NASA CONTRACTOR REPORT 166449

(NASA-CR-166449) MULTI-MAN FLIGHT SIMULATOR Progress Report, Oct. 1979 - Sep. 1980 (Tufts Univ.) 44 p HC A03/MF A01 CSCL 17B

N83-21210

Unclas
G3/32 09177

Multi-Man Flight Simulator System

Gary MacDonald

CONTRACT NCC-2-93
February 1983

NASA

NASA CONTRACTOR REPORT 166449

ORIGINAL PAGE IS
OF POOR QUALITY

Multi-Man Flight Simulator System

Gary MacDonald
Tufts University
Department of Engineering Design
Medford, Massachusetts

Prepared for
Ames Research Center
Under Grant NCC-2-93

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

ACKNOWLEDGMENTS

System components have been developed and are mentioned in the text with the permission, agreement of or contract by the following vendors:

The Digital Equipment Corporation (DEC), Maynard, Mass.;

The ADAC Corporation, Woburn, Mass.;

The PACER Corporation, Burlington, Mass.;

The Aviation Simulation Technology Corporation (AST), Bedford, Mass.

TABLE OF CONTENTS

INTRODUCTION

Purpose

The purpose of this document is to summarize progress on the Multi-man Flight Simulator Project (NASA NSG-2156) for the period September 1979 to September 1980. The goal of this report is to present this period's work in context of all previous work and especially as it pertains to the design changes that were initiated in the September 1978-September 1979 period.

Organization

This report begins with a general description of the Multi-man System as it currently exists, followed by a brief review of the evolution of the system in terms of work accomplished in the previous three and this current reporting period, with the other sections of the report discussing the details of recent progress. Also, a glossary of terms is provided in Appendix 1.

Design Overview

The complete Air Traffic Control (ATC) experimental facility consists of three major systems as shown in Figure 1.

Multi-man System

The Multi-man System shown in Figure 2 is modularly designed to accommodate up to eight Flight Simulators by the Host Subsystem and Data Communications Systems. The Multi-man System may be operated as a facility without support by the Mainframe as long as CRT graphics are not required, as for example, during program development, or during experiments not needing polit CRT displays and in which only limited data storage is required. This design feature eliminates any unnecessary burden on the Mainframe/graphics, releasing them for other experimental, developmental or maintenance purposes. Used in the standalone or "local" mode, the Multi-man System can, for example, support experiments using standard flight instruments. An auxiliary benefit of the multiple independently programmable simulation is the ability to run subjects simultaneously rather than sequentially to increase experimental throughput.

Flight Simulator Subsystem

The Flight Simulator Subsystem consists of a commercially available[2] instrument trainer, a minicomputer and an interfacing panel as shown in Figure 3.

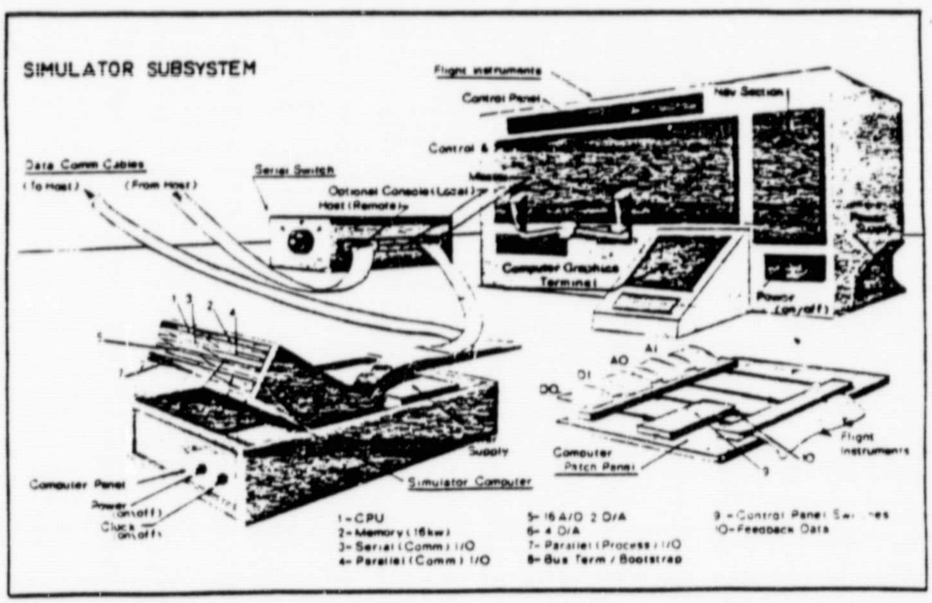


Fig. 3: Flight Simulator Subsystem

Flight Panel

The instrument trainer (Flight Panel) provides a high degree of visual and functional fidelity and can be operated as a self contained unit (local mode) without the Simulator Computer and is useful in this mode for training, etc. For developmental or experimental purposes, the Panel operates in remote mode supplying pilot inputs (switch selections, aileron, elevator, throttle, etc.) to the Simulator Computer which computes flight dynamics and outputs signals for the electro-mechanical flight instruments. Compensated signals drive phase-locked loops around each motor driven instrument using circuitry supplied by the Panel manufacturer. It is possible, although not presently configured, to input to the minicomputer the pilot inputs and quantities displayed on the flight and navigation instruments when the Panel is operated in the local mode. This would permit use of the Panel's internal flight dynamics and navigation sections allowing the minicomputer, for example, to compute and forward to the Host the flight path predictor, X, Y, Z position, etc.

Simulator Computer

The Simulator Computer is a commercially available mini/micro computer structured around Digital Equipment Corporation's LSI-11, as diagrammed in Figure 4. The Simulator Computer uses 16K-words of MOS memory for on-line

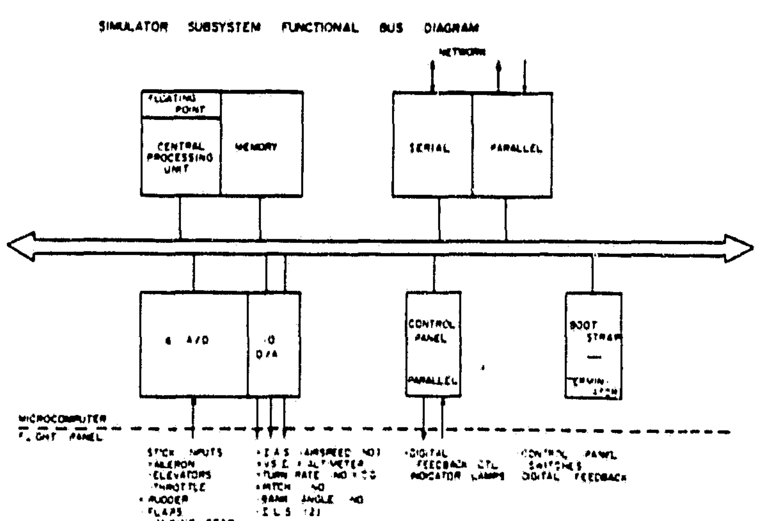


Fig. 4: Simulator Subsystem Bus Diagram

experiment related software which is downloaded from the Host at initial startup. Standard vendor logic boards are used for CPU and RAM and a micro-coded floating point chip is used for on-line software arithmetic. The basic 100 ms software duty cycle is segmented by a standard 100 Hz or 40 Hz clock. Eight (12 bit) analog outputs and seven (12 bit) analog inputs are used in interfacing to the Panel. Sixteen discrete inputs for Control Panel operation and eight discrete outputs for status annunication are required. Four additional discrete outputs are required for phase-encoded feedback circuit control and 12 inputs (plus interrupt) are needed for feedback circuit data input. These latter interfaces are sofeware multi-plexed with a single circuit on the Patch Panel. Resolution is expandable by Patch Panel-Local oscillator tuning and additional discrete inputs.

The Patch-Panel interfaces the Simulator Computer and Flight Panel with phase-encoded analog feedback circuits and minimal hardware signal transducing.

Software Components

Figure 5 shows the off-line and on-line program structures in A and B. The Simulator Computer software provides on-line experiment flight controls. A specially-developed foreground/background operating system provides real-time synchronous 100 ms framing for flight dynamics, predictor, ILS and data communications to the Host. The background function provides control panel and data communications from the Host when the foreground is not running. Foreground functions require approximately 80 ms as follows: flight dynamics (40 ms), predictor, ILS and data communications (40 ms about equally divided).

