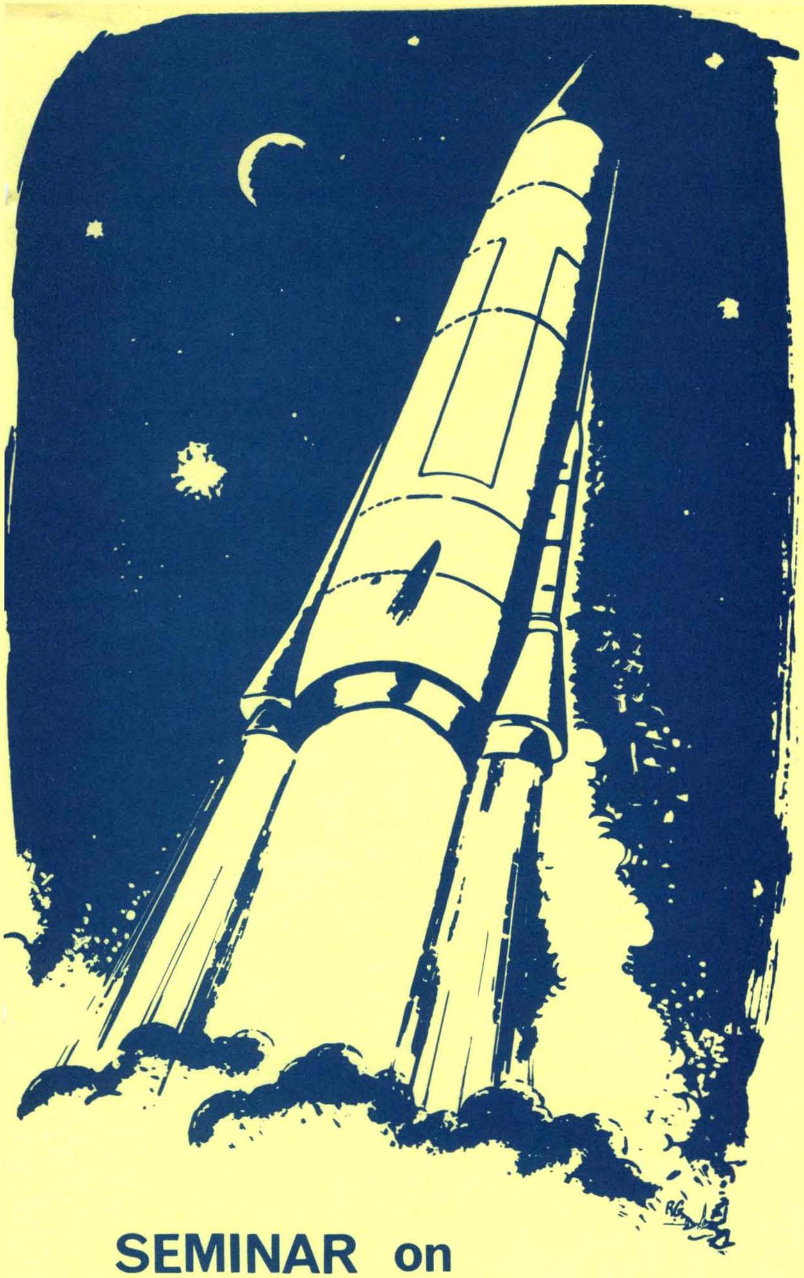


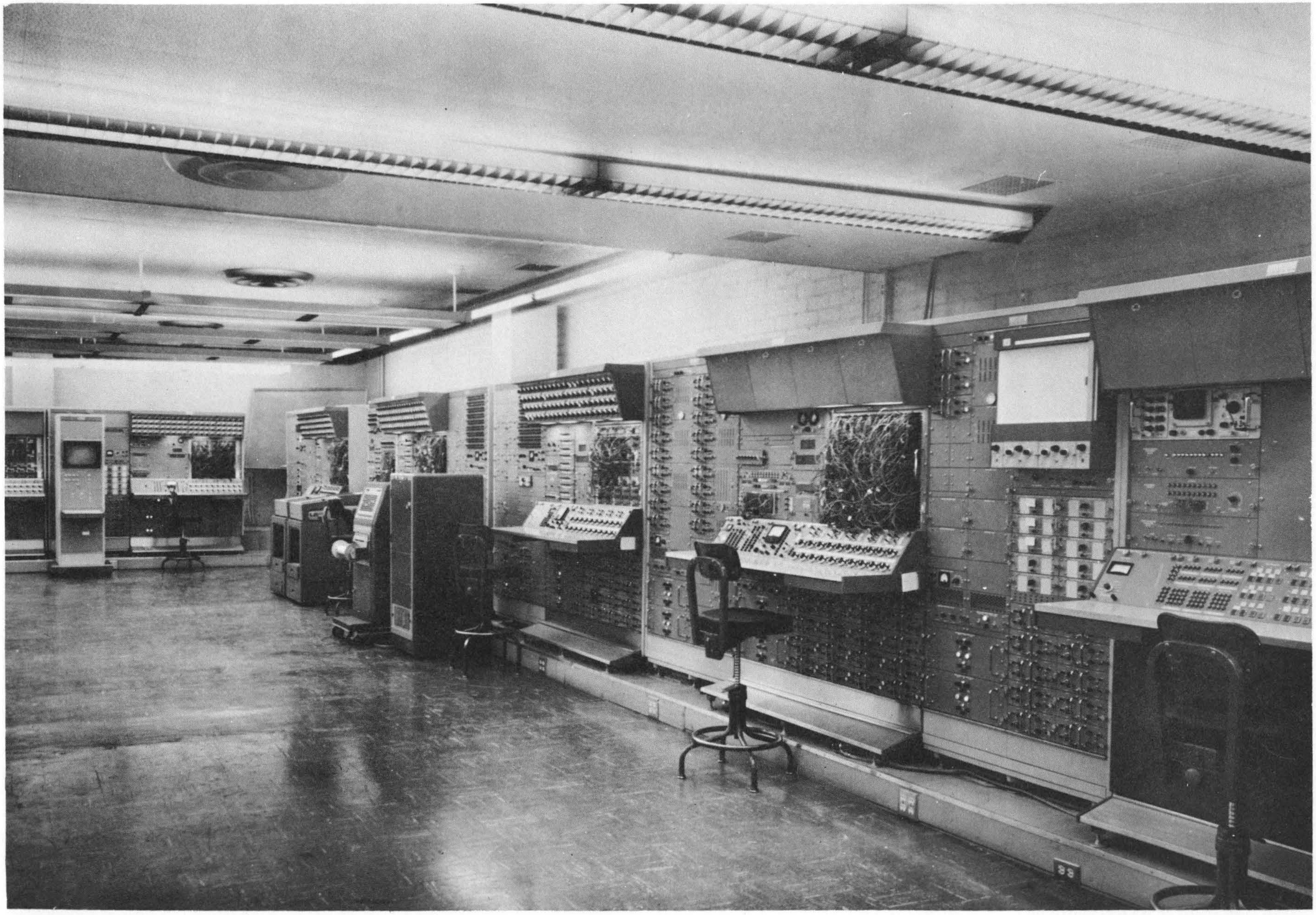
SEMINAR on HYBRID COMPUTATION as applied to the AEROSPACE FIELD

EAI®
ELECTRONIC ASSOCIATES, INC.

SEMINAR ON HYBRID COMPUTATION

TABLE OF CONTENTS


EAI
COMPUTATION CENTER
ELECTRONIC ASSOCIATES INC.



COMPUTERS IN THE AEROSPACE CORPORATION

I. Introduction

In the modern aerospace corporation, computers are used widely in the scientific, engineering and other technical operations. Probably more so than in any other industry, they have been accepted as key contributors in the equipment research, design and development activities and beyond these they play increasing roles in manufacturing, test and check-out, and field operations. Computers are now so commonplace in the typical company that in contrast to the situation 15 years ago when one of the pioneers of the industry thought that a few computers spread around the country would satisfy all possible needs, one wonders whether a few computers spread around a company will satisfy the needs within that company. With them playing a dominant role in the technical activities it is useful to review their functions and clarify by reason of these functions, the qualities to be sought within the computer and the manner of its use in research, design, and manufacturing. Rather than having a pre-disposition to this computer or that computer we shall look at general purpose computation and by relating the desired result with the possible computer mechanization derive a look ahead to the future aerospace computer department.

Technological breaks-through in the components of computers are not expected within the next decade. Solid state circuitry is commonplace and wherever it provides the desirable performance it is presently in use. High speed memory devices in digital computers and widebandwidth computing components in analog computers, equal to the tasks they perform are available in every manufacturers equipment. The improvements to be anticipated are in man-machine communication and computer organization, the first to improve the ease and flexibility with which the computer is applied, the second to increase computing capability and value. Both improvements require detailed attention to be paid to the way computers are used in solving useful problems, that is to say we must look to where, why and in what general manner rather than to the detail of individual programming. Fortunately, there is a consistency of methods and procedures in the many aerospace companies that allows us to do this. We can categorize and classify, and this procedure is very useful and informative.

II. Aerospace Systems Design and Development

Looking at the progressive development of a system in an aerospace company, be it a shoulder-launched, rocket-propelled, infra-red homing missile for use against low-flying aircraft, a super-sonic variable wing, intercontinental airliner, or a manned orbiting space station intended for one year's successful operation, the ste-by-step procedures are similar. Perhaps slight variations enter and possibly there are different useful arrangements of the list but the eight steps to successful operation of the system are:

1. Conceptual Studies aimed at sketching out the overall performance to be attempted, the state-of-the-art components, the organization and integration of

the system, the major difficulties to be attacked and solved, the alternative ideas that can usefully be investigated when all of the way-out possibilities have been legitimately discarded. Conceptual studies are performed by men with ideas, researchers and designers both, but those very much interested in the future rather than the present. They produce typically a paper proposal of a system never before built, not able to be built today, but certainly by projecting today's laboratory experimental device into tomorrow's mass-produced item, something practical in the future. Until one begins to build it, the limitations of pencil and paper must be supplanted by the almost unlimited boundaries of the computer to assist in the testing of ideas.

2. Experiments to establish basic physical data. Typically a projected aerospace system is touching the bounds of the unknown at several points. The strength of alloy materials at high temperatures, the effects of fatigue on plastic materials, the structural integrity of a thin variable plan-form wing under gust loading, the sloshing in a liquid-fuel rocket and its effect on vibration and thrust control, the aerodynamic forces at high Mach number, the expansion of a turbine blade at high speed and temperature, the spectrum of visible and invisible light that is useful for guidance, the effects of cross-coupling non-linearities on attitude control, etc. etc.. Basic experiments must be performed to establish the data for use in the design of a total system and these experiment's both in their planning, their efficient execution, and in the appropriate reduction of the measured data to a more usable form benefit from the use of computers.

3. Component Design. A total system is the integration of a set of subsystems made of components. From the projected performance of the system one places requirements on the behavior of the sub-systems and thereby the components. A first step in the practical realization of the system must therefore be the design of components; gyros, accelerometers, hysteresis motors, clutches, synchros, gear trains for the homing system of a missile; pumps, turbines, tachos, variable ducts, combustion chambers, fuel injectors, for the jet engines of a supersonic transport; thermocouples, humidity devices, gas analyzers, air conditions, air purifiers and all of the other components of an environmental control system for a manned orbiting space station. In some instances, component design may not be necessary for many components will be available from other systems but wherever it is necessary, the modern computer has a part to play in establishing desirable parameter values for optimum performance according to recognized criteria.

4. Sub-system Performance Analysis. The assembly of components into a subsystem (a propulsion system, an attitude control system, a guidance system, a structural system, etc.) always introduces the possibility of poor interaction between the components. This is particular true where energy feedback loops exist, permitting dynamic instabilities and completely unacceptable behavior. The individual components can be treated as "black boxes" having transfer relationships between input stimuli and output signals, as established in the component design phase of the program, and their assembly into a working sub-system is thereby amenable to overall performance analysis, usually making use of a computer.

5. System Integration and Hardware Check-out. As the system design moves along and pieces start falling into place in the form of real hardware, the dynamic

behavior of the total system, made up as it is from the many sub-systems, comes into focus. The test of overall performance before all parts are available is a standard procedure to avoid a "back-to-the-drawing-board" situation when the first prototype system "flies". Computers, of course, play key roles here for theoretical analysis and "seta-of-the-pants" performance conjecture is quite inadequate. It is at this point that the concept of the engineering simulator comes through in big terms, permitting a real-time representation of parts of the system not yet available in real hardware to be used in the exercise of existing hardware. The homing missile control sub-system hardware can be exercised in a realistic target approach situation, the transition phase in the variable wing plan-form can be exercised before flight under control of a simulated auto-pilog, the attitude control system of the manned orbiting space station can be tested just as though it were in space and without the assistance of aerodynamic damping. Beyond this engineering simulator role, computers assist in the solution of many problems in the organization of the total system and its eventual production.

