C PLACE THE MISSILE PAST THE TARGET, TERMINATE.
27: C
28: C IF(X+DT*XT > XT + PR. Z+ZDOT*DT > ZT )
29: 1 INAIT = 0; GO TO 40
30: C
31: C IF TIME IS PAST THE INPUT STOP TIME, TERMINATE.
32: C
33: C IF(NTF > NTAPE) INAIT = 0; GO TO 40
34: C
35: C NO TERMINATION, SO SEE IF TIME FOR SEEKER UPDATE.
36: C
37: C IF(NTF < SKRCNT) GO TO 10
38: C
39: C TRIGGER INTERRUPT 8 TO ENTER SUBROUTINE SEEKER.
40: C
41: CALL INTSER(7,8)
42: SKRCNT = SKRCNT + ISKR
43: C
44: C SEE IF TIME FOR TAPE WRITE.
45: C
46: 10 IF(NTF < NTAPE) GO TO 30
47: IF(TAPESTAT.EQ.1) GO TO 30
48: 20 CALL BUFFER OUT(2,1,NTF,38,TAPESTAT)
49: NTAPE = NTAPE + INCRTAPE
50: C
51: 30 IF(INAIT > 0) RETURN
52: C
53: C RUN TERMINATION: STOP THE CLOCK, AND GO TO RESET MODE.
54: C WRITE FINAL TAPE RECORD AND PRINT DATA.
55: C
56: 40 CALL REC(141) CALL MADE(12)
57: DAC6 = 0.0
58: CALL PATTERN(5)
59: TIME = DT*F9AT(NTF)
60: IF(KBPRT > 0) GO TO 200
61: ISKP = ISKP + 1
62: IF(ISKP > 0) GO TO 60

```

Reproduced from
best available copy.



A-11

Best Available Copy