The flight dynamics are based on a simplified "NAVION" model and

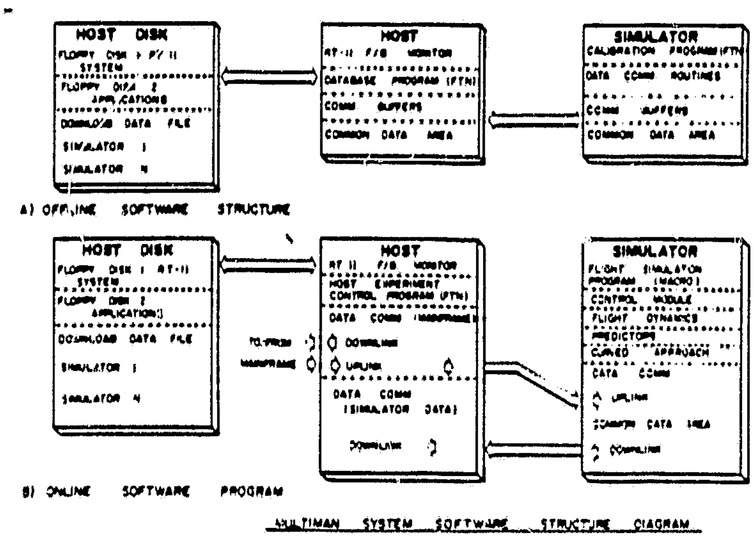


Fig. 5: Multi-man System Software Structure Design

are written in assembly language since the Simulator Computer does not support FORTRAN. A number of calculations and automatic control related macro-codes were developed to support modularization of the flight dynamics and other functions in conjunction with the LSI-11 Computer.

A predictor is programmed to project the aircraft's position at six points into the near future. The present (but programmable) pilot selectable options are for a 15 or 30 second predictor. The predictor-model uses bank angle and airspeed to compute positions relative to the aircraft's absolute position. Wind effects are then added and the values are converted from floating point to integer for updating to the Host and the Mainframe computers.

At present, a curved approach deviation function is computed and displayed on the ILS for any of five approaches used in a previous experiment. Once an approach is selected by the pilot, the ILS Guide Slope and localized indicators are driven to indicate position deviations from the ideal path maintained in the software model. A command function from the Host can it during calibration sequences then uploads the (tuned) segment back to the Host for storage. When on-line software is downloaded to the Simulator Computer, it receives the updated common segment from the Host and then begins operations.

Data Communications

Each simulator has two (point-to-point) data communications links with the Host. A bidirectional, serial, asychronous, RS-232C link provides download support (serial, binary). Power-up/start-up LSI-11 ROM "Absolute Leader" code is used for downloading. Applications software uses parallel, asychronous channels in two-way, alternate, "burst" mode for data transfer. A specially developed protocol is used to provide data transfer of 16-words (Simulator-to-Host), 8-word (Host-to-Simulator) and variable length words for off-line communication in either direction. Transfers require a single interrupt per frame with an appropriate 200 microseconds average word transfer rate including all latencies and service overhead.

Host Subsystem

The Host Subsystem shown in Pictorial and Functional form in Figures 6 and 7 performs three major functions. The first function is the primary program development base for the Multi-man System supported by mass storage on dual floppy disks and hardcopy by the teleprinter. The second function is the source and controller for downloading Simulator computers over 9600-band serial asychronous lines (parallel line downloading is also supported). The third function is the data concentrator/distributor "store-and-forward" center between Mainframe and Simulator computers during on-line experiments. Bidirectional parallel DMA links between Host and Mainframe and "star" configuration asychronous, parallel links between

ORIGINAL PAGE IS
OF POOR QUALITY

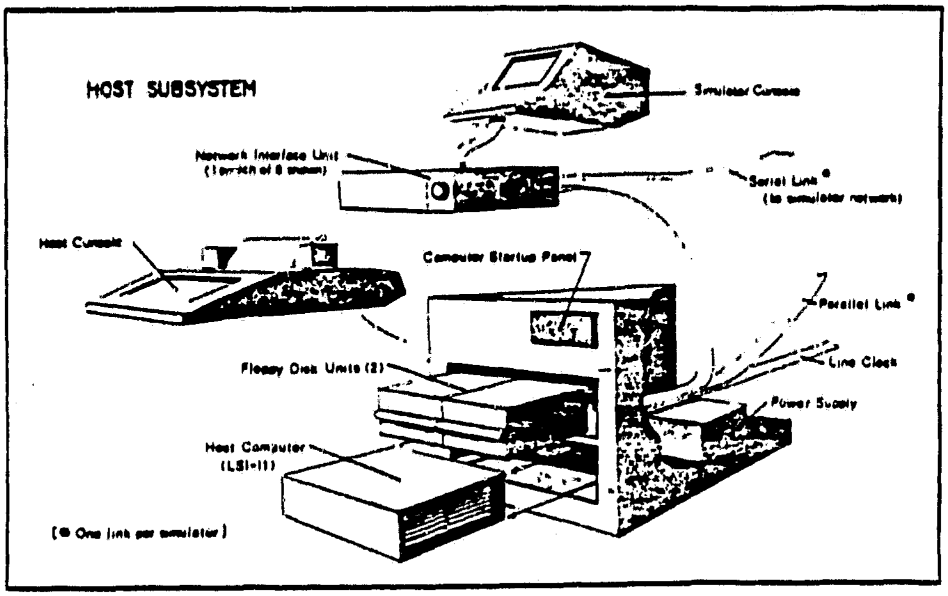


Fig. 6: Host Subsystem Pictoral

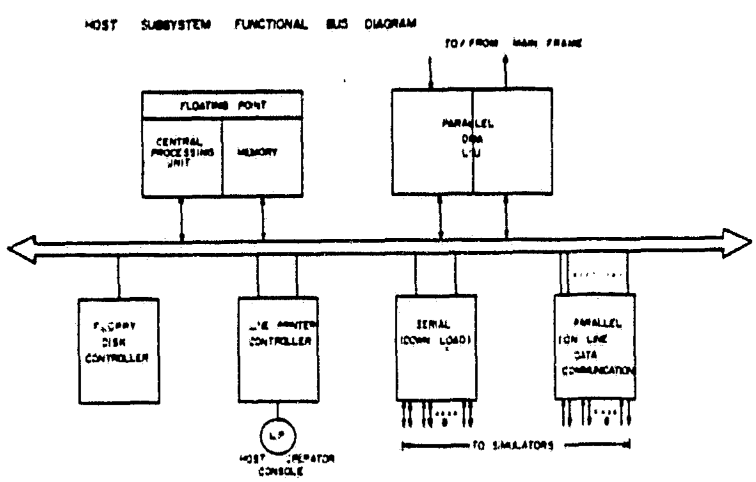


Fig. 7: Host Subsystem Functional Bus Diagram

Host and Simulators support on-line data communications.

The Host's software development base is Digital Equipment Corporation's RT-11 (Foreground/Background) operating system, using FORTRAN. Off-line simulator programs are FORTRAN-based ("stand alone FORTRAN") using MACRO-11-based data communications modules. On-line simulator programs are MACRO-11-based using "FORTRAN-callable" subroutines for ease of conversion to/from FORTRAN and for remote debug.

A manual switching unit at the Host is used with a portable CRT for remote/local manual operation of a Simulator Computer and for download. A specially developed asynchronous driver permits downloading of binary data over serial lines. Higher-speed downloading over parallel lines is also used, although RT-11 "load image" alterations must be made for this case.

The Host maintains a database on floppy disk which corresponds to the calibration data for each simulator. This data area is downloaded to the Calibration program running in a Simulator at the start of a calibration session. Subsequently, the same data segment is re-downloaded to the Flight Simulation Program for on-line startup; individual elements of the database may be changed on-line by the Uplink Data Communications function. This dynamic property is used for changing parameters such as winds and turbulence factors.

The Host's on-line program serves as a store-and-forward processor. Data is "pipelined" from Simulators to Mainframe on the "downlink", and commands are dispatched to Simulators (originating in the Host for Multi-man link control and in the Mainframe for overall experiment control) over the "uplink". Some experimental programs have performed data storage on the Host's floppy disk, but gaps in data result from device latency limitations; therefore, on-line Host-local data storage is not specified.

The uplink data communications structure provides for 8-word messages. Uplink messages cause overlaying of the Simulator's Control Panel Mask for remote Host control; in addition, a number of experimental control messages are used to alter parameters dynamically. The downlink data communications structure provides for 16-word messages. The downlink in the Host sees one 16-word message per simulator per 1/10 second; this amounts of 16 X 10 X 8 (simulators), or 1280 words/second over the parallel, asynchronous links, necessitating "burst mode" processing. The average latency in the Host is about 200-250 microsecond/words on the downlink; this amounts to 3.2-4 millisecond/message, or about 26-32 millisecond/100 millisecond real-time cycle (26-32%).

At initial startup, about 30 seconds are required to download a Simulator Computer of which the actual data transfer requires about 20 seconds. Complete design and operating details of the Multi-man Simulator System can be found in Kreifeldt[3].

BACKGROUND

This section is a brief review of the evolution of the Multi-man System through the present. This reporting period spans September of 1979 through the summer of 1980, which is referred to as "Phase 4". Previous "phases" of development correspond to the previous progress reports.