6. Mission Planning. Aerospace systems, be they for research, commercial transportation, or military purposes are always intended to move from one point to another. The missile is launched and approaches the target, the supersonic transport flies from New York to London, the orbiting space station is placed in orbit and then must be re-supplied by satellite vehicles. There is typically a requirement, therefore, to plan these missions so that by some criterion of performance they are completed successfully -- the missile must destroy the target no matter from which point it is launched, the transport must consume minimum fuel and maintain its schedule through a bad-weather landing condition, the supply vehicles must rendez-vous successfully with good energy management. In all cases the point of interest has now moved from the design of the system to its effective exploitation in its projected environment. Analyses of this kind are, to be sure, beyond simple tasks of calculation and are now possible only through the extensive use of computers. Quite naturally, one of the most important missions to be planned is that of the test flight, or normally the many test flights, to ensure effective instrumentation and flight coverage in the proving of the system.

7. Crew Training. Commonplace in the aerospace industry is the concept of training simulators, by which computer models of the designed system and the environment in which the system operates are used to train the staff who are to operate the system. For the all-important reasons of cost and time, crew safety, and instructional convenience, training simulators have become a significant part of the overall aerospace systems' business. Once they used special purpose computers, but now they depend almost exclusively on general purpose equipment frequently equal or similar to that used in the engineering simulator for system integration. The infantryman must fire simulated missiles, the pilot must fly a simulated airplane, and the crew must control a simulated space station because for training purposes it is cheaper, more efficient, safer and convenient.

8. System Check-out, Flight Testing and Evaluation. The final step in the design of any aerospace system is taken by a demonstration of satisfactory performance. The many ranges in this country and others attest to the frequency with which aerospace systems are checked out tested. In every case the large volume of data that flows from tracking and telemetry stations presents a formidable task in computer data reduction and evaluation. Without modern general purpose computers the task could not be completed.

These eight phases of the modern aerospace system design process require computers at every step. The tasks they perform are very varied and it is clear that the demands placed on the computer, the mathematical operations performed, are peculiar to each task. Despite this, the methods and procedures of modern design cause such tasks to be separable into three groups and these groups result computationally in a need for (a) Simple Calculation, (b) Data Reduction and Signal Analysis, or (c) System Simulation.

(a) Simple Calculation. The tasks of scientific research and engineering design have been continually increasing in complexity. However, many times a task requires relatively simple calculations involving algebraic operations and perhaps a few transcondental functions. When the volume of work warrants it, this drudgery is now transferrable to small calculators, readily programmed, and rapid in their operation. They range in size and cost from PDS-1020 computer ($20K) to the IBM 7094 ($3M).

One would expect to find the need for simple calculations throughout the program reviewed above, but particularly in phases 2 and 3.

(b) Data Reduction and Signal Analysis, as the name implies, requires the computer to accept input data on which it acts to provide a refinement, either a reduction in sheer volume with the establishment of important parameters that are equally valuable (mean, standard deviation, correlation coefficients, power spectral density, equivalent system constants, etc.) or a tranformation to a form more useful to the engineer who needs to use the information, or yet again the establishment within the data of some key signal (a natural frequency, a mode shape, a sequence of events, etc.). The task is typified by relatively simple mathematical processing of fairly large volumes of data. It suggests relatively unsophisticated computation but a need for appreciable speedy operation and good machine communications. In conventional hardware, the computer could be either analog or digital but because data reduction and signal processing has typically been an off-line operation with a consequent need to store the information, it is now widely performed with digital computers the signals to be processed being sampled and stored on magnetic tape. It is interesting to speculate on the value of a hybrid computer for such tasks for this has both the data storage capability of digital equipment and the signal filtering qualities of the continuous analog devices.

One might expect to have tasks in data reduction and signal analysis in phases 2, 3, 4, and 8.

(c) Systems Simulation. The major capability provided by modern computers in the activities of aerospace systems design, the one that has changed the procedures of design over the last twenty years, is that of system simulation. The ability to model on a computer the important behavior of any physical system and manipulate this behavior at will by adjusting first the system parameters and then, if not satisfied, the structure of the system until acceptable behavior is obtained is extremely valuable. Before any item is produced, it is thereby possible to "destroy a target", "fly to London", "orbit the earth", at least, the ideas of how to destroy a target, how to fly to London, how to orbit the earth can be tested.

For system simulation purposes, the computer requirements contrast with those of data reduction. Here the need is for relatively sophisiticated mathematical programs acting on very small amounts of input data (system parameters, conditions, and environmental stimuli). Conventionally the task has been one matched by the speed and program sophistication of the parallel analog computer. Analog simulators have "flown" many aircraft and missiles into combat before the prototype left the runway. More recently the increasing speed of digital computers, though painfully slow in comparison, has caused simulation programs to be developed for the more capable of these machines.

One might expect to apply simulation procedures in all phases of a systems design with the possible exception of experiments to establish basic physical data. The capacity required from phase to phase varies and the methods of exploiting simulation change but computer simulation is so universal that it deserves considerable discussion both to place it in perspective and to derive the consequent computer characteristics.

III. Design by Computer Simulation

Question: What is the sequence of the design process at any point of the development of an aerospace system?

The answer to this question, at least for a high proportion of situations, is well demonstrated by a simple block diagram of a form familiar to control systems engineers. The design process begins with a statement of requirements, usually incomplete, sometimes supplied by another agency, department, or section, which establishes a required performance. We can look upon the statement as a block in the diagram and the signal at the output of this block can be viewed as the required performance, P_r.

The signal P_r is applied to the engineer who acts in the form of a comparator with error-detecting qualities. He applies -- a corrective signal, E, which one can look upon as an excitation, to a block which we can describe as creative design. This block, powered by two supplies, knowledge (of the useful physical phenomena) and experience (of the past successful and unsuccessful uses of these physical phonomena) synthesizes a design which can be represented as the information or signal, D, at the output of the block diagram.

Having produced a design, the process works in a feedback manner to test the design. The feedback loop contains a block which we can title analysis for with a proposed design it is possible to analyze performance, P_α, in order to check design performance against required performance, P_r. Any difference results in an excitation, E, stimulating the creative design. Hopefully we have a stable, though high-gain, system and obtain successful results rapidly. Occasionally the process is unable to bring E to zero and adjustment is necessary in required performance, P_r.

In this analogy between the design process and a feedback control system the content of the block termed analysis is informative, if not a little amusing. Three kinds of feedback element are possible:

(1) Mathematical analysis, characterized by an inexpensive two step process of (a) problem description and (b) theoretical evaluation. The problem description is today a wide bandwidth, unity gain, filter able to pass adequately the qualities of a complicated and possibly sophisticated system. The theoretical evaluation, however, is a narrow bandwidth, high order filter unable to handle any high frequency qualities of system sophistication. The feedback element, therefore, provides a rapid but severely limited evaluation of performance, obscuring the sophisticated qualities.

(2) Physical experimentation, characterized by a typically expensive wide bandwidth filter in cascade with an extensive transport delay, passes the qualities of a system for evaluation very well but only after considerable time. This element demands the very low gain in the creative design forward path that existed previous to World War II, for stability to be maintained. The result is very slow progress.

(3) Computer simulation provides an optimum feedback element for it places a short circuit between the more desirable parts of mathematical analysis and physical experimentation by-passing the theoretical evaluation and the transport delay of building experimental equipment. It represents an inexpensive two-step process of mathematical model building and "computer" experimentation which provides for a rapid, high gain design process even when complicated systems are to be designed.

With this introduction simulation as an essential part of modern aerospace design and as the most significant use for the modern computer, we face the question of what kind of computer matches the needs of today and tomorrow. Perhaps one computer designed to fulfill the simulation function will also qualify as a data processor and calculator. Perhaps not.

IV. Computers for Simulation

A computer which is to be used for simulation purposes must provide facility for building a structure which corresponds in space and time (or at least gives the impression outwardly of corresponding in space and time) to the system and environment simulated. It must have the added qualities of permitting a simple building process, a simple injection of desirable stimuli and the effective measurement of consequent behavior, and for the future it must permit the development of automatic design procedures in order to reduce time-consuming "cut-and-try" methods of today.

If one considers the typical use of a design simulator one recognizes three parts to its programming which are essential to the experimental procedure. Firstly, there is a physical system to be represented. This is the subject of design and its representation must allow adjustment of its structure and parameters. Secondly, to understand and thereby evaluate the behavior (performance) of the system when simulated it must be placed within an environment which is also programmed into the simulator. Thirdly, the evaluation process must be controllable by the injection of desirable stimuli with consequent measurement and the possible use of automatic design up-dating.

Although individual examples have particular program demands for these three parts, sometimes containing little environmental program and at other times an extensive one, an example is useful to illustrate this program division. Consider a simulation aimed at designing an automatic landing system for the supersonic transport. It has features similar to those found in the simulation of the homing maneuver for the guided missile, of a space vehicle rendezvous and docking maneuver, and many other situations in which two bodies are required to meet in space with an appropriate attitude but no specification of time. From the point of view of the simulation program what do we find in the three parts?

The physical system probably contains relative position and velocity sensing devices (a radar possibly), an autopilot for attitude control, a jet engine thrust controller, and the various aerodynamic control surfaces with their local position controllers. The environment, in this example, contains the aerodynamic forces and their consequent integration through the equations of motion to establish aircraft position and attitude. The stimuli are both predictable (changing initial situations as might result from consideration of the local air traffic) and random (wind gusts, radar noise, etc.). The measurements of performance are landing force and velocity, and the accelerations experienced at the pilot's seat during let-down.

This example suggests the kind of programming that is needed in aerospace simulation, and the qualities (amplitude and frequency) in a real-time representation.

For many years now analog computers have been used with great success to simulate physical systems, especially those which were the subject of engineering design. It is likely that they will continue to be used with equal success in the future. They are characterized by favorable qualities for this kind of work, the most important of which should be stated in a discussion of possible computers for simulation.

(a) Analog computers are readily programmed for simulation by the parallel interconnection of elementary computing devices to match the equations that describe the original system.

(b) Analog computers obtain a relatively high speed computing capability from their "one device per required operation" organization which permits a concurrent computation of many parts and thereby typically a "faster than real time" simulation of common aerospace systems and situations.

(c) Analog computers contain an extremely simple method for integration with respect to time, facilitating ready solution of nonlinear differential equations and thereby a simple means for simulating the time behavior of physical systems.

(d) Analog computers represent the variables describing a system to be simulated by continuously variable voltages which are then readily displayed in parallel on commonly-used strip chart recorders, oscilloscopes, and x-y plotters.

(e) Analog computers have a "get-at-ability" which is considered by most scientists and engineers to be desirable when employing an equipment for simulation purposes. One has a feeling of contact with the system being represented, through the "one-to-one" relationship between sub-systems and computer hardware and also through the ease with which program rearrangements and system parameter adjustments are obtained.

(f) Analog computers havea resolution equal to or better than the needs of most engineering systems to be simulated. However, care must be exercised in parts of problems which are of increasing importance in the aerospace industry.

Does this list of favorable qualities omit any that are desirable in computers for simulation? In other words in the years during which analog computers have been used in simulating systems like the automatic landing system would any additional ability have been useful, or can we foresee an increasing number of situations where other qualities would be desirable.

Consider the automatic landing problem and we can draw some conclusions. Firstly the generation of aerodynamic derivatives of one, two, or possibly three variables is inconveniently accommodated by the common analog function generators. A difficult problem to solve in any manner but we can seek alternative methods of information storage. Secondly, the possible need to incorporate aircraft structural behavior due to its interaction with the attitude control system suggests at least a more economical method of representing distributed systems. In the same vein, the completely parallel nature of the analog computer organization may not be solely a desirable feature for it implies an unreasonable large number of components being required for complicated simulations, a number which can rapidly get out-of-control. Thus we might seek to incorporate in the computer a method of using components sequentially at high speed, similarly to the use of the arithmetic unit of a digital computer.

Next we are almost certainly today in the simulation of complex aerospace systems to find need for representing sampled-data devices, and logical elements. More and more digital devices are being incorporated in such systems and thus a simple computer mechanization of their characteristics is necessary. Finally, the resolution with which variables of primary importance to an investigation can be represented is increasingly insufficiently handled on the analog computer. It is true that rescaling is possible, even automatic rescaling, but it would be much more acceptable if the computer could represent an approach and touch-down maneuver from 10 miles range and 2,000 feet altitude without any changes.

We can therefore suggest adding the following abilities to the computer for simulation.

(a) An ability to store data in a flexibly changeable manner so that system parameters and environmental functions can be readily loaded into the simulator and adjusted at will, even automatically under the control of the computer.

(b) An ability to perform computations at high speed and sequentially in order to economize on the use of analog components that are assigned to exactly similar operations (e.g. in the representation of a distributed system having important structural, thermal, acoustic, or electromagnetic behavior outwardly the simulator would appear to represent the physical system faithfully in space and time, even though only sample points of the structure are displayed.)

(c) An ability to simulate logic systems described by discrete-valued binary variables rather than continuously-valued variables.

(d) An ability to provide when necessary an increased resolution in the simulation of a situation. (Homing studies, Space-vehicle trajectories, etc.).

To these abilities in which the conventional analog computer is deficient can be added two very desirable computer qualities that are not well-represented in the analog computer.