```

53: 50 IF(TAPESTAT .EQ.1) GO TO 50
54: 51 TO 20
55: 60 ICKIP = ICKIP + 2
56: 70 IF(TAPESTAT .EQ.1) GO TO 70
57: END FILE 2
58: REWIND 2
59: IF(NTF > NTEST) GO TO 100
60: C
61: C MISS DISTANCE CALCULATION.
62: C
63: C NEGLECTING ACCELERATION, THE RANGE FROM TARGET TO MISSILE IS
64: C
65: C  $R = \text{SQRT}((X1 - X + XDOT*(T - TIME))^2 + (Z1 - Z + ZDOT*(T - TIME))^2)$ 
66: C
67: C WHERE 'TIME' IS THE TIME OF RUN TERMINATION. SETTING THE
68: C DERIVATIVE OF R WRT T TO ZERO AND SOLVING FOR T GIVES THE
69: C PROJECTED TIME OF CLOSEST APPROACH.
70: C
71: C  $T = (XDOT*(X1 - X + XDOT*TIME) + ZDOT*(Z1 - Z + ZDOT*TIME)) /$ 
72: C  $(XDOT^2 + ZDOT^2)$ 
73: C
74: C  $XMIS = X1 - X + XDOT*(T - TIME)$ 
75: C  $ZMIS = Z1 - Z + ZDOT*(T - TIME)$ 
76: C  $R = \text{SQRT}(XMIS^2 + ZMIS^2)$ 
77: C
78: C PRINT MISS DISTANCE.
79: C
80: C WRITE(109,100) R,T, XMIS,ZMIS
81: C GO TO 110
82: C
83: C WRITE(109,101) TIME
84: C
85: C VIA PRINT DATA FROM TAPE.
86: C
87: 110 CALL BUFFER I (2,1,NTF,35,TAPESTAT)
88: 120 GO TO (120,140,110,130), TAPESTAT
89: 130 WRITE(109,102)
90: 140 TIME = DT*FLOOR(NTF)
91: THETA = ATAN2(SIN(THA,CST(THA)*RTD)
92: WRITE(108,103) TIME,XDOT,XDOT,Y,ZDOT,ZDOT,ZALPHA*RTD
93: 1
94: DELTA*RTD,THETA,DAC1,DAC2,DAC3,
95: DAC4,DAC5,FXA,FZA,FXB,FZB,MASS,IYY,IJE,
96: 3 VELOCITY,SS,MACH,2*HALFR49,CX,CNALP,
97: 4 CNDELT,CNALP,CNDELT,RTD*THETAS1,
98: 5 RTD*THETA_SS,PR
99:
100: GO TO 110
101: WRITE(108,104) TIME
102: 200
103: WRITE(108,104) TIME
104: RETURN
105: C
106: 1000 FORMAT(/' *** MISS DISTANCE ***'/'10.2'/' FT AT TIME ='
107: 1 10.3'/' SECONDS. XMIS ='/'10.2'/' ZMIS ='/'10.2'/' )
108: 1010 FORMAT(/' *** INTERCEPT HAS NOT OCCURRED BY TIME'/'10.3'/' )
109: 1020 FORMAT(/' TAPE READ ERROR DETECTED IN THE FOLLOWING RECHRD.' )
110: 1030 FORMAT(/'5X,1P5E24.4T10'/'TIME'/'B4'/'X ACC'/'T5'/'X VEL'/'T8'/'X'/'T106
111: 1 1'/'Z ACC'/'5X5E24.4T10'/'Z VEL'/'T34'/'Z'/'T58'/'A'/'P44'/'T82'/'DELTA'/'T106
112: 2 1'/'THETA'/'5X5E24.4T10'/'DAC1'/'T34'/'DAC2'/'T58'/'DAC3'/'T82'/'DAC4'/'T106
113: 3 1'/'DAC5'/'5X5E24.4T10'/'FXA'/'T34'/'FZA'/'T58'/'FXB'/'T82'/'FZB'/'T106'/'MASS'/'
114: 4 5X5E24.4T10'/'IYY'/'T34'/'IJE'/'T58'/'VELOCITY'/'T82'/'SS'/'T106'/'MACH'/'
115: 5 5X5E24.4T10'/'R49'/'T34'/'CX'/'T58'/'CNALP'/'T82'/'CNDELT'/'T106'/'CNALP'/'
116: 6 5X5E24.4T10'/'CNDELT'/'T34'/'THETAS1'/'T58'/'THETA_SS'/'T82'/'PR'/' )
117: 1040 FORMAT(/'10('' *'')' R J N A B B R T E D A T I F S.2
118: 1 1 ' S E C O N D S '10('' *'')
119: END

```

SUBROUTINE SEEKER

THIS SUBROUTINE IS CONNECTED TO EXTERNAL INTERRUPT WHICH IS TRIGGERED BY SUBROUTINE SCHEDULE 'PULSE' TIMES PER SECOND TO PERFORM SEEKER UPDATES.

COMMON /VARIABLE/ NT, X200T, XDOT, X, Z200T, ZDOT, Z, CSTHA, SNTHA, ALPHA, DELTA, THETA, DAC1, DAC2, DAC3, DA24, DAC5, DAC6, FXA, FZA, FXB, FZB, MASS, IYY, QUE, VELOCITY, SS, MACH, HALER, HX, CNALP, CNDEL, CMA, P, CNDEL, CMO, THETA, THETAS, AR, X200TM1, Z200TM1, XDOTC, ZDOTC

COMMON /CONSTANT/ IWAIT, BORT, DT, DT02, SKRDT, THENAX, PRMAX, XT, ZT, NTACQ, NTBURN, NTSTOP, SKRONT, ISKR, NTAPE, INCRTAPE, FASCAL, DELTSCAL, THESCALE, SEDAC1, SEDAC2, SEDAC3, SEDAC4, SEDAC5, DELMASS, DELIYY, S, CBAR, J, THRUST, RTS, RUNUM, ICOUNT, REAL MASS, IYY, MACH, INTEGER RUNUM, SKRONT, INTEGER FLAG, DATA FLAG, /Z/

IF (IF < TACQ) GO TO 20
IF (FLAG .EQ. RUNUM) GO TO 5
THETANI = ATAN2(SNTHA, CSTHA)
THETA = THETANI
THETADIF = THETANI - THETAS
IF (ABS(THETADIF) > THENAX) PR = -PRMAX + THETADIF
PR = PRMAX + THETADIF / THENAX
CONTINUE

SCALE PRECESSION RATE AND OUTPUT ON A DAC.

DAC6 = PR / PRMAX
CALL PATTN(3)
RETURN
END

Reproduced from
best available copy.

SUBROUTINE ABORT

```

1:
2:
3: C THIS SUBROUTINE IS CONNECTED TO EXTERNAL INTERRUPT 3 WHICH
4: C IS PATCHED TO A LOGIC SWITCH ON THE CI-500 AND IS USED TO
5: C ABORT A RUN.
6: C
7: COMMON /CONSTANT/ IWAIT, <BORT, DT, DT02, SKRDT, THEMAY, PPMAX,
8: 1 XT, ZT, NTACO, NT3, PN, NTSTOP, SKRCNT, ISKR,
9: 2 NTAPE, INCRTAPE, ALFASCA, DELTSCAL, THESCALE,
10: 3 SEDAC1, SEDAC2, SEDAC3, SEDAC4, SEDAC5,
11: 4 DELWASS, DELIYY, S, COAR, G, THRUST, RTD, RUNJM
12: REAL MASS, IYY, MMCH
13: INTEGER RUNJM, SKRCNT
14: CALL RTC(1,1)
15: <BORT = 1
16: IWAIT = 0
17: RETURN
18: END

```

Reproduced from
best available copy.

A-14

Best Available Copy