Phase 1 Summary[4]

The prototype flight simulator system, consisting of PDP-11/03 micro-(mini) computer system and PACER MK 11[5] flight instrument and interfaces, was set up in the Engineering Design Laboratory at Tufts University, under the direction of Professor J. G. Kreifeldt. A software base consisting of NASA-AMES (Research Center, CA)-developed "NAVION" flight dynamics and Tufts-developed support programs was prototyped and benchmarked. It was concluded that the Computer-Flight Instrument system was viable as a standalone simulator vehicle. The prototype software used about 10+K-words of primary storage, and the software cycle time for flight dynamics and predictors was about 60-65 milliseconds. Complete programming in FORTRAN was not possible, primarily due to the slowness of execution speed when using standard FORTRAN sine/cosine calls; therefore, a number of MACRO-11 assembly language-based modules were needed to achieve the cycle time noted. The computer used for interfacing was the intended "Host" computer, and a separate "standalone simulator" computer was ordered.

A number of performance improvements were identified, including more complete functionality and better flight dynamic performance, data communications and download/startup of a distributed simulator system.

Phase 2 Summary[6]

Functional performance of the simulator software was improved with the addition of a curved approach function, more refined flight dynamics (related to the actual flight instrument being used), better calibration procedures, and control panel functions. The requirements for distributed data communications, on-line and download/startup, were analyzed in depth. The requirements for Host on-line performance were analyzed and specifications included in the progress report for further development during the present period. All of the software modules were prototyped in assembly-language for improved cycle time and in anticipation of distribution to a "remote", standalone computer. The standalone computer was assembled with CPU, memory, and console.

A major architectural change was specified in the re-assignment of on-line data communications from serial, asynchronous to parallel, asynchronous. This change was based on analysis of data loading with the Mainframe computer driving the CRT-based graphics at the simulators. The data volume resulting from transmitting predictors from simulator through Host to Mainframe every 100 milliseconds could not be handled effectively by the Host using serial methods. The Host-Mainframe link was specified to be "synchronous", parallel (DMA) in order to accommodate N (N=8-10) simulators' data at the rate of 10 times per second.

It was noted that the interface of the "NAVION" flight dynamics model with the electro-mechanical flight instrument introduced errors into the altitude, heading, and bank angle control loops. This resulted because of "lags" in the movement of motors (driving indicator disks or needles); the result was that calculated values deviated substantially from actual indications. Decoupling of rate indicators from "position" indicators and driving both separately in an error-corrected scheme was recommended for development during the present period. Also noted were noise and feedback resolution-retrace problems in the motor interfaces.

Improvements were recommended in the maintenance and downloading of calibration-related database. Reliance on source-edit and re-assembly and re-linking was unwieldy considering the number of datapoints that needed to be maintained. Furthermore, algorithmic least-squares linear fitting resulted in calibration values which could be further refined when verified in actual operation.

Phase 3 Summary[6]

The standalone simulator system was set up and process interfaces were removed from the Host and implemented in the simulator computer. Data communications handlers were developed for download startup of the simulator using serial, asynchronous and parallel, asynchronous media. Data communications handlers were developed for on-line data updating of the Host from simulators using parallel, asynchronous; handlers were developed for on-line data communications from Host to simulators for experiment control. Control loops in the flight simulator software were improved such that calculated positions and actual positions are closely related. Predictor calculations were improved to provide more accurate prediction of "aircraft" position based on control inputs. Program development software was updated with a newer version of the RT-11 operating system[7] and a newer version of the FORTRAN compiler with an effective standalone object-time system.[8] The calibrations utility was greatly improved using FORTRAN; a database utility was developed and interfaced to the flight simulator software and calibrations software for on-line, global database maintenance and downloading.

Limitations in simulation performance using the present panel led to analysis of a new flight instrument, the Aviation Simulation Technology (AST), Model 300.[9] A design specification for implementation of the AST panel in place of the present panel was developed. An AST flight instrument panel was ordered and is expected to arrive at the end of the present period; implementation is intended during the next development period.

The Mainframe computer system performance requirements and expectations were analyzed with respect to interface to the Multi-man System. Design specifications for the Host subsystem were prepared and are included in this report. Mainframe architecture is based on discussions with development groups at NASA-AMES.

At this point in development of the standalone simulator, all major structures have been prototyped. The simulator can be started up via download from the Host, commanded by the local user ("pilot") or Host computer to begin flight simulation and/or data communications with the Host, operated to perform curved approaches along approach paths used in present experiments at NASA-AMES, and taken off-line without degrading on-line experiments with other simulators. Off-line calibrations and database utilities can be used to provide efficient tuning of flight instruments, and the database utility can be used with the system global database for on-line downloading of current calibration data.

All major structures at the Host have been prototyped with the exception of Host-Mainframe data communications. The Host can be used to maintain the global database, startup individual or all standalone simulators, operate in the context of a data collection-distribution center for the Multi-man when operating experiments, and operate in the context of a data storage center and command distribution center for simulator control in remote mode. Host-Mainframe data communications specifications are noted in this report and will have some impact on further development of the Host; however, development of on-line Host functions have taken into account expected Mainframe operations in order to minimize architectural impacts.

Phase 4 Summary

The principal efforts during this period have focused on integrating the Aviation Simulation Technology Corporation (AST) Model 300 Flight Simulator into the Multi-man System. The major aspect of the integration process was the development of two interfaces; the first, between the AST panel and the existing A/D and D/A computer interface hardward, the second, an error correction interface for motor position control during operation of the flight panel in its "remote" mode.

Development of the interfaces was conducted as a parallel process beginning with a thorough study of the AST schematics and physical panel. This study was directed towards understanding the functional architecture as it pertained to identifying and locating the various input/output control signals required by the Multi-man System. Also for evaluating the motor control scheme implemented by the panel so as to adopt an optimal control scheme for the feedback interface to the simulator computer.

The design adopted for the A/D and D/A interface stems from functional organisation of the AST panel into four major printed circuit boards, two responsible for flight characteristic computations, one responsible for navigation functions and the fourth housing the Intel 8085 micro-processor.

The chief requirement of the inferfaces' function was to permit access to control signals (i.e., elevator, ailerons, throttle, airspeed, etc.). Careful study of the AST printed circuit boards permitted us to isolate the needed signals on the two flight computation boards. Further, we were able to intercept most of these signals at the interface of these boards to the AST backpanel. The core of our interface became an "extender" card that is inserted directly between the four AST printed circuit boards and the AST backplane. Those signals, which were not directly accessable along this connection, were made at the most convenient and practical location on the board. Throughout this process we have attempted to minimize any direct intervension on our part to the AST circuitry. Switches have been installed for the Simulator Computer input (through the D/A hardware) thus permitting operation of the flight panel in either a "remote" or "local" mode through activation of these switches. Those signals which were only sampled (i.e., rudder, throttle, flaps, etc.) were realized as shunts. The A/D and D/A interface has been constructed as a moduler and highly flexible system which would assist in allowing for future modifications as well as for testing and maintenance.

The error correction interface developed is based upon a pulse-duration-modulation (PDM) signal decoding circuit which can be multi-plexed under computer control to read up to five motor-driven instruments (i.e. vertical speed) and will produce a 12 bit integer value corresponding to the actual position. The choice of this type of circuit has resulted from the need to read the true position of a motor-driven indicator on the flight panel, whose value is input to the Simulator Computer via the A/D channel. The advantages offerred by an interface consisting of such a circuit is that any motor signals on the AST panel may be easily interfaced (i.e., altimeter, directional gyro, bank angle, YOR/OBS 1, 2). By supplementing the PDM circuit with an additional circuit for multi-plexing the PDM feedback circuit with the control panel switches we have also eliminated the need for any new I/O computer hardware. This scheme is considered to be practical since the software background uses the control panel and the sofeware foreground uses the error correction feedback.

We have also begun to write the software drivers to accompany the error correction interface to complete compatability with the existing Simulator Computer software.

With the two interfaces installed we have begun calibration of the AST flight panel which is done to relate hardware and software.

The calibrations are used to derive scaling coefficients which convert the MKS panel instrument indications through dimensionless A/D and D/A binary computer registers to internal (software) SI units. The calibrations are performed via the calibration program created during Phase 3 known as CALIB.

Work remaining for the integration process may be briefly listed as, (1) completion of calibration procedure, including preparation of calibration data files used later in support of FLTMAC on-line simulator program, (2) re-coding FLTMAC flight dynamic module with new control scheme adopted for error correction of motor position and using updated A/D-D/A list, and (3) validation of on-line simulator software for functional operation.

DISCUSSION

Functional Description of Flight Panel

Development of the Multi-man System during this fourth phase (September 1979-September 1980) has focused upon integrating the Aviation Simulation Technology Corporation (AST) Model 300 Flight Simulator into the system as a replacement for the PACER MK II Flight Instrument Trainer.

This section consists of a more detailed discussion of this period's work than was covered in the previous Background section. Material here includes a brief account of the functional organization of the AST panel and its motor control scheme, which is information concerned with the two interfaces developed as the main elements of the integration process.