(e) A convenience in applying the program to the computer, checking it out, and storing it for future use, (in place of the present mechanical patch panel, etc.). This would reduce the turn-around time for the general purpose computer.

(f) A self-programming quality, or at least in place of it the possibility of "software" rather than hardware which reduces the tasks of programming.

Not unexpectedly, these desired additional abilities and qualities are well represented in the conventional general purpose digital computer. Its three prime features are (1) flexible program and data storage in random access, high speed memory or slower speed bulk storage, (2) an economical sequential computing procedure which uses a common operating unit to perform many mathematical operations, and (3) a method of computation which provides increased resolution, theoretically without limit at the expense of time for computation. The general purpose digital computer has been constructed conventionally in a manner which does not provide the favorable qualities listed above for the analog computer. It is therefore reasonable to attempt to match the requirements of modern aerospace simulation by a derivative of the two computers -- a hybrid computer.

To be universally valuable in the role of a simulator a hybrid computer <u>cannot</u> be a simple combination of an analog computer and a digital computer, each allowed to communicate with the other. It must be a derivative of the two equipments (use both continuous voltage and binary word representations of variables which describe the physical system), of the two organizations (operate in parallel and serially), and of the two computing methods (be programmed so that units can be operated sequentially and/or concurrently as best suits the requirements).

To be practical, the hybrid computer of today will take advantage of existing hardware. It will be modular in construction, allowing the build-up of capacity with time from a basic unit. It can be viewed as having five parts:

1. An analog section with a parallel array of high speed integrators, summers, multipliers and function generators.

2. A digital section with logic and arithmetic ability, and volatile data storage.

3. An analog-digital data communication system to translate analog voltages to binary numbers and vice-versa.

4. A program control capability to maintain the timing of events in the computer.

5. Input-output display and equipment monitoring for program checkout.

A typical use of such a system would find the analog section performing high-speed integration of non-linear differential equations with the digital section providing precise low speed computation of slowly changing variables. These variables might require the numerical integration of differential equations or might be the result of an algebraic function generation. Any other function generation or algebraic transformation might fit on the analog equipment or be performed digitally. Samples of computed data would be stored digitally and logical data evaluation would also be performed digitally. The program timing and control would be a responsibility of the digital equipment. However consistent with their normal separate operation, an application would require all operations on the digital computer to be completed cyclically, whilst the analog computer might have two modes of operation, one a high speed iterative mode, the other a continuous computation mode.

The kind of operation envisaged for such a system, does not demand an expensive digital computer. As might be expected in the combined use of analog and digital computers, the 3-4 decimal place accuracy of the analog computer (~ 12 bits) is well matched by a 16-32 bit range in word length on the digital computer. This requirement appears to follow through no matter the type of digital section considered, be it gpds, DDA, special purpose or multi component digital computer. The digital section needs to be fast, for the parallel, high-speed nature of the general purpose analog section is otherwise frequently wasted. Other features of the digital section are the subject of considerable discusiion.

The choice of linkage equipment is rather easily made. The accuracy of the analog section determines the word length required in the converters. Typically it is 12-13 bits plus sign.

A number of A-D channels are required. Signals can be simultaneously scanned, stored in analog track-and-store units, and converted sequentially by one converter to binary number form for storage in the digital section. Otherwise they can be sequentially scanned, without the need for analog storage, and converted to binary number form. The D-A channels are either separate, each channel accepting binary numbers representing consecutive values of a variable and converting them to analog voltages which are updated at the operating rate of the converter, or they also can use a multiplexer. If a multiplexer is used then again the analog signals are stored between the updating times on track-and-store devices. The conversion rates depend on the settling time of the converter. The D-A converter operates in a parallel manner and can be designed to reach a steady value in no more than 10 microseconds. The A-D converter operates serially, the diode gates settling one after another, and can be designed to reach a steady state in approximately 60 microseconds. Thus analog signals can be translated into binary numbers at a rate of 15,000 samples per second.

The control operations desirable in a hybrid computer must cover the needs of typical applications. Certainly there is need for being able to control the mode of operation of the analog section rapidly and consistently. The operation of individual integrators or their operation in groups must be controlled from computed signals. The intention to exploit the high-speed capability of the analog section implies that all switching operations desirably be electronic. Relay switching but presumably will slow the simulation of some systems. The sampling of computed values in the analog section must be controllable. Read-out and read-in of data should be automatically controllable. This suggests automatic coefficient setting equipment under the control of paper-tape, digital memory, or computer values, and a complete monitoring system similar to these already available on modern analog computers (ADIOS on the EAI 231-R system).

V. Mathematical Characteristics of Design Problems Requiring Hybrid Simulation

From experience with problems to be solved in the design of aerospace systems one can suggest that problems having one or more of the following mathematical features should lead to a consideration of hybrid computation for system simulation.

1. Simultaneous differential equations with widely different parameters which produce both low and high frequencies in the solution.

2. Differential equations to be solved at high speed, their solutions for different initial conditions or parameters being used in a prediction, iteration or optimization process.

3. Combinations of continuous and discrete variables as in the description of a sample data system or a computer control system.

4. Perturbation analysis about slowly changing, precisely established solutions.

5. Statistical analysis requiring repeated solution of differential equations, including Monte Carlo methods for deterministic problems. This is essentially a data storage and simple evaluation task around the solution of differential equations.

6. Filtering and processing continuous and sampled data for evaluation purposes.

7. Partial differential equations to be solved by serial integration procedures.

8. Ordinary differential equations accompanied by transport delays.

A few examples of practical applications of hybrid computation in aerospace system design help to clarify the situations in which these mathematical features occur. Consider the problem of designing the guidance and attitude control system for a space vehicle attempting to dock with a target vehicle, possibly the manned orbiting space station. This docking maneuver might be manually controlled by the pilot in which case the design of the control mechanisms must take due consideration of the problems of a man-machine system. A close-range maneuver control system design suggests immediately the use of an analog computer to simulate the intended system. The choice of the analog computer is particularly emphasized by the desirability of checking the design with a man "in the loop" in real time. Also the selection of control system parameter values, and the possible re-arrangement of any use of sensors for feedback purposes is particularly easy when they are mechanized on such a computer. The later inclusion of real hardware is more readily accomplished using the analog computer. However, there are features of such a control system which are discrete or sampled and, therefore, suggest the use of digital circuitry. The reaction jets which adjust the attitude of the vehicle are "on-off" rather than continuously adjustable. The manual 3-axis control stick used by the pilot energises in a number of pre-selectable ways some, but rarely all of, say, 16 jets. Between the control stick and the jets there is a logic unit whose operation is modified by other manual inputs. These two features of the system could be mechanized with difficulty on a conventional analog computer. They are readily established with the logic capability of digital equipment.

The same maneuver control system could be completely automatic with a beacon-homing device replacing the pilot's decisions, but in either case, the system is most likely to many parts which are discrete in their operation and thus call for digital equipment in their representation.

A second example of the desirable use of a hybrid computer is in the analsys of rocket parameters and staging times for maximum efficiency in orbit injection. A multi-stage rocket's flight into orbit with the necessary considerations of aerodynamic forces, on-board control system, and staging with its consequent sudden changes in mass, inertia and thrust, is a complicated problem in simulation. To achieve any efficiency in considering many possible system parameters and flight path designs, a high speed simulation of a typical six minute flight to injection is essential. Thus, the speed of parallel computation of an analog computer is desirable. The automatic control and evaluation of the many different sets of parameters with their effect on the selected criterian of efficiency is most appropriately achieved using a digital computer. Such an arrangement would save considerably on the time presently required by an analog computer and even more so on that of an IBM 7090. This kind of problem, a search for an optimum set of parameters for a dynamic physical system, is one having many varied application. Its implications in computation - the solution of differential equations plus the need for up-dating stored values and making of logical decisions according to a criterian - suggest strongly the use of a hybrid computer.

There are many other examples of problem types which place a mixture of demands on computers. They should be most readily solved by the use of a hybrid computer. In choosing such a computer there is good reason to consider both the technical features essential in the computer, and the economic comparison between the different possible organizations.

VI. Hybrid Computer Applications

Although the different parts of a hybrid computer can be usefully applied to simulation tasks by themselves it is instructive to describe problems which can take advantage of dual capacity of the computer, i.e. its analog/digital capacity in equipment type, organization, and concept.

A problem in simulation can be well-suited to the hybrid computer in any of three ways:

1. Because the physical system is best described in terms of both continuous and discrete-valued variables. It is a hybrid system containing sampled data devices, logical control, or digital computers alongside equipments best described by transfer functions or differential equations relating continuous variables.

2. Because the method of programming uses high speed continuous analog circuits sampled and controlled from digital logic and storage equipment. The system might be a distributed system in which spatial variations of temperature, stress, potential, etc. is important to the simulation and thus a partial differential equation must be solved by the iterative matching of boundary conditions. Within this class one must include cases where increased resolution requirements demand that an otherwise analog program must use digital equipment for part of the simulation.

3. Because the experimental procedure to be used with the simulation of a continuous analog-type physical system requires an information storage and evaluation procedure that is to be automatic. This represents a possibility for automatic design in which the design engineers evaluation of performance is replaced by a computer evaluation and parameter changes are effected automatically.

Hybrid physical systems are becoming more common in aerospace design as more sampled-data, digital devices are incorporated. Typical devices are pulse radar systems, communications systems, on-off control systems, digital guidance computers, digital sensor.

Hybrid programming, at present, relates to the use of analog circuitry (a few amplifiers,attenuators, multipliers, etc.) as a complex "arithmetic unit" with a digital program having iterative/sequential operation and information storage. Typically this technique is used to represent distributed flow and diffusion situations as might be found in the design of ablative surfaces, rocket and jet engine nozzles, ion engines, nuclear reactor controls, etc.

The second type of direct hybrid programming is well represented by the simulation of missile homing, rendezvous guidance, automatic landing systems, etc., where the translational motion and function generation is well simulated with digital equipment, and most other simulation requirements are efficiently computed with analog equipment, providing the speed of simulation and output display features desired.

Hybrid experimental procedures provide the analog simulator with the program and data adjustment capability of a stored program machine and promise interesting possibilities for the future.

THE EVOLUTION OF HYBRID COMPUTERS

Hybrid computation came into being, as early as 1958, as a means of simulating the complete mission of certain aerospace vehicles. The task required computational speed exceeding the capabilities of the largest digital computers and a precision beyond that of the best analog computers. Combined systems of large general purpose analog and digital computers were created to solve this type of problem. The experience of these efforts and other experiments with analog and digital techniques in recent years have led to a growing hybrid computer technology. It is considered that these are the formative years in the evolution of a type of computer simulation in which the differences between analog and digital computing technique will dissolve through the development of general purpose scientific simulator system.

I. Some Historical Notes

During the nineteen-fifties the capabilities of electronic computers expanded so fast as to stay well ahead of the needs of the average computer user. Such was the case in both the analog computer and scientific digital computer fields. One effect of this situation was the formation of two schools of experts with opposite views on the choice of the "best" general purpose scientific computer. Differences of training, experience, and semantics led to a serious barrier to communications between these two groups.

The earliest attempts to combine the computation of analog and digital computers took place in about 1958 at the Convair Astronautics plant in San Diego and at the Space Technology Laboratories in Los Angeles. In both cases the job at hand was the complete mission simulation of the trajectory of a long range missile. The speed of the analog computer was a necessary element in the study to permit a "real time" simulation of the rapid motion of the vehicle and of control surfaces. However, the dynamic range required of the simulation was in excess of that of the best analog computers. That is, the ratio of the total range of the trajectory to the required terminal phase resolution (a dynamic range of 10^5 to 10^7) was greater than 10^4, the upper limit of analog computer dynamic range for small programs. Hence the digital computer was used to calculate those variables for which such dynamic range was necessary. The most important of these were the navigational coordinates -- the digital computer performed the open integration of velocities to determine the vehicle's position plus the dynamic pressure, a function that is very sensitive to altitude and velocity.

It is fortunate that in such long range aerospace trajectory simulations the variables with wide dynamic range requiring precise calculation are not, at the same time, rapidly changing. Moreover the "high speed" variables do not require precise calculation. The early combined computer systems employed the largest and fastest digital computers available at the time -- Univac 1103A and IBM 704 -- together with 300 to 400 amplifiers of general purpose PACE analog equipment. In both cases even these fast digital computers were only just fast enough to perform the required repetitive calculations for the slowly changing variables of the simulation in real time.

Since the installations of the first combined computers at least a dozen computer laboratories have employed general purpose analog and digital computers together to solve simulation problems, and a number of attempts have been made to devise special purpose systems of analog and digital devices. Among the latter are the CADDA of the National Bureau of Standards and the "pulsed analog computer" of the MIT Electronic Systems Laboratory hybrid computers of a unique type are the combinations of a general purpose digital computer and a digital differential analyzer (dda), illustrated by the Bendix G15 with the DA-1 attachment and the Packard Bell PB250 with the Trice dda. The former system consists of a small, slow computer with an even smaller serial dda ($50,000 and $100,000 respectively). In contrast to this the Packard Bell system combines a small, medium speed computer $40,000) with a large serial-parallel dda ($500,000). Among the systems of general purpose computers, generally large analog computers have been combined with both large (IBM 7090) and small digital computers (PB 250, LGP 30).

A brief analysis of the applications to which existing installations of combined systems have been applied leads to these generalizations. For the most part the analog computers in these systems have been employed in a normal manner to simulate the dynamic behavior of physical systems by solving sets of non-linear, ordinary differential equations, while the digital computer has performed one or more of the following three functions: complex control logic, storage of arbitrary functions or sampled analog functions, and high precision arithmetic primarily for numerical integration. Examples of the applications are:

A. Analog Computer Plus Digital Control Logic

A system that in itself contains discrete control functions of continuous dynamic variables is appropriately simulated by a hybrid computer. The kinetics of a chemical process are simulated by continuous analog means while its digital control system is represented by a digital program. Similarly an aerospace vehicle with an on-board digital computer, control system, or autopilot is simulated by hybrid techniques.

B. Analog Computer Plus Digital Memory

A very common difficulty in the simulation of a chemical or nuclear reactor is providing an adequate representation of the transport of fluid in pipes from one point to another -- from reactor to heat exchanger. The simulation of this transport delay of a dynamic variable, such as the time variation of the fluid temperature, is very nicely accomplished by the use of a digital computer for storage of the temperature function for a fixed, or variable, length of time. Digital computer memory has also been used effectively to store multivariable arbitrary functions -- an operation which is seriously limited in the analog computer.

C. Analog Computer Plus Digital Arithmetic

This type of application is the "classic" one where the digital computer is used to perform precise, numerical integration of space vehicle velocities to keep track of the exact position of the vehicle over a very long range flight.

It should be noted that a significant difference is apparent in the applications of computer systems employing a very small digital computer and those with large, very fast, computers. In general the small machines are limited to execution of control logic programs, one or two channels of transport delay simulation, or limited function generation programs. Since numerical integration and complex function generation by digital programs require considerable time for each calculation, for each discrete step in time, only the fastest digital machines can be used effectively for these tasks.

II. The Elements of Hybrid Computers

A. Digital Computers

Many conflicting factors influenced the choice of digital computers used in hybrid systems. Computer speed and economics have probably been dominant. Since there are so many computers on the market today that have sufficient speed and that span the complete range of prices, it is more instructive to examine the features that are essential for hybrid computation.

1. Speed. The speed of execution of arithmetic operations is most important, and this is a function of memory access time and multiplier speed. The access times of 2 to 5 microseconds are currently popular. This means the time for addition of two numbers is 4 to 10 microseconds. Multiplication and division take longer -- times of 15 to 40 microseconds are generally available and quite satisfactory. Overall program speed can be increased by the use of index registers -- three registers is desirable; more are useful. Special instructions for subroutine entry, for executing commands out of sequence, and for testing and skipping can help increase computing speeds.

2. Word Structure. The basic requirement is for a fixed point, binary word of at least 24 bits. Since round-off errors affect the last several bits a smaller word size would result in a dynamic range limitation of less the 10^6. A longer word may be useful in a few applications where fixed point scaling may be

difficult. Floating point computations may make things easier for the programmer but should not be depended upon at the expense of computational speed. It may be noted that the equivalent of fixed point scaling is a necessary part of the analog program, and hence floating point operations may not prove as advantageous as for some all-digital programs. Decimal format and character oriented machines do not offer any advantages for hybrid computation, and usually they are slower than equivalent sized binary computers.

3. Input/Output. High data rates in and out of core memory and any feature that minimizes loss of computing time for input/output operations are highly desirable. In addition a fast, flexible means for communicating control signals to and from the analog section of the hybrid system is necessary. Three kinds of control signals are usually provided: interrupt lines, sense lines, and output control signals. It is by means of these controls that the sequential operations of the digital machine are made compatible with the parallel simultaneous operations of the analog machine. Since communications must be made with many points in the analog computer a number of these control signals are needed. Interrupt signals from outside the computer stop the current sequence of calculations and force transfer to another sequence. Sense lines simply indicate to the digital program the current state of devices outside the computer which may be sensed by specific programmed instruction. Other programmed instructions will send control signals outside the computer on the output control lines.

4. Memory. As noted above, the digital computer main memory should be a high-speed, magnetic core. Since most hybrid applications do not require a large memory for either program instructions or data, four, eight or twelve thousand words of core memory should suffice. Larger memories may be desired for special digital programs and larger hybrid problems when more experience has been gained in this field, thus expandability of a memory to 16K words is a good feature. Newer computers are being introduced with small, very high speed, "scratch pad" memories. Such memories may have cycle times less than a microsecond and are used to store intermediate arithmetic results. This feature increases the overall computation speed of the computer.

The normal manner of operating an analog computer involves a fair amount of non-computing time when the computer remains in the Hold or Reset-mode. These intervals may range from seconds to minutes while adjustments are made, pots are set, or recorders change, or while the programmer analyzes results. It is not possible for the analog computer to operate on other programs at these times, however, with a hybrid system, where such waiting periods are likely to occur also, it is reasonable to consider having the digital computer work on a different program during the intervals, whatever their length. Appropriate "interrupt" and "memory lock-out" features are possible to permit time sharing of the digital machine without affecting the hybrid program and without the danger of one program interfering with the other. The secondary program (a strictly digital problem) is simply stored in a "protected" part of the core memory and utilizes all the bits of time not required by the hybrid program.

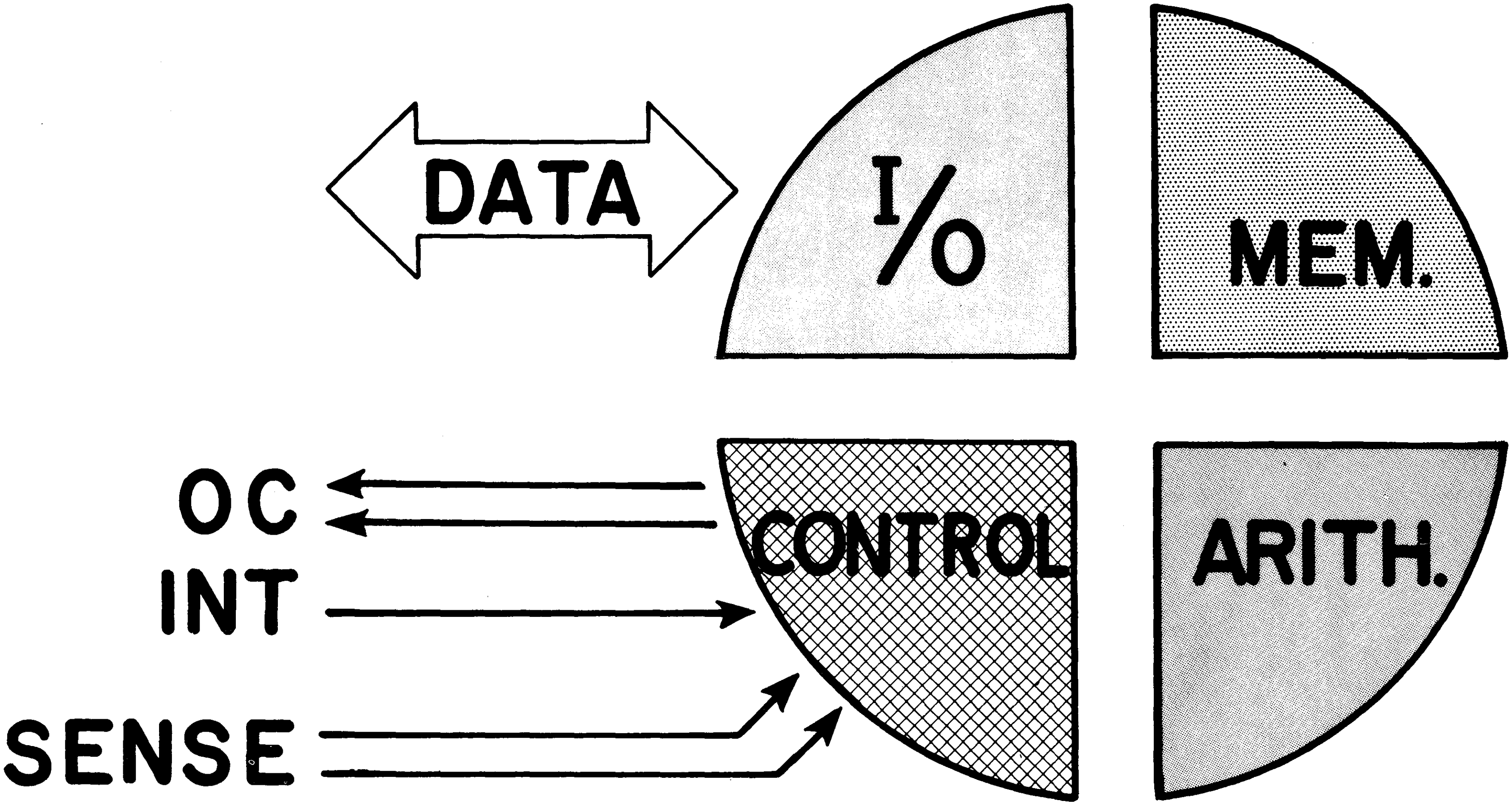
5. Peripheral Equipment. In many digital computer installations the investment in peripheral equipment rivals that in the central computer. Current hybrid computer applications require only a minimum of digital peripheral equipment. The graphic output equipment associated with the analog computer is sufficient for computational results. Punched paper tape reader and punch and typewriter may be all that is required for programming. Larger systems in the future will employ punched cards and magnetic tape, primarily for rapid change-over of problem and automatic check-out. Large off-line data storage does not appear necessary for most applications.

In summary the digital computer must be characterized as a sequential machine (Fig. 1). For effective use within a hybrid system the machine (a) must have sufficiently high internal speed for it to appear as though a number of calculations were taking place simultaneously; (b) must be organized for maximum speed in executing mathematical calculations and (c) must have efficient means for input and output of data during calculation.

B. Analog Computers

In contrast to the above the analog computer is a parallel machine with many computing components and I/0 devices operating in concert. There are few, if any, features of the modern analog computer that are not appropriate to a hybrid system. However, only the largest analog machines have been used for general purpose hybrid simulation. The common measure of a large computer is that it has 100 to 200 operational amplifiers. Since two or more computers may be "slaved" together, larger systems are possible when required.

Analog computer features that are important for hybrid systems can be simply listed as:



SEQUENTIAL COMPUTER

FIG. 1 ORGANIZATION OF DIGITAL COMPUTER

- Integrators with multiple time scales
- Amplifiers for tracking and storing voltages
- Very fast control of the modes of individual amplifiers
- Automatic, remote control of the setting of potentiometers
- Fast, accurate multipliers and trigonometric resolvers
- High speed comparators with logic signal outputs
- Electronic switches (logic signal gating of analog signals)

In the early days logic equations or switching functions were programmed with relays and stepping switches, which were connected to the patch board by various means. Present day technology employs electronic switching of integrator modes and voltage signals at high speeds, and the delays inherent in relay devices can no longer be tolerated for logic operations. The logic building blocks common to the digital computer designer (flip-flops, gates, inverters, monostable multivibrators, shift registers, and counters) are ideally suited to these operations. Thus with electronic switches replacing relay contacts, logic modules have become an integral part of all new, large, analog systems. These modules are programmed like the other analog components by interconnections at a patch panel. Many signals occur simultaneously but they are logic signals -- two values, Zero and One, that change as functions of time. Input signals to logic programs come from comparators, manual switches, and external control signals. Logic program outputs go to integrator mode controls, storage amplifier controls, electronic switches (DA switches) to gate analog signals. As will be shown later, it is essential for a hybrid system to have a very significant complement of digital logic components. A few gates and flip-flops are not sufficient. The requirements for use of logic components in an analog computer for hybrid operation are so great that the EAI HYDAC Digital Operations System is an entire computer console with its own patching system used entirely for the programming of digital components for parallel computation. This console is really a complete logic computer. It is used together with a conventional analog computer to form what is truly an all parallel hybrid computer.

In summary, the modern analog computer must be characterized as a parallel machine (Fig. 2). It is not solely for continuous variables. It is a parallel assemblage of building blocks: integrators, multipliers, etc., for continuous variables. It is organized for convenient representation of an "analogous" physical system by means of a computer model constructed of these building blocks.