Work for this period will be more easily understood if the reader is first made familiar with two aspects of the AST Flight Simulator design. The two aspects are concerned with the two interfaces designed during the integration process; one interface between the AST flight panel and the existing A/D and D/A Simulator Computer interface, the other an interface designed to supply signals from the AST panel to the Simulator Computer to be used in producing error correction signals to the flight panel's motor-driven instruments during "remote" operation of the flight simulator.

The functional aspects of the AST Flight Simulator relevant to the two interfaces developed may be briefly outlined in terms of how the engine and flight characteristics are computed, the principal electronics functions which support panel operation and how they relate to component organization of the AST panel and the motor control scheme employed.

The engine characteristics are computed from data gathered from a throttle quadrant and the engine logic circuitry and computes from that data the power generated by the various engines and the differential power of an engine imbalance. This engine characteristics' circuitry utilizes analog input signals from the throttle's and tachometer's input controls. From these signals the various states of the engines are determined. Output of this circuitry appears in terms of signals on the manifold pressure and tachometer indicators and the engine instrument cluster.

The flight characteristics are computed by a hybrid analog and pulse duration circuit. Inputs to this circuit are the yoke and rudder signals and the engine power signal. The circuitry full flight characteristics and its output drives the various indicators such as airspeed, vertical speed, pitch, turn and slip. This circuitry also generates analog signals proportional to airspeed change rate, turn rate, altitude change rate, and roll rate. The signal's input to this circuitry are pulse duration modulated signals, and navigation information produced in the onboard INTEL 8085 used in support of the panel's navigation functions.

There are three remaining electronic circuits which comprise the basic functional anatomy of the flight simulator; a master four phase oscillator, a master phase locked oscillator and a rate to phase converter. These circuits are responsible for several functions for the panel but are only discussed here because of their direct bearing on motor control of the panel's instruments.

The four phase master oscillator generates four 200 Hz signals, each one phased 90 degrees from the preceeding one. These signals are used to generate the reference signals for the resolvers used on the various motor-driven instruments, and the OBS of the two navigation indicators. The indicators which utilize four phase signals as reference include the compass, the heading indicator, the bank portion of the attitude indicator, and the ADF indicator. The two navigation indicators utilize "four wire potentiometers" to generate phase signals from the OBS dials.

The master locked oscillator is responsible for locking a signal 12 bits higher than the master four phase oscillator to the four phase oscillator. This signal, therefore, generates a square wave of exactly the same phase and frequency as the master four phase oscillator discussed above. The output of this circuitry is then phase locked to the master four phase analog oscillator. This signal is used in all rates to phase converters and the analog-to-digital converters.

There are four rates to phase converters in the flight panel. These converters convert an analog signal into a rate of change of phase speed. The phase of this signal is relative to the master phase locked oscillator. As an example consider: a square pulse signal of 200 Hz represents the heading of the "aircraft". The heading in degrees is proportional to the difference in degrees between the master phase locked oscillator and the output of the heading phase signal. In order to turn the "aircraft", an input is generated in analog form proportional to the turn rate of the aircraft. If, for example, the analog signal was proportional to a standard rate to turn, then the phase of the output of the heading phase converter with respect to the master phase locked oscillator, would vary by 180 degrees per minute. Such that if the master phase locked oscillator and the output of heading phase converter were viewed on an occiliscope, the phase of the phase converter would vary by 180 degrees in a period of one minute.

There are also several phase locked loop (PLL) circuits used in the flight panel for comparison of the various indicators to the outputs of the phase converters. Such that if the heading of the "aircraft" was 45 degrees, the output of the compass phase locked circuit will be proportional to the difference in phase between the heading phase converter and actual position of the compass. The output of the PLL will generate an analog signal which will be used to drive the motor of the compass. Therefore, these PLL circuits are designed to maintain the actual position of a motor-driven indicator to match the output of the rate-to-phase converters.

This constitutes the motor control scheme of the AST flight panel, which is also employed on other indicators besides the compass, such as the heading indicator, and the roll portion of the attitude indicator.

Simulator Computer/AST Panel Interface

The A/D and D/A hardware of the Simulator Computer/AST Panel interface may be more easily explained and visualized if a brief component organization of the AST Panel is given first.

The AST Flight Simulator is organized as four major printed circuit boards which provide the main computational logic utilized by the panel. There are two boards which are responsible primarily for generating the flight characteristics (FC-1, FC-2) and two for navigation functions (NC-1, NC-2). These four boards connect via 112-144 pin edge connectors to a back panel where signals are routed to the different functional groups of the four computational boards as well as the panel instruments and power supplies.

The first flight computation board, FC-1, performs the computations for engine and flight characteristics, and contains the master four phase oscillator. The second flight computation board, FC-2, contains the rate-to-phase converters, the phase locked loop circuits of the instruments, the altimeter phase locked loop, the AC comparators for the OBS and ADF feedback, and the master phase locked oscillator. The first navigation computation board, NC-1, houses the micro-processor (INTEL 8085), it also contains inputs for the various front panel switches on the simulator, and twelve 8 bit D/A converts. The fourth board, NC-2, is the second navigation board and it contains the memory elements for the micro-processor on NC-1.

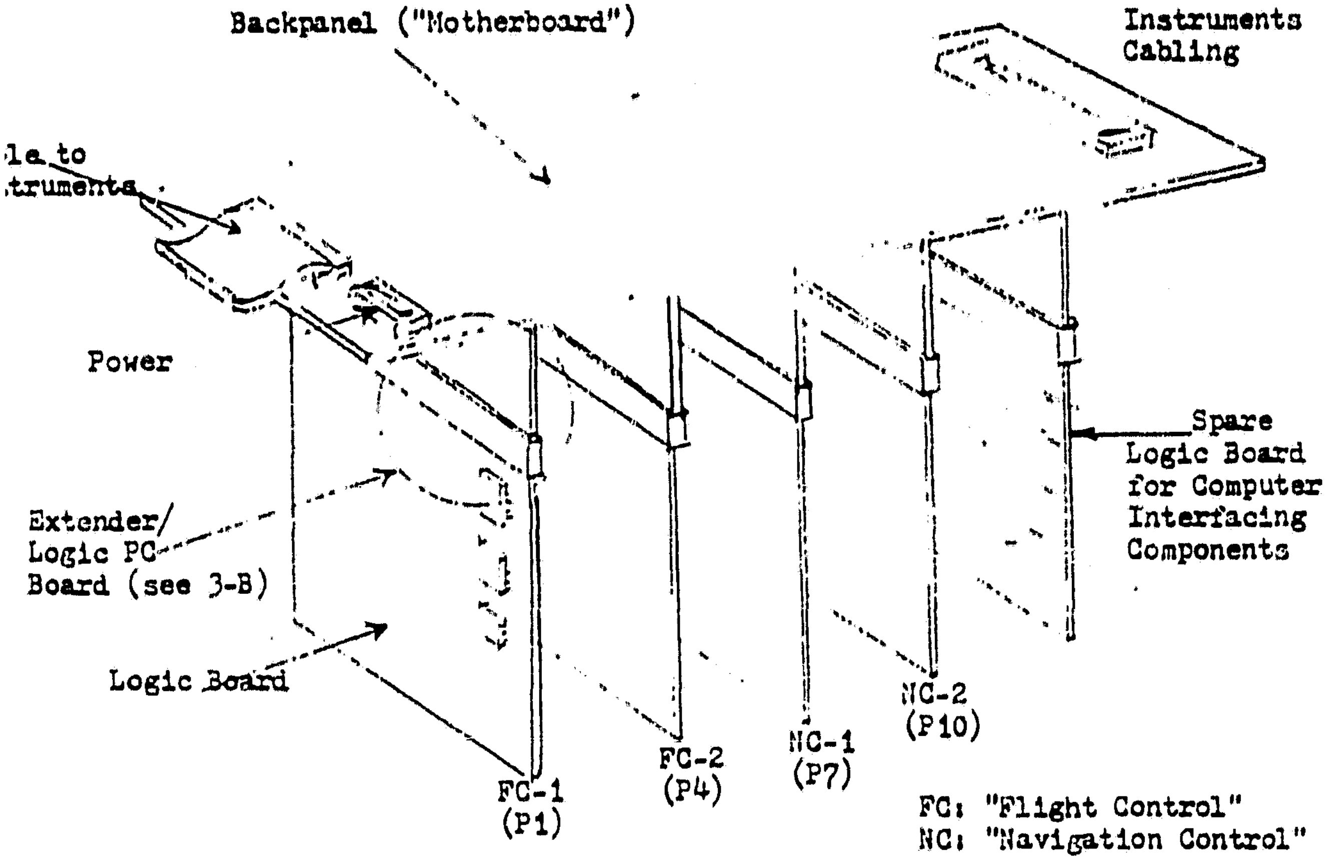
The Simulator Computer/AST interface was accomplished exclusively through use of the existing A/D and D/A channels previously used with the PACER panel. The same channel assignments of the converts had been maintained which is an advantage for the software integration. The initial problem encountered in the design of the interface was to locate the necessary signals within the AST circuitry (see Table 1). Through careful study of the AST schematics and physical panel suitable forms of the various input/output control signals were identified. There were three major considerations which guided the design of the interface: first, that its basic function was to permit access to the control signals (i.e., elevator, ailerons, throttle, airspeed, etc.), second, that the design should accommodate switches for operation of the flight panel in either "remote" or "local" mode such that computer generated signals could be switched out or into the AST circuitry, and third, that the design permit ease in testing and maintenance. Through careful study of the AST schematics and physical panel suitable forms of the various input/output control signals were identified and isolated[10]. Most of all the signals were available on either FC-1 or FC-2. Further, most of these signals could be intercepted at the interface of the boards to the AST backpanel. The core of the interface design, therefore, became an "extender" card that is inserted directly between the four AST computational boards and the backpanel (see Fig. 8). Those