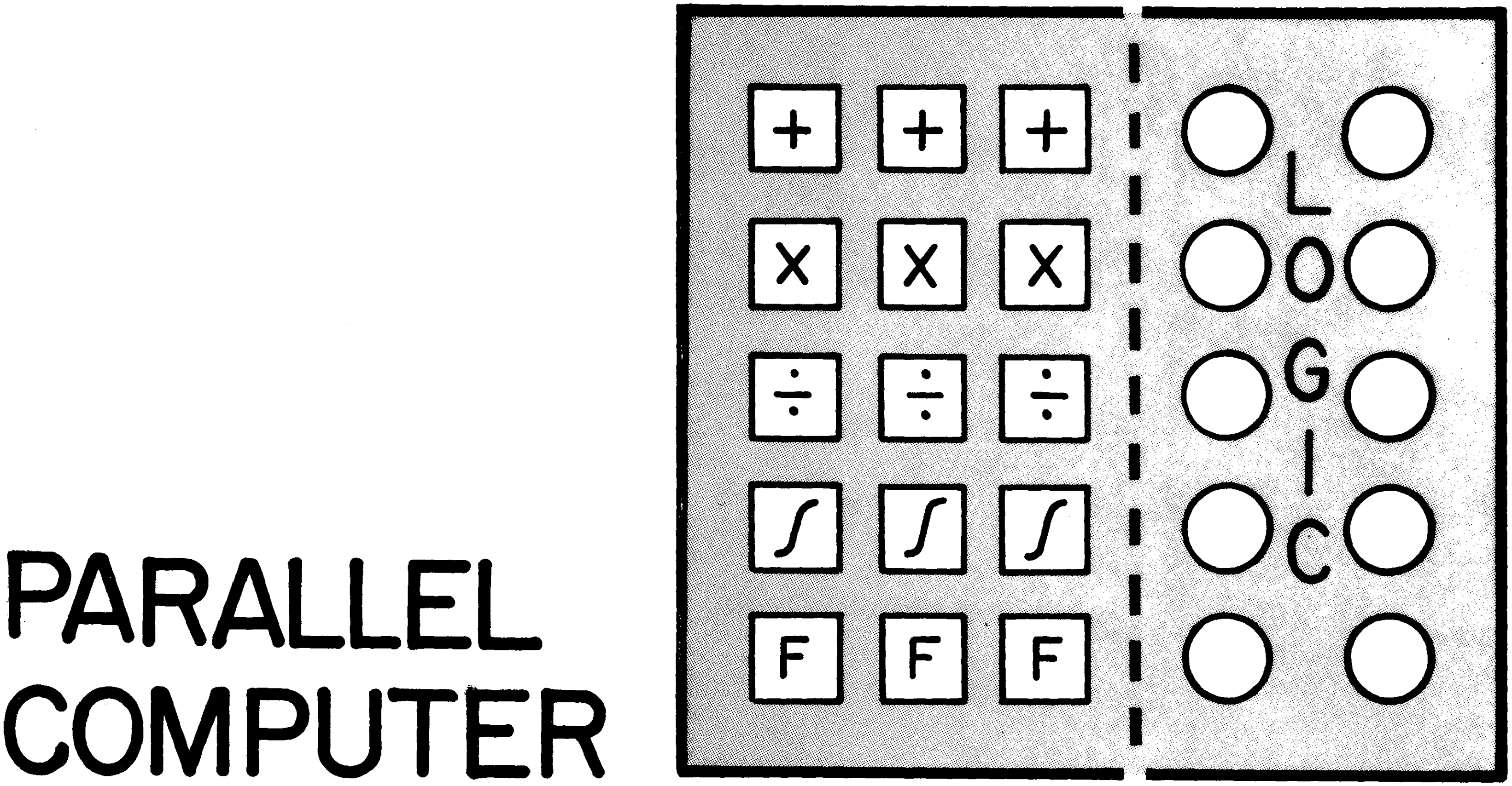


FIG. 2 ORGANIZATION OF ANALOG COMPUTER

C. Conversion Devices

In providing data communications between a sequential computer and a parallel computer, three kinds of devices are commonly used: the multiplexer, the analog-to-digital converter (ADC), and the digital-to-analog converter (DAC). In addition, all early systems have employed a timing and control unit which performs a relatively fixed set of operations, with manual switches to select options such as sampling frequency, and number of channels. Such "linkage systems" thus consisted of a timer unit plus a group of linkage building blocks prewired to perform a specific task. With the integration of digital logic components into the parallel computer, however, greater programming flexibility is possible by use of these logic units for timing control of the data conversions. Furthermore the converters and multiplexer can act very naturally as additional building blocks in the parallel computer. Thus it is likely that future hybrid systems will simply incorporate the "linkage system" within the parallel computer.

Usually several or many analog signals in a hybrid program will be sampled, converted and transmitted periodically to the sequential digital program. The numbers, of the several sequences of numbers to be entered into the core memory, can be accepted only one at a time. Since this is so, the conversions from voltage to number form can be performed one at a time -- first from one analog variable and then from another. The multiplexer is used to select one from many analog signals, to step through a sequence of signals, and thus to furnish voltage input signals to the ADC.

The output of commonly used ADC's is a binary number of 10 to 14 bits. A 13 bit binary output probably is the best compromise; for it represents a resolution of one part in eight thousand, and resolution of analog voltage signals is at best one part in ten thousand. Conversion times range from 50 to 300 microseconds. A typical time of 100 microseconds would allow the converter to be shared by 16 analog signals each with a frequency spectrum extending to 20 or 30 cycles per second.

DAC units should have the same binary word size as the ADC, except for special low accuracy uses. Conversion times are not determined by the converter but rather by the bandwidth of the analog amplifier following the converter. Data from the sequential computer appear only one word at a time, and some means for retaining the latest value, of each sequence of numbers, for each

LINKAGE PATHS

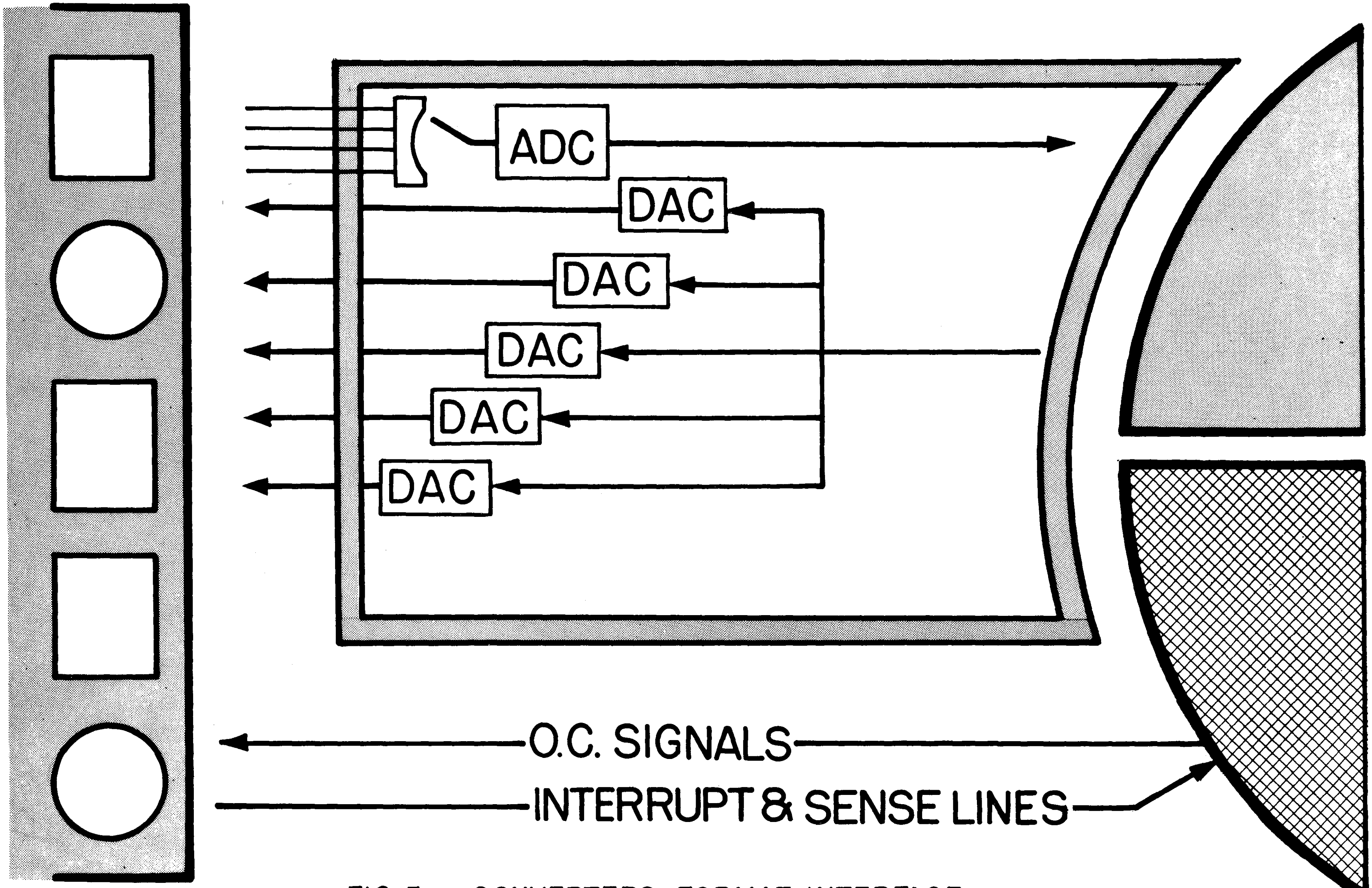


FIG. 3 CONVERTERS FORMAT INTERFACE

output function is needed. The sequence of numbers coming from the computer may first be converted to voltage values by a single DAC, and then distributed to storage amplifiers for each channel. It is more customary, however, to hold each of the latest words for each channel in a digital register which is an integral part of the DAC assigned to each channel. (Fig. 3)

D. Special Forms

As a passing thought it may be noted that while the primary emphasis here is being placed on the distinction between parallel and sequential operation, the term "hybrid", historically, has been used to imply the combination of continuous and discrete calculations, and that therefore consideration might be given to two special kinds of hybrid computers:

1. The Parallel Hybrid Computer, which is a proper term for the EAI HYDAC 2000 machine. This system combines an analog computer with a general purpose system of programmable logic building blocks, multiplexer, ADC, DAC's digital memory units for storage of sampled analog functions, and several digital numerical adders and subtractors. The application of this system encompasses an intermediate range of hybrid problems, such as:

   a. Transport delay simulation
   b. Single and multi-variable function generation
   c. Logic control systems
   d. Automation of the analog computer for parameter searches and optimization studies
   e. Simulation of numerical and sampled data control systems

2. The Sequential Hybrid Computer, which is exemplified by the experimental "pulsed analog computer" techniques developed at M. I. T. for use in an aircraft flight trainer. This system employs a sequential digital computer which controls a small number of analog functional components -- one multiplier, one reciprocal generator, one integrator, and several adding units. These units are interconnected and receive inputs by digital program control. They form, in effect, "analog subroutines" for the sequential computer.

III. The Sequential Parallel Hybrid Computer

A. Description of System

The term "hybrid" is most appropriately used to indicate the combined use of sequential and parallel computing techniques, first because the future growth in hybrid simulation will be predominantly in this direction. Second, and more important, is that from the standpoint of the programmer who must bring the two types of computers together to find a useful solution to a problem, the only really significant disparity lies between sequential and parallel operations. The difference between continuous voltage and discrete number is simply one of format. It would make little difference if the analog signals were frequency modulated, pulse code modulated, hydraulic, or pneumatic -- appropriate format converters could be found. The feature of hybrid computation that is of importance is: in part of the machine many operations are taking place simultaneously, and many, time-varying problem variables exist in parallel; while elsewhere a number of operations take place, one at a time in a repetitive manner, so as to effect the generation of several problem variables, as if they occurred simultaneously. Furthermore, the parallel computation is tied to a real time base: the very passage of time itself accounts for the changing of the basic independent computer variable. The sequential program is asynchronous -- not controlled by a clock. Operations are executed in sequence at whatever rate is possible, and for any reference to be made to the actual elapsed time external communication is necessary. This basic incompatibility requires that the interface between the two types of operation embody more than the simple format conversions performed by the ADC and DAC's. It is necessary for data and control information in the parallel machine to be available to the sequential machine and conversely that the latter be able to send data and control signals to many points in the former. Coincidence or simultaneity of events communicated to and controlled from the sequential program are particularly difficult to handle. The logic and data control of the interface equipment must resolve these differences in timing and operation. What might be termed an "impedance matching" device is needed between the parallel and sequential program in order to make most efficient use of both machines. The exact manner in which this is done will vary from problem to problem. (Fig. 4)

B. A Simple Example

An example will illustrate some basic considerations in defining a general purpose hybrid system. First the operation of the simplest of linkage systems: an ADC and multiplexer, a number of DAC, and an "interrupt clock". The flexibility of the stored program digital computer is relied on for control of these units. Assume 10 analog signals are to be converted one at a time. These words are placed in memory (average program time: 40μsec. per word) and then

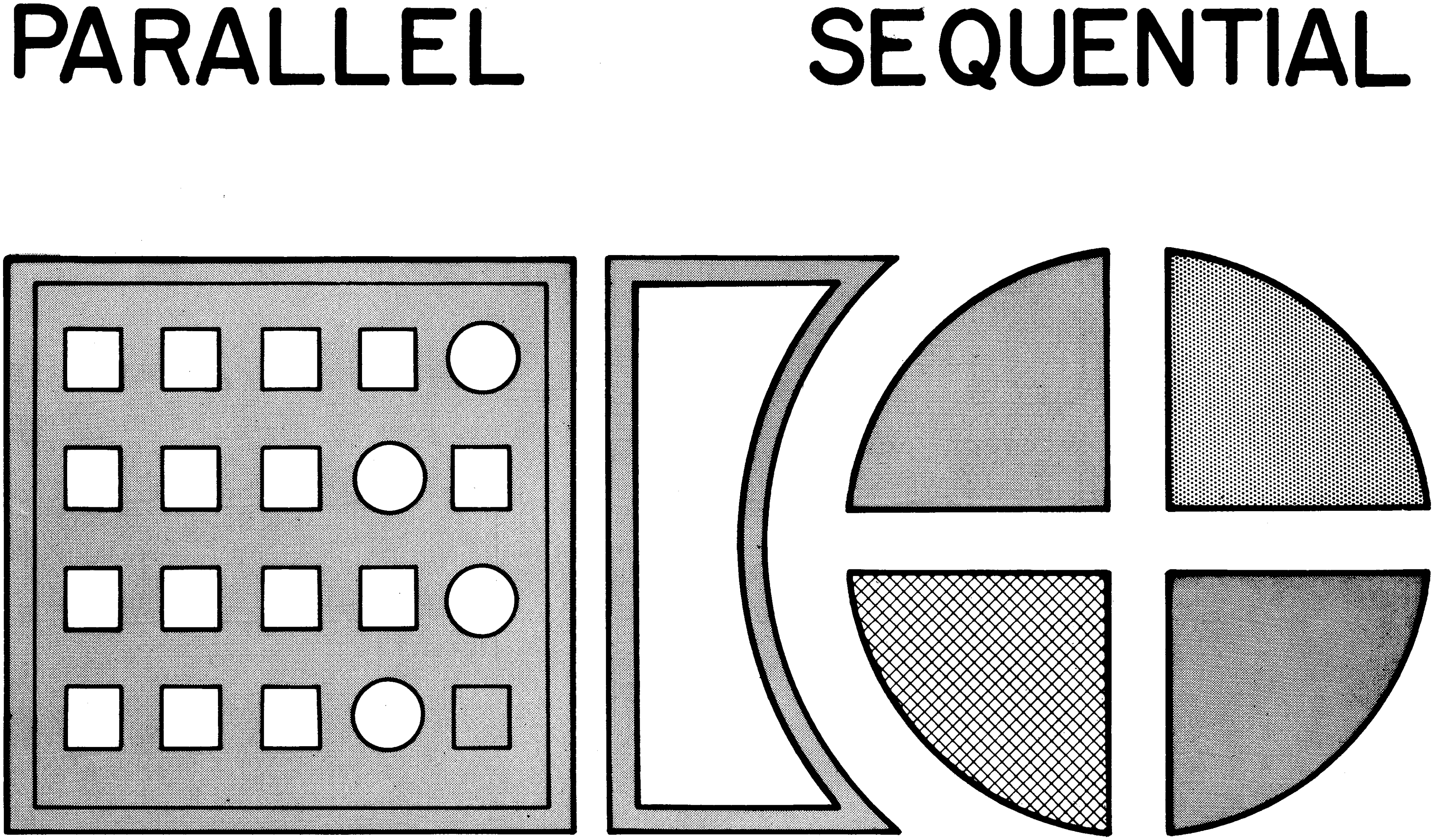


FIG. 4 THE SEQUENTIAL PARALLEL HYBRID COMPUTER

about 7.5 or 8.0 milliseconds of sequential, digital calculations takes place, followed by output from memory of ten words (20 sec. per word) to ten DAC's. The entire cycle requires 7500 +10 x 40 (input) + 10 x 20 (output) = 8100 microseconds of digital program time. If it is assumed the conversion of the data (A to D) requires 150 microseconds per word then 1.5 additional milliseconds, or 9 ms are needed if everything proceeds sequentially. Assume further that because of the frequencies of the analog signals it is necessary to sample at least some, and therefore all, of the channels at 100 samples per second. A "real time interrupt clock" is set at 100 cycles per second. This timer unit is an adjustable oscillator that sends an interrupt pulse to the digital computer. The latter then activates the ADC, waits 150 m seconds for a completion, or "Ready" signal, steps the multiplexer to the next channel, stores the converted word in memory, and then repeats this cycle ten times. With the tenth step the multiplexer resets to the first channel. The program then proceeds with the 7.5 milliseconds of calculation, outputs ten words one at a time to ten DAC's and then waits for the next interrupt pulse. (See Fig. 5, 6, and 7.) Manual controls are provided for selecting the interrupt clock frequency and the number of channels in the multiplexer stepping cycle.

C. Problem Areas

This is certainly a simple system and it appears to satisfy the basic requirements for communications. Some of the shortcomings of the system are apparent: Sampling and outputting of each channel do not take place simultaneously; 15% of the sequential program time is "waiting" time, and 3 to 5% is used to select and control devices external to the sequential computer. In other problems these percentages may be higher. The other weaknesses in this system lie in the fact that it was not designed to be a general purpose system. For example, the placing of logic, timing, control and storage functions in the digital computer severely limits its arithmetic processing capability. This approach is restricted in application to a class of problems for which the periodic "input/calculate/output" cycle is useful.

D. System Improvements or Variations

By programming the parallel digital components of the parallel computer to perform timing and control functions for the system the following changes to the above system are suggested:

1. Simultaneous sampling. If the sequential program operates on two or more of the input numbers together to calculate an output, then errors may occur since the input numbers were sampled at different times and correspond to different values of the independent variable. A similar effect may occur at the