TABLE 1

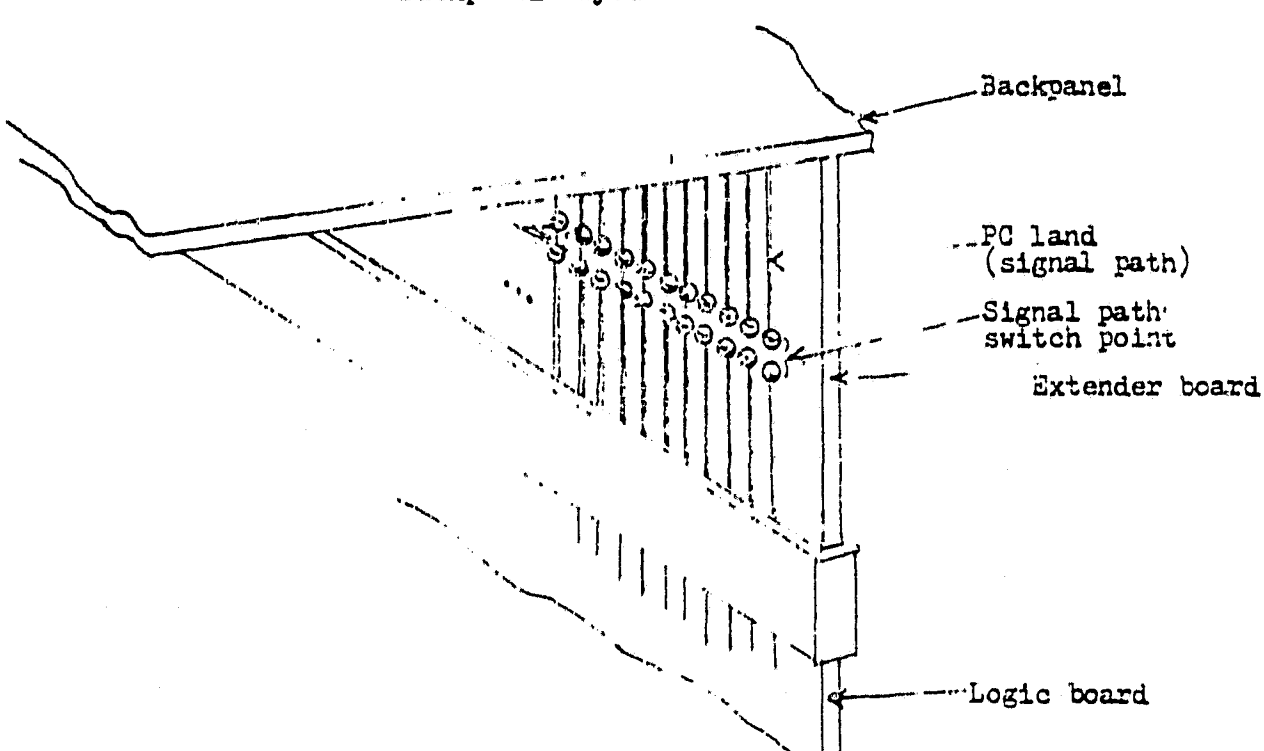
INPUT/OUTPUT CONTROL SIGNALS

Input Signals		Output Signals	
A/D	Function	D/A	Function
0	Elevator	0	Pred. Horizontal
1	Aileron	1	Pred. Vertical
2	Rudder	2	ILS Localizer
3	Throttle	3	ILS Glide Slope
4	Flaps	4	Directional Gyro (Turn Rate)
5	Landing Gear	5	Pitch
6	Altitude	6	Artifical Horizon (Roll Rate)
7	Atitude	7	Altitude
8	Heading	8	Airspeed
		9	Vertical Speed

ORIGINAL PAGE IS OF POOR QUALITY



Backpanel Layout



Expanded View of Extender/Logic PC Board

Fig. 8: AST Panel Instrument/Cabling Scheme

signals which were not directly available along these connections were made at the most convenient and practical location on that board. Signals to be supplied during "remote" mode were made available through switches, such that when "on" the signal would be supplied by the Simulator Computer via the D/A interface, and when in an "off" position the signal would be maintained as the AST signal. For those control signals which are only sampled, such as rudder, flaps, landing gear, etc., they have been realized as shunts which are input to the simulator computer via the A/D channels. The various elements comprising the interface as shown in Figure 9 are connected by ribbon cable to the system back plane. The system backplane consists of several mass terminated connectors. At this panel the various signals coming off the different interface elements to the panel are arranged into proper form for the RS232 connectors of the A/D and D/A of the simulator computer (refer to Figure 10). At a later time, and especially if the simulator is duplicated, serious consideration should be given to realizing the system backplane as a printed circuit with built-on RS232 connectors due to the unavoidable chaos that is associated with using terminal strips to manage such a large number of connections. Therefore, the resulting interface has been constructed as a modular and highly flexible system which would assist in allowing for future modifications as well as for testing and maintenance.

Error Correction Interface

This interface is used for motor position control of the motor-driven instruments on the AST flight panel when being operated under "remote" mode. The interface as a whole is composed of both a software "filter" and an associated circuit. The following discussion consists of two parts;

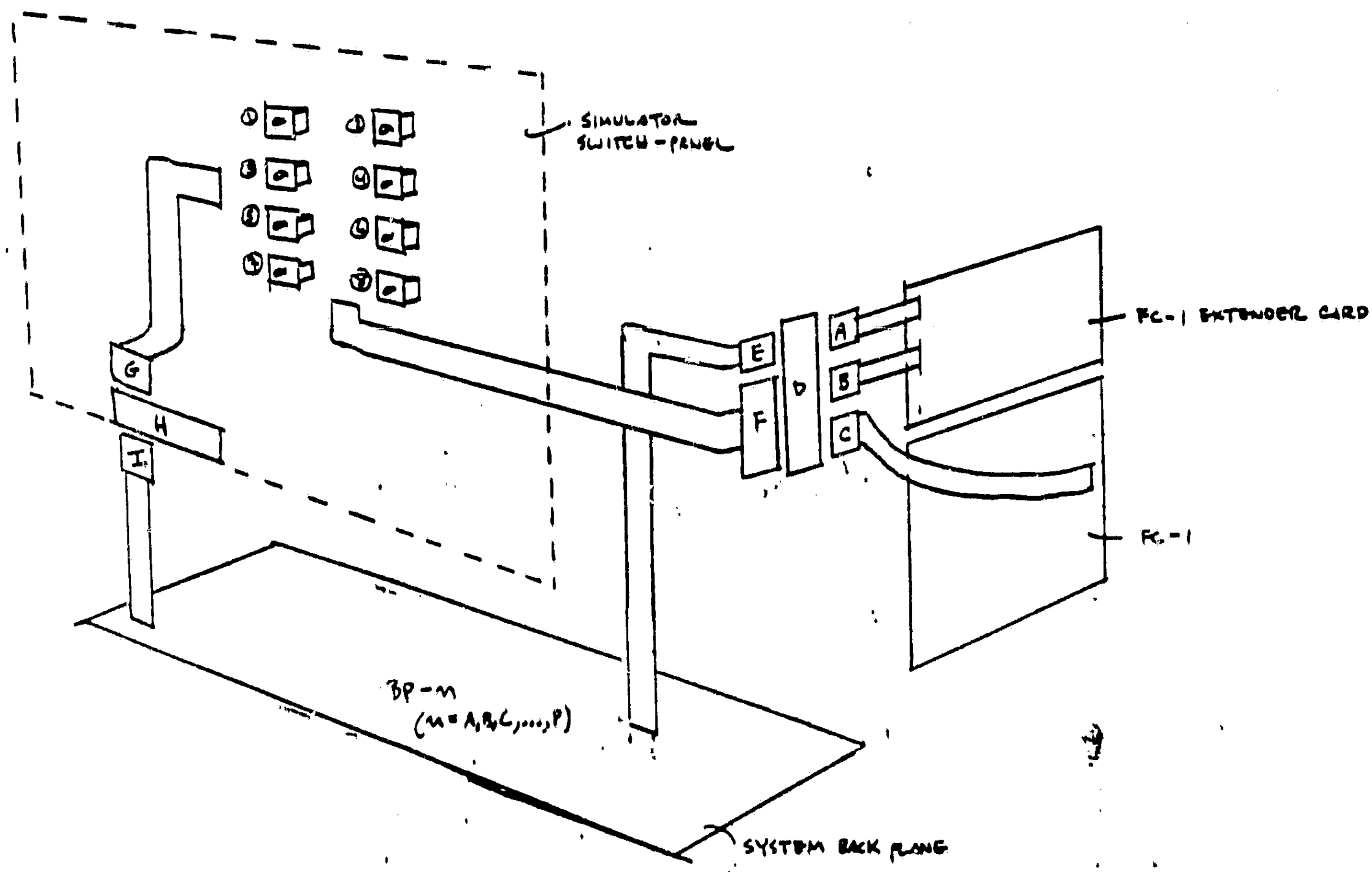


Fig. 9: AST/Simulator Computer A/D and D/A Interface

ORIGINAL PAGE IS
OF POOR QUALITY

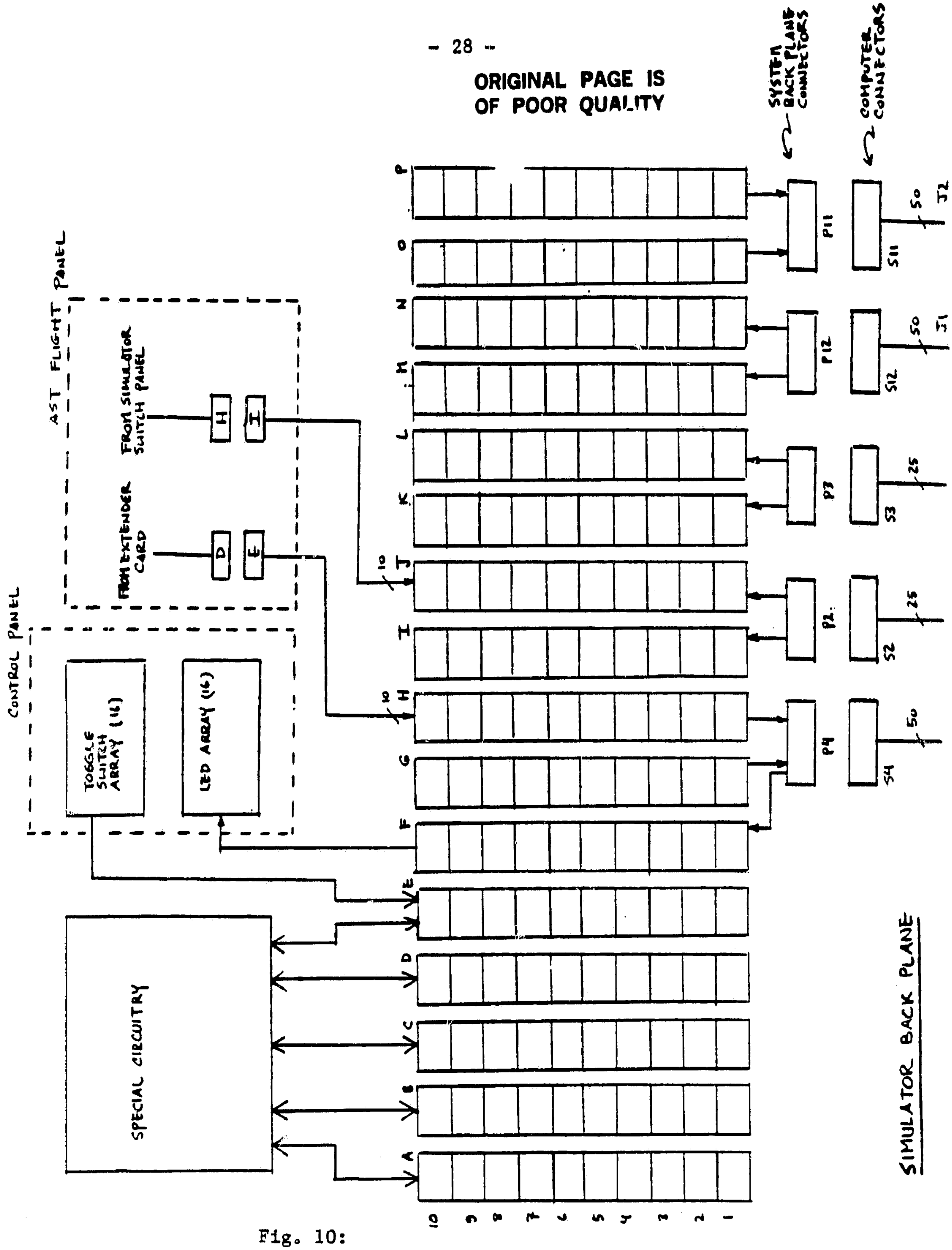


Fig. 10:

first a brief explanation of the motor control scheme employed, which will include an outline of proposed "filters" and the second part, a discussion of the circuit designed to implement the hardware portion of the scheme.

The control scheme for motor loops in the AST panel is proposed as follows (see Fig. 11)[6].

1. Software obtains control inputs, calculates rate and position, outputting rate to the indicating meter/motor circuit in the panel;

2. Software calculates an error value for position, based on feedback from the panel. This value is passed through a software "filter", which corrects the RATE output; the rate output is still used to drive the motor/meter circuit with a single signal.

The "filter" to be used should adjust the rate output minimally. Since the rate indicating meters are used by the pilot for aircraft control, small deviations from "actual" should be tolerated; a deviation of approximately less than 10 percent should be used. Several possibilities for filtering may be acceptable. These are, in order of expected implementation:

1. "Chopper": The error-correction value, initially in units of position (radians) is divided by the realtime period to derive a rate, which if maintained for the next realtime period, would correct the motor position. If the error-rate exceeds the calculated control-input-based rate, then the error-rate is output; otherwise, the calculated rate is output. This method would have the effect of causing rate output beyond the time the pilot is activating inputs to alter position; it would also have the effect of periodically "boosting" rate output during a control impulse period based on the error rate. This method has the advantage of being simple to implement and has the disadvantage of possibly causing faulty rate indication.