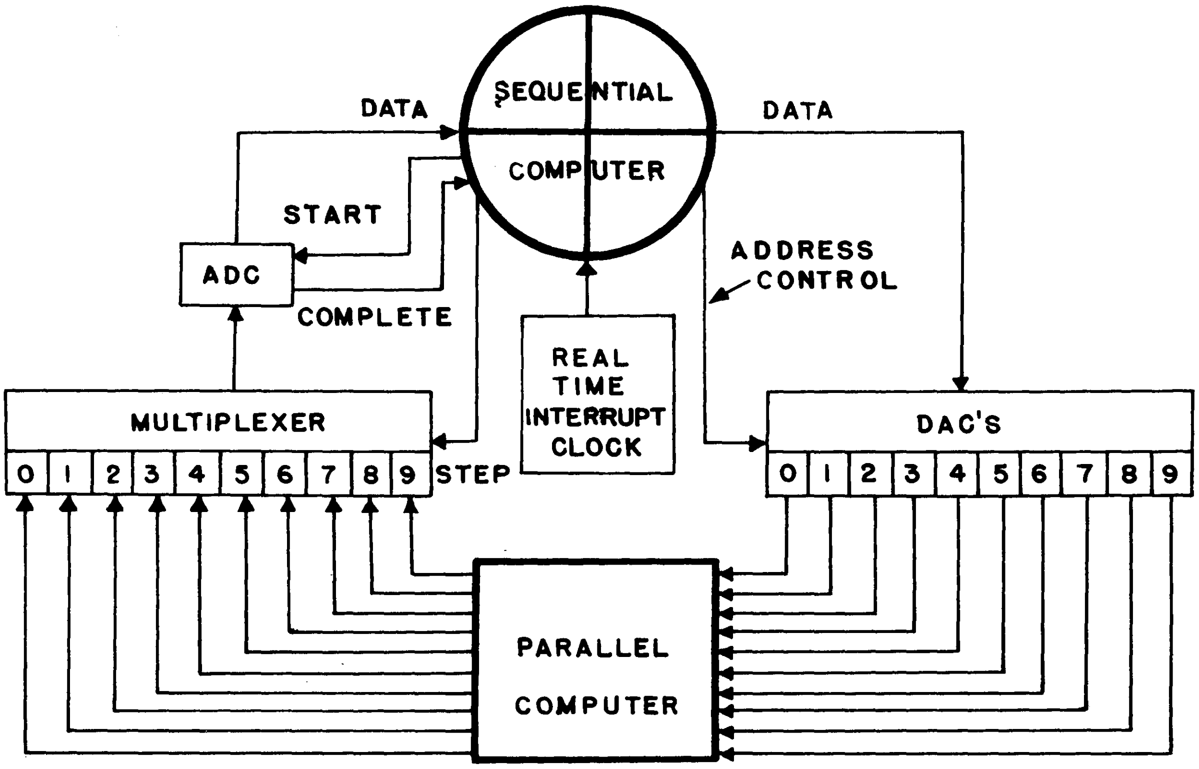


FIG. 5 AN EARLY HYBRID SYSTEM

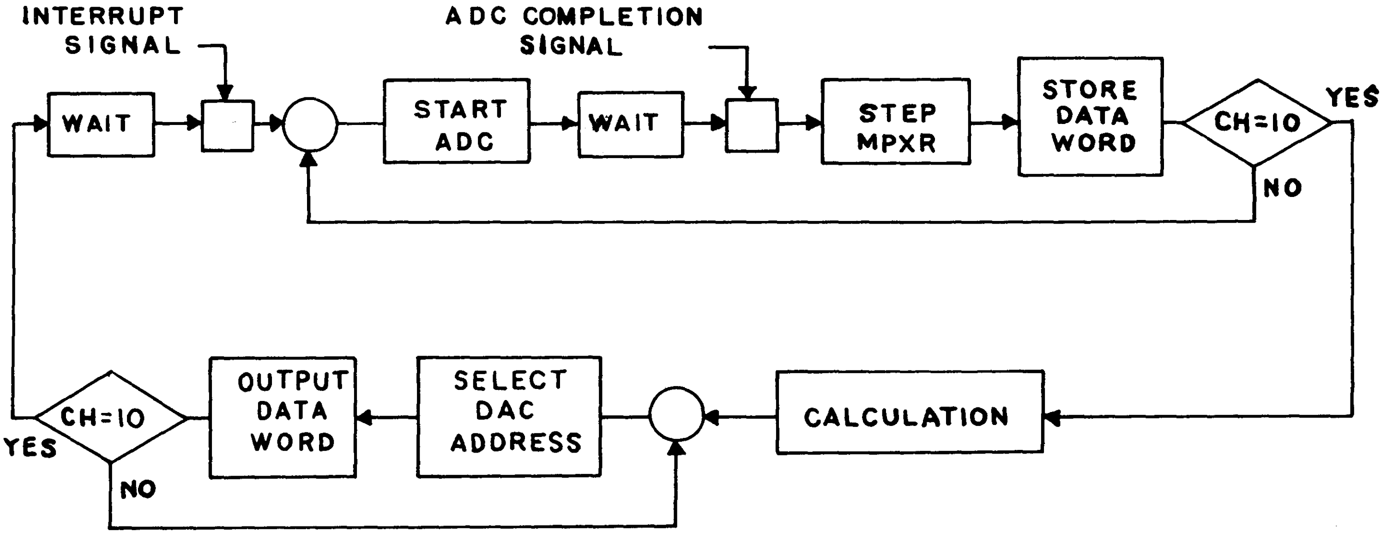


FIG. 6 SEQUENTIAL PROGRAM FLOW DIAGRAM

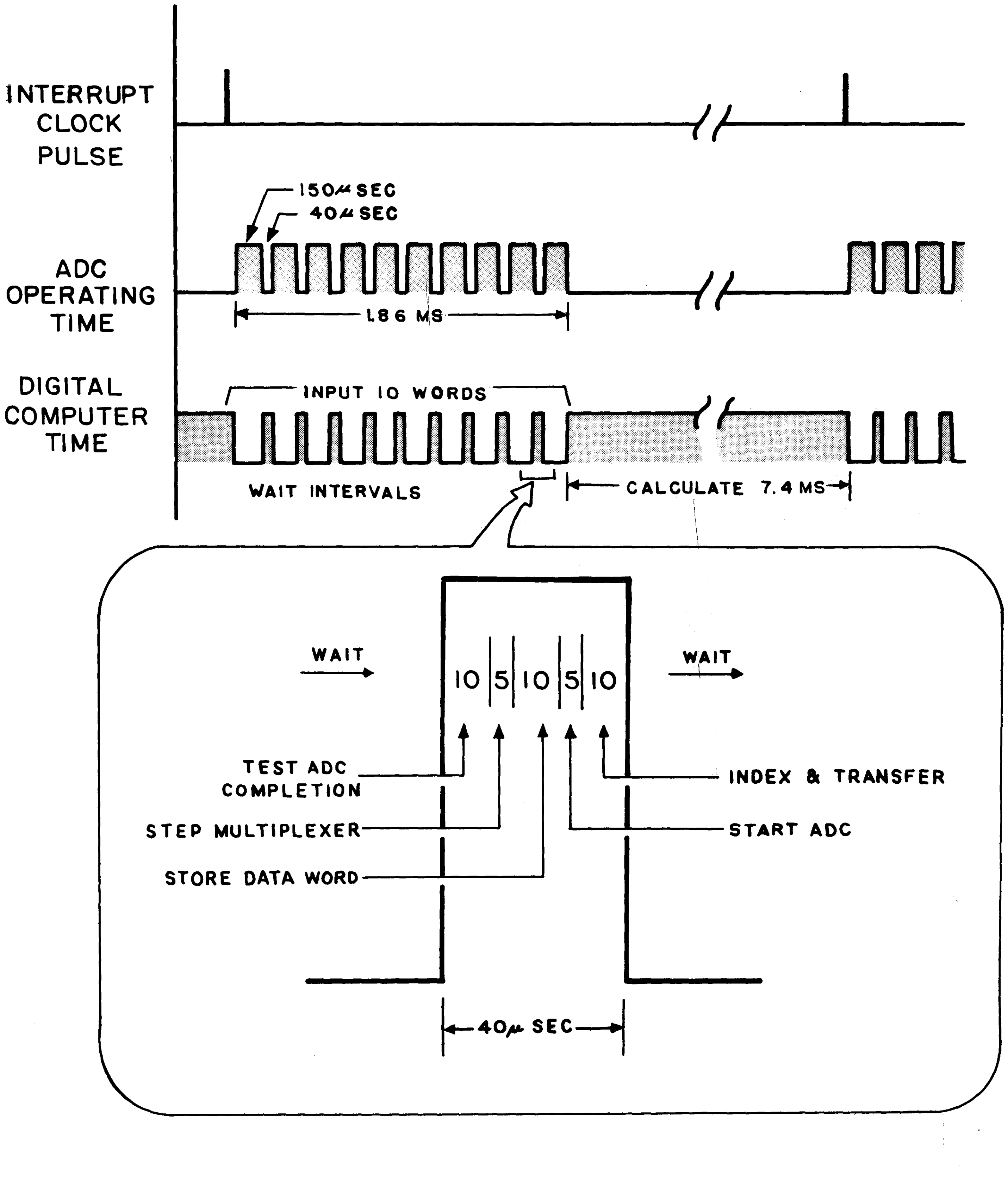


FIG.7 PROGRAM TIMING DIAGRAM

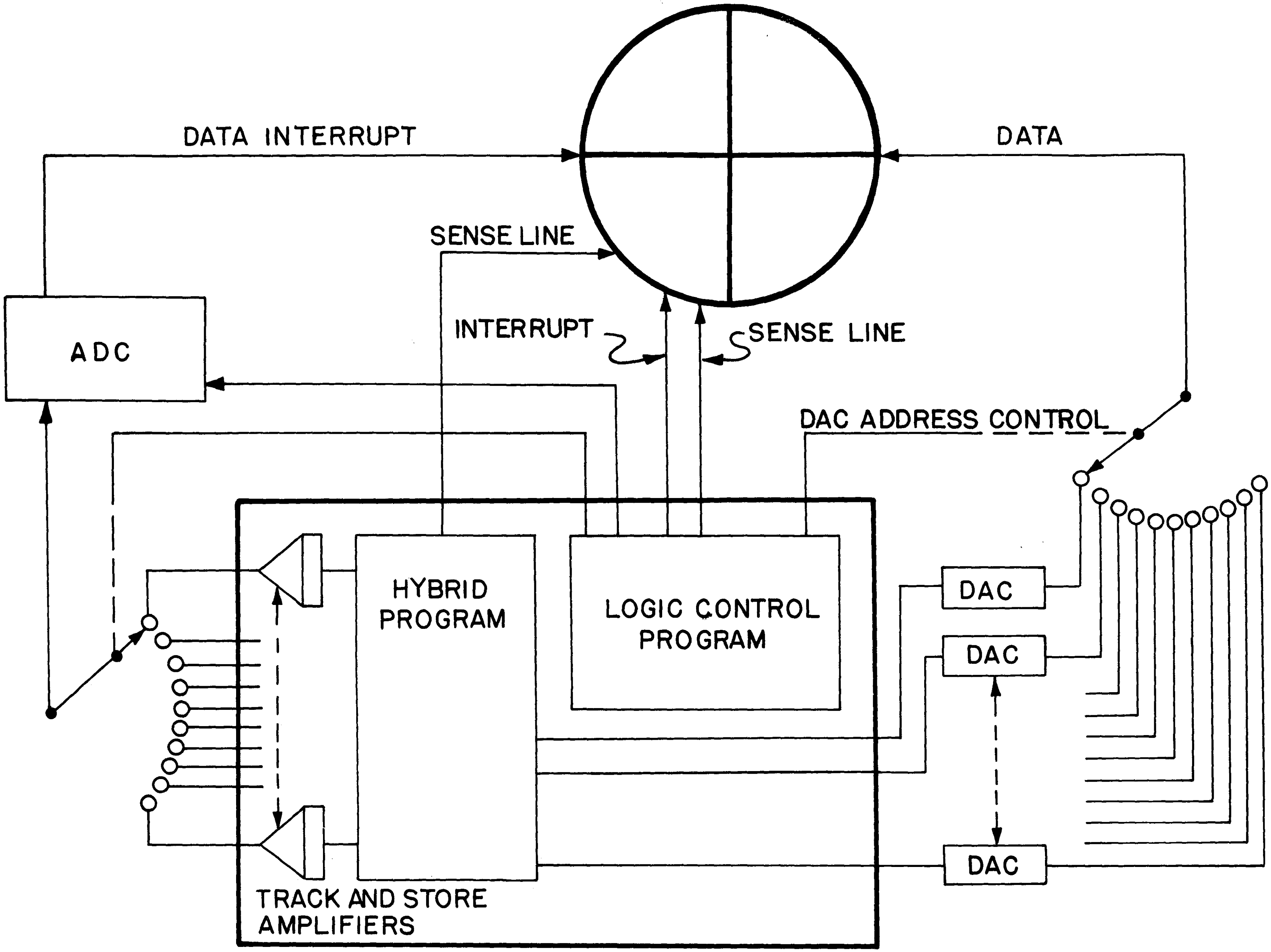


FIG.8 A MORE SOPHISTICATED INTERFACE

output since the numbers in a group of ten appear at the ten DAC's at different times. It is certainly possible by numerical means to compensate for the errors, at the penalty of additional program time. The common solution is to add memory to each of the ten input and ten output channels. Ten Track/Store amplifiers are added before the multiplexer and a control signal causes them all to sample, by storing the voltages, simultaneously. At the output, 13 bit registers are added in front of the DAC's. When all ten registers have been loaded, a transfer pulse causes all DAC values to be updated at the same time.

2. External timing of ADC and multiplexer. Sequential program time can be saved by permitting the control of the ADC, Multiplexer, TrackStore cycle to be controlled externally. (Fig. 8). A simple clock, counter, flip-flop, and group of logic gates will permit the input conversion cycle to run at its own rate -- interrupting the sequential program only at the completion of a conversion. Thus the conversion time can overlap the calculation time, eliminating the waiting time. Upon interrupt only 10 to 20 seconds may be required per sample; many control steps are eliminated. Similarly on output, the addressing and selection of output channels can be done by simple circuits rather than using sequential program time.

3. Real time clock to establish sampling frequency. If the sequential calculation involves numerical integration over a long term, the accuracy of the sampling interval is just as important as the round-off and truncation error in the numerical calculation. Although numerical means may be resorted to for very accurate integration, in a hybrid program the calculations still need to be referred to a real time base. This is done by using an accurately calibrated source for setting the sampling interval, or frequency. A good, high-frequency, crystal stabilized oscillator is an important part of the parallel digital subsystem. Sampling frequencies lower than the oscillator frequency are selected by use of preset counters.

4. Use of sense lines to reduce number of conversions. In the simple example problem only whole number data are transmitted to the sequential program . Thus, if the relative magnitude (greater or less than) of two analog signals is needed in the digital calculation, the two numbers must be converted, stored and then compared. This can be accomplished more simply by use of an analog comparator the output of which is sent to the digital computer by a sense line -- saving time and equipment. The state of any parallel logic component may be monitored conveniently by sense lines. These are tested in one memory cycle (2 - 5μsec.). If many such communications are needed the savings will be significant. Sense lines should also be added to allow the sequential program to determine the modes of the analog computer, the relative sizes and signs of error quantities, and the states of recording devices.

5. Detection of random events. With fixed, periodic sampling the sequential program cannot tell exactly when events take place in the parallel machine. With comparators and parallel logic, complex functions of analog variables can be monitored. For example, it might be required to determine when the overshoot in x_1 exceeds x_2 after the third cycle; but only when x_3 is negative and x_4 is less than x_5. After determination the sequential program can be interrupted to perform specific conversions and calculations -- asynchronously with respect to the primary conversion cycle (Fig. 9). In this manner the parallel logic avoids the delays in the sequential program and uses the latter only when required. The parallel logic program analyzes the data, interrupts the sequential program, sets up the proper channels for conversion in and out of the digital computer.

6. Multiple sampling frequencies. In the example problem all channels are sampled at a frequency determined by the highest frequency present in any one channel. It may often be the case that there are two or more groups of variables with different ranges of variable frequencies. It may then be appropriate to sample each group at different frequencies. Another approach using different sampling rates is to use several eight channel multiplexers in cascade so that the output of two of them feed two channels of a third which feeds the ADC. On each cycle of the third unit the first two are stepped, yielding different variables for those two channels on each cycle. Alternatively, each time the third steps to the two special inputs the corresponding multiplexer makes a full cycle. Timing control of these operations is performed by parallel logic components. (Fig. 10).

7. Asynchronous sampling. A completely asynchronous conversion system has been designed by one computer laboratory, in which the sequential program is interrupted only by comparators. Twenty analog problem variables are compared to reference values that are adjusted by the digital computer when necessary. Each comparator calls for conversion of some group of variables (the same variables may be called for by different comparators). When two or more comparator signals occur simultaneously or during a conversion operation, two levels of priorities are set up by logic elements to determine what interrupts are to be made. While the system appears complex, it is accomplished in a simple fashion in the parallel computer and makes good use of the sequential computer time.

A longer list of useful variations in the control and timing of sequential/parallel communications can be compiled. It can be said, however, that the omission of the above capabilities in the hybrid computer severely limits the flexibility and high speed capability of the computing system. For the most part, these features should be explained in terms of the particular problem applications.

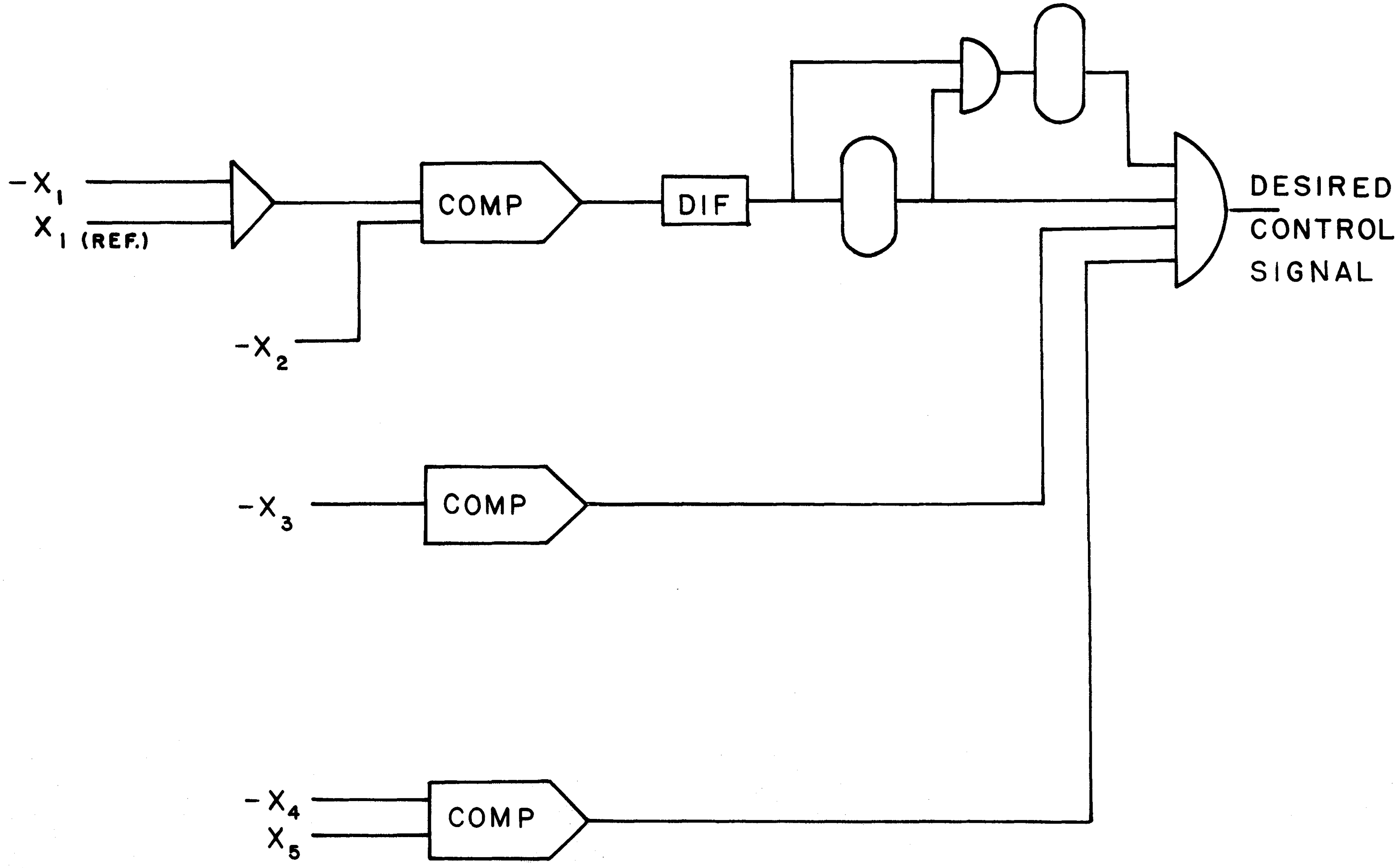


FIG. 9 DETECTION OF RANDOM EVENTS

E. Operating Times for Typical Mathematical Functions

The repeated emphasis upon the efficient utilization of the sequential program time, high arithmetic speeds, and programming tricks to gain speed can be seen to be warranted when one examines the sequential operating times for several typical mathematical functions, which, on the analog computer, would be executed continuously and in parallel.

a.	The sum:	$a + b + c + d$	40 microsec.
b.	The expression:	$ax + by + cz$	160 microsec.
c.	Sin ωt or cos ωt		215 microsec.
d.	Square root of $x^2 + y^2$		432 microsec.
e.	Generate $z = f(x, y)$, where two dimensional interpolation is required between functional values evenly placed in x and y:		0.5 to 1.5 millisec.
f.	Rotate a vector through three coordinate angles:		2 to 6 millisec.
g.	Perform one integration of a single derivative for a single time step:		0.1 to 1.3 millisec.