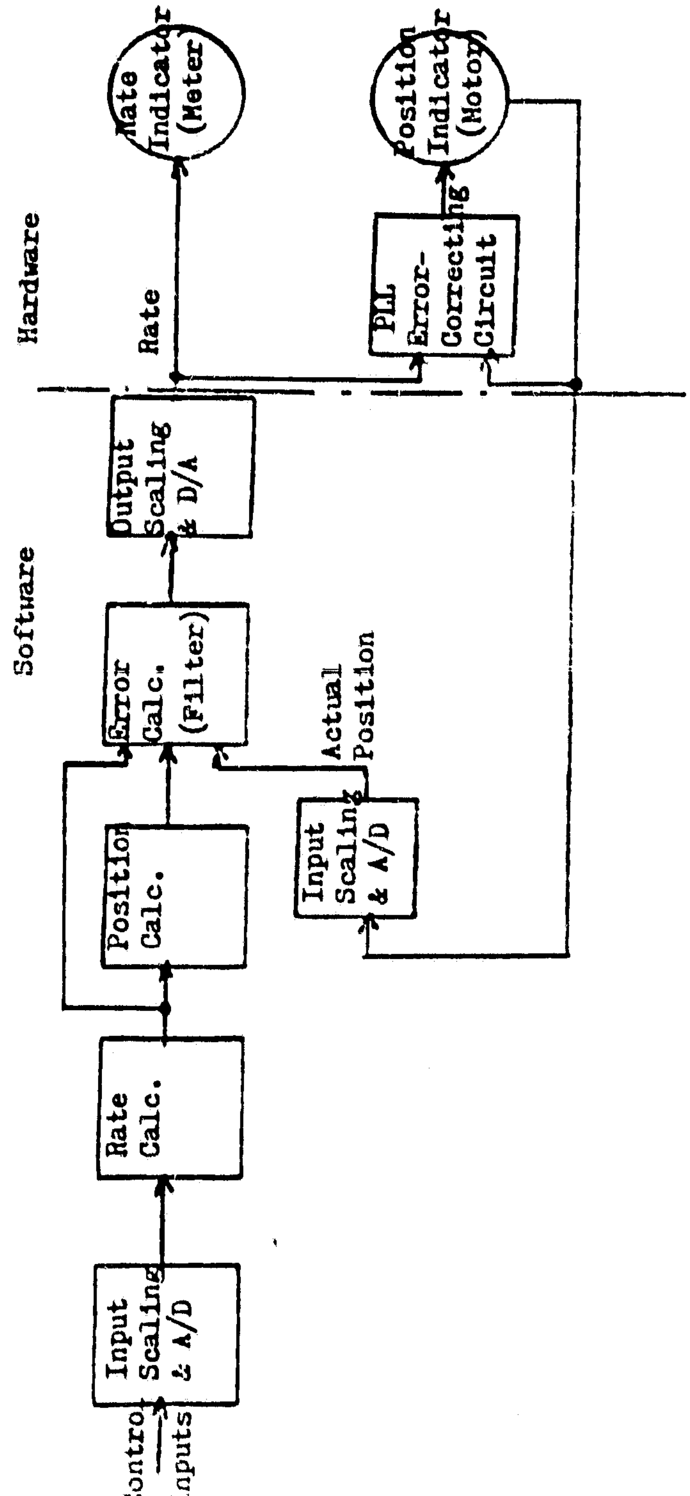


Fig. 11: AST-Remote Computer Interface

2. "Dynamic calibration": The error-correction value would actually appear as an adjustment to the scaling factors used for D/A output of the rate. The error-correction base would have to be the total integration of calculated "delta positions" maintained by software; the actual total integration of motor position changes over time would be divided by the calculated integration value, yielding a normalized value which would be used to alter the output scaling coefficients in the meter/motor loop in software. This method would have the effect of correcting the rate output to values which cause the calculated position changes to result. It is noted that some indication error in the rate indicating meter would still result, but this may be acceptable if it is assumed that most of the error buildup between software and hardware results from dynamic changes in signal/response characteristics in the hardware. Non-linearities would have a strong effect on this method and should be carefully analyzed. Also, deviations in total integration would not necessarily be corrected by this method; some additional position-correcting scheme would have to be used. Implementation of this method at a minimum would require software calculations of total calculated and total actual position change as well as periodic alteration of the output scaling coefficients.

3. "Bang/Bang Pulse": In this method, a periodic, small-duration maximum/minimum pulse would be sent through the output circuit in order to cause a large change in the motor position while causing an (hopefully) imperceptible change in rate (meter) indication. It is thought that a 100 msec high-magnitude signal to the rate indicator will not be perceptible, but will cause a definite position change in the motor. This method might work well with the AST hardware since integrators are used in the PPL-correcting

circuit to cause reference position to change. This method might be used with method (2) to cause accrued position errors to be eliminated. As an independent control scheme, though, there is the disadvantage that meter needle "vibration" may result.

4. "Meter-Motor Decoupling": This method would be the same as used in the present panel, shown in Figure 11 and discussed above. In all, 6-10 D/As would be needed (two each for altimeter, gyro, artificial horizon, VOR 1, VOR 2), which would require an additional D/A module in the computer at a cost of $500 (ADAC)[6] to $900 (DEC)[7]. Another disadvantage is that the in-hardware PLL-loops would not be used; since this is a primary enhancement of the AST over the PACER panel, it is questionable whether the AST panel should be used in this manner.

As mentioned above, the basic motor control scheme for the AST/Remote operation requires some circuitry for processing the phase-encoded motor position information. The circuit designed during this period to perform that function is based in part on the technique employed in the AST panel itself. (see Fig. 12). The AST technique consists of a data counter onboard the AST panel's 8085 is used to count up transitions of a master timing signal; when the phase-encoding counter carries, signifying the phase-encoded reference "position" of the motor maintained in hardware, an interrupt occurs to the micro-computer and the data counter is read and scaled into floating point representing the motor hardware-maintained "position". One count of the data counter corresponds to a scaled unit of deviation of the motor zero reference point from reference zero. The alternate proposed scheme for the Simulator Computer (LSI-11) would require the addition of:

1. a data counter and controlling circuitry (strobe, reset, data latches/gates);

2. a parallel Input/Output logic board on the LSI-11 (DRV-11 compatible)

   for:

   a. addressing of the AST multi-plexor to select motor feedback channel,

   b. data lines to read the data counter,

   c. control lines and interrupt lines for circuit control;

3. a software driver to select, read, and scale the counter.

(In addition, a time-base synchronization between AST panel and LSI-11 computer may be needed, since the start of the hardware phase period may be out of synchronization with the start of the LSI-11's realtime/linetime clock period.)

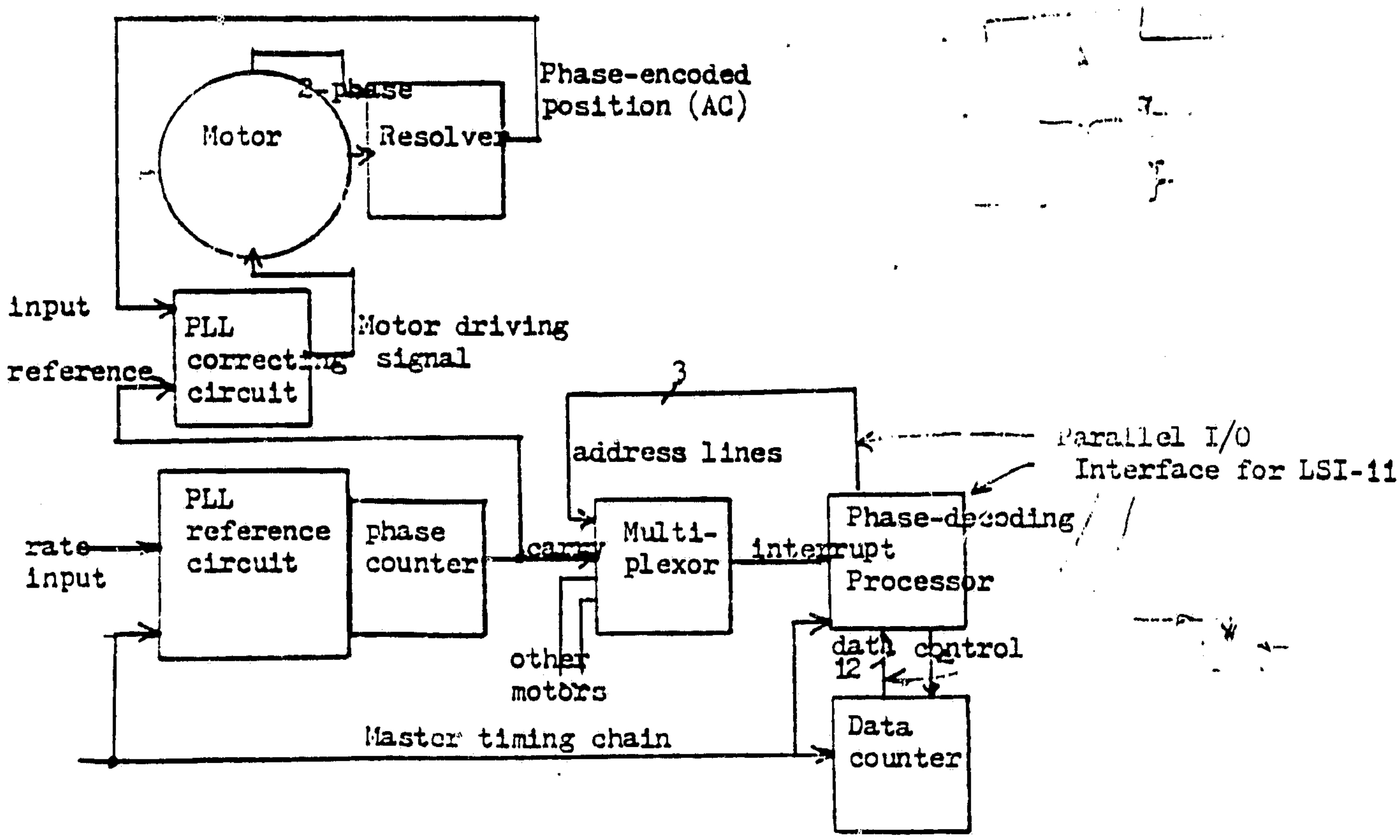


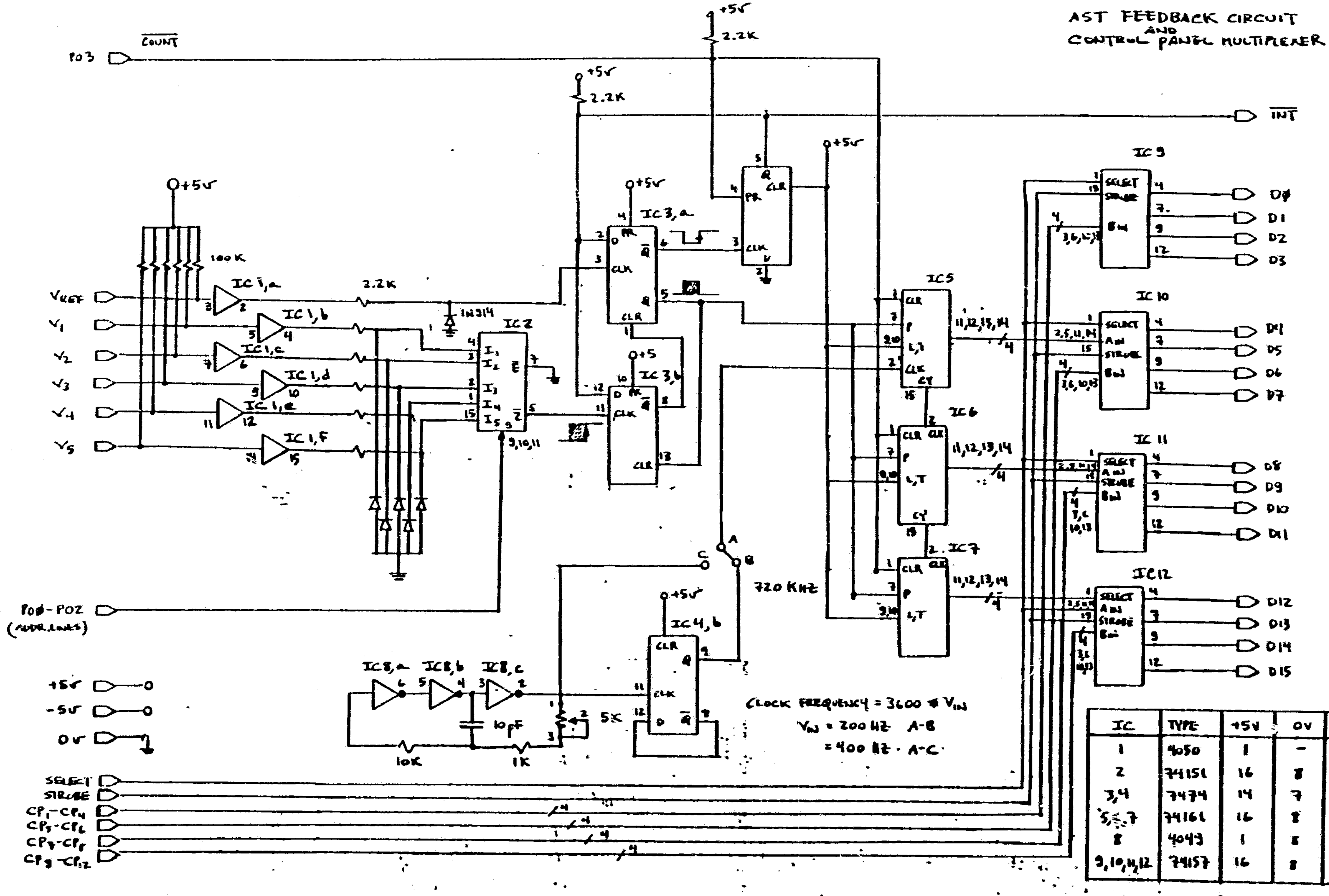
Fig. 12: Pulse-Encoded Motor Feedback Scheme

Based upon the hardware needs of the alternate control scheme, a pulse-duration modulation (PDM) signal decoding circuit was developed. The circuit has been designed to permit multi-plexing under computer control to read up to five motor-driven instruments (i.e., vertical speed) and will produce a 12 bit integer value corresponding to the actual motor position. This feedback scheme for the PDM-encoded signals requires four discrete outputs (one command, three address) plus 13 discrete inputs (one interrupt, 12 data). In an effort to work with the existing Multi-man System hardware, an additional circuit was constructed which will permit the error correcting circuit to share one of the parallel I/O logic boards (DRV 11) of the Simulator Computer[11]. The error correction circuit will be multi-plexed with the control panel, where the control panel is a set of switches used to transmit pilot commands to software. The multi-plexing scheme is considered practical since the software background uses the control panel and the software foreground will use the error correction feedback circuitry. Figure 13 is a schematic of the combined circuits, the PDM-error correction circuit and the multi-plexer circuit for data output control of either error correction output or control panel outputs.

The circuitry described above as of this writing has only been tested off-line due to the need for completing AST/Simulator Computer interface which permits closed-loop operation of flight panel under "remote" mode. This circuit has been located in the "fifth" board's position in the AST card cage, as shown in Figure 9. The circuit is connected directly to the system backplane where both circuit inputs and outputs are obtained.

Development of the software drivers for operating the circuit is also currently underway.

AST FEEDBACK CIRCUIT AND CONTROL PANEL MULTIPLEXER



IC	TYPE	+5V	0V	-5V
1	4050	1	–	8
2	74151	16	8	–
3,4	7474	14	7	–
5,6,7	74161	16	8	–
8	4049	1	8	–
9,10,11,12	74157	16	8	–

JUNE 1980

Calibration of Flight Panel

The final aspect of this period's work concerns one of our other present activities -- calibration of the AST panel's instruments via the interfaces discussed. The calibration procedures is used to relate hardware and software. This is accomplished through a program developed during the previous period,[6] CALIB. Calibrations are used to derive scaling coefficients which convert the MKS panel's instrument indications through the dimensionless A/D and D/A binary computer registers to internal (software) SI units. Scaling coefficients determined for the various flight instruments are then maintained in a database which can be transmitted to the Host for disk storage in support of an on-line simulator program, FLTMAC.

Completion of Integration Process

Below is a brief outline of the work remaining in the integration of the AST Flight Simulator into the Multi-man System.

1. Completion of calibration procedure, including preparation of calibration data files used later in support of FLTMAC on-line simulator program.

2. Re-coding FLTMAC flight dynamic module with new control scheme adopted for error correction of motor position and using updated A/D-D/A list.

3. Validation of on-line simulator software for functional operation.

FOOTNOTES

[1] Kreifeldt, J. G., Economical Design of a Programmable Multi-Manned Flight Simulator Facility. Tufts University, Medford, MA.

[2] Aviation Simulation Technology, Bedford, MA.

[3] Kreifeldt, J., A Study of Pilot-Controller Interaction Under Various Simulated Advanced ATC and Navigation Concepts. Progress Report, 12/01/78-8/31/79, Department of Engineering Design, Tufts University, Medford, MA, September, 1979.

[4] Progress Report: Multi-man Flight Simulator System, for the period September, 1977-March, 1978. J. Wittenber.

[5] "PDP-11" and "LSI-11" are trademarkes of the Digital Equipment Corporation and are used with its permission; "PACER MK II" is a trademark of the PACER Corporation and is used with its permission.

[6] Progress Report: Multi-man Flight Simulator System, for the period March, 1978-October, 1978. J. Wittenber.

[7] RT-11, Version 3, Revision 1B.

[8] RT-11, FORTRAN IV, Version 2, Revision 1.

[9] Used with permission of the Aviation Simulation Technology Corporation.

[10] It should be noted that this process was slightly encumbered initially due to incorrect schematics supplies by AST and an overt effort on our part to minimize any direct intervension or modification to the AST circuitry.

[11] Recommendation for AST Panel Feedback. Internal Memo, J. Wittenber, March 25, 1980.

APPENDIX 1

Terminology

The following terminology is listed in order to prevent confusion:

Uplink:	Data transmission in the direction of computer "central" to flight vehicle, e.g. from Mainframe to Simulator. This definition corresponds to normal ATC terminology, although in computer science terms uplink normally refers to direction from lower-level network element to higher-level network element.
Downlink:	Data transmission in the direction of flight vehicle to "central", e.g. from Simulator to Mainframe. This definition is the converse of Uplink; see above.
Link:	A logical data communications path; this definition implies data linkage without respect to the physical medium used for transmission (e.g. line).
Line:	A physical data communications path; this definition implies the possibility of more than one physical "wire" connecting two computers (e.g. parallel versus serial).
Mainframe:	The highest-order, or "central" computer system entity in the network; this entity may be comprised of several computers.
Host:	The Multi-man System data-concentrator/program development subsystem computer-interface.
Simulator:	The "standalone" flight simulator, including computer and flight instrument panel.
Parallel, asynchronous:	Data communications medium in which data are transferred a word at a time and in which CPU instructions need to be executed iteratively for transfer of each word. This medium may be "driven" by an interrupt handler which is accessed once per word or once per message; the former is referred to as "word" mode, and the latter is referred to as "burst" mode.
Alternate:	Data communications message transfer method in which two-way data traffic is handled in one direction at a time. See "simultaneous". Default method for Multi-man System.
Simultaneous:	Data communications message transfer method in which two-way data traffic may operate at the same time. See "alternate".

Flight Instrument:	A subsystem of the Simulator which comprises the "cockpit" panel.
Remote Mode:	A Simulator which is being controlled by the Host via data communications message commands. Also, a Flight Instrument which is being driven by a Simulator computer.
Local Mode:	A Simulator which is being operated without control by the Host. Also, a Flight Instrument which is being operated without control by the Simulator computer.
On-line:	Being used to conduct experiments.
Off-line:	Not being used to conduct experiments. Off-line is comprised of two modes: Maintenance, i.e. calibration of hardware maintenance; or Training, i.e. being used as a standalone flight simulator without being accessed by the Host except for program memory download.