Allowing another millisecond or two for control and input/output instructions one can estimate the real time speed performance of this program. The speed is best expressed in terms of the useful upper frequency (at full scale) in a problem variable that can be represented by the computer . Although the exampl equations have no real meaning the frequency limit for such a program is about 1 1/2 cps. This does not seem like very fast performance for such a few simple equations. On the other hand it is fast compared to frequencies of some of the variables in an aerospace simulation program for which digital precision is required. In any event it should be clear for best utilization of the sequential computer care should be taken to reduce operating time whenever possible.

IV. HYDAC 2400 System Description

Experience derived from the successes and failures of the early

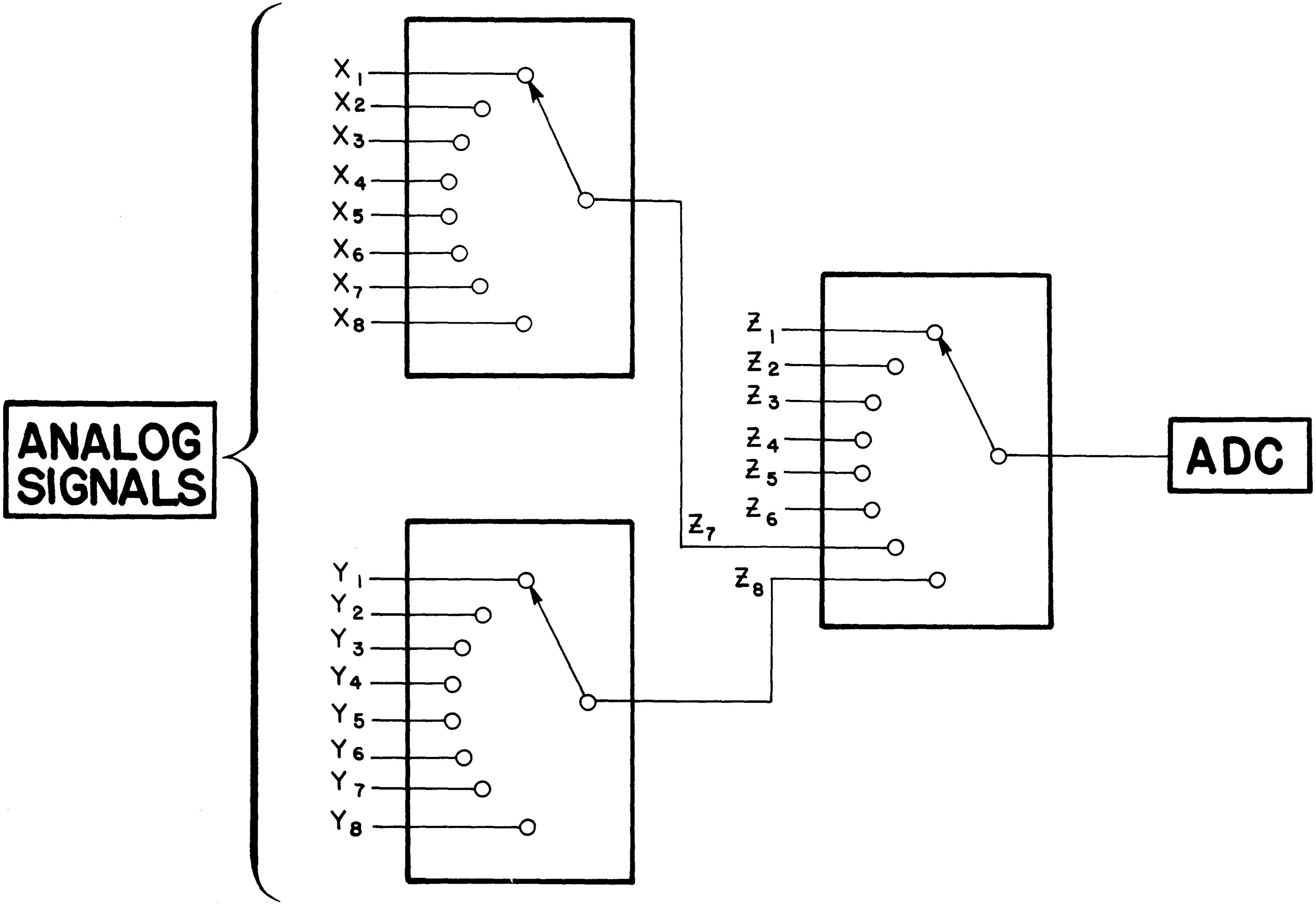


FIG.10 CASCADE MULTIPLEXERS

hybrid systems as well as considerations such as those above led to the first general-purpose hybrid computer, the HYDAC 2400. This computer satisfies all prerequisites of a compatible system of parallel and sequential computing components as discussed earlier.

In order to facilitate the understanding of the programming aspects of an aerospace problem on the HYDAC 2400 discussed in the next chapter, a brief description of the system is given here.

The HYDAC 2400 computer system is a successful integration of general-purpose analog and digital computers. It consists of three computers: the 231R-V analog computer, the DOS 350, and the 375 (3C DDP-24) digital computer. (For system block diagram, see Fig. 11) In the following, a short outline is given of each section of the HYDAC 2400, and the communication between sections.

A. 231R-V Analog Computer System

The 231R-V analog computer consists of standard computing components under control of a versatile, highly sophisticated control system. The heart of this control system is the analog memory and logic system (MLG). The MLG system adds a capability for programmed multiple time base and multi-speed repetitive and iterative computer operation. The system utilizes electronic mode control and high-speed analog memory in order to maintain the accuracy necessary for "real-time" operation. Various forms of analog memory (micro-store, track and store amplifiers) are essential for simple and efficient data transfer into the analog computer. The separate patchboard allows the controls to be pre-patched, or control commands from external digital computing equipment may be introduced to vary the program during problem solution. These control signals may affect the operation of the analog computer or some of its components many different ways:

(1) change mode of operation
(2) change time constant of an integrator
(3) control analog memory
(4) set or monitor a potentiometer, or other components
(5) make a logical decision on an AND gate, etc.

The high accuracy solid state multipliers, resolvers and function generators are also important contributors to the overall system performance.

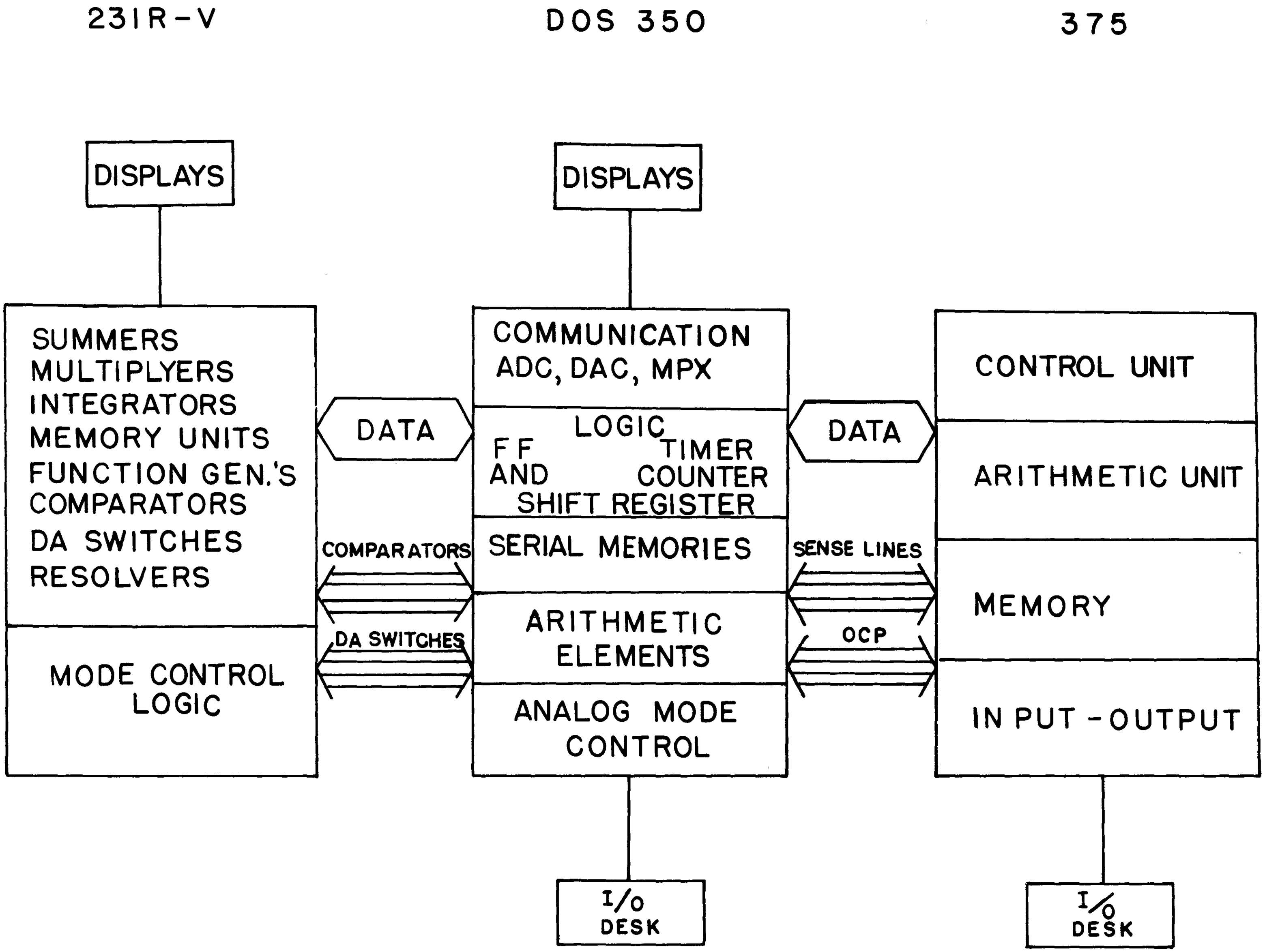


FIG. 11 HYDAC 2400 SYSTEM BLOCK DIAGRAM

B. 375 Digital Computing System

The 375 forms the arithmetic and primary memory capability of the HYDAC 2400 system. It is normally controlled by the DOS 350.

The 375 is a high-speed, all solid state general purpose digital computer with 24-bit word length and sign-magnitude binary number representation. The size of the random access memory can range from 4,096 words to 32,768 words, with a memory cycle time of 5 microseconds. The 10μsec. addition and 31μsec. multiplication times are typical of the speed of the arithmetic unit.

Standard input/output capabilities include four data input/output channels: A buffered character input channel, a buffered character output channel, a parallel 24-bit input channel and a parallel 24-bit output channel. The automatic interrupt may be used in connection with any channel desired. In addition,16 simple sense lines and 16 output control lines are provided for efficient control signal communication as will be described later.

C. The DOS 350 Digital Operations System

The DOS 350 is the control and communication center for the HYDAC 2400. Its modular, functional structure of digital computing and conversion components satisfies the stringent requirements established by the basic differences of the parallel analog and the sequential digital computers.

The logic building blocks, counters and arithmetic elements provide an extremely high speed parallel counting, decision making, control and arithmetic capability that cannot be satisfactorily provided elsewhere in the system. The same is true for the optional storage units that may be included in the DOS 350.

The communication system provides for the flow of information throughout the system. The information handled can generally be classified as either whole word data or as bits or signals that must be communicated between the sections. The conversion of whole word data is performed by the analog-to-digital and digital-to-analog converters. The appropriate channels for conversion are selected and sampled by the multiplex equipment.

In addition to the transfer of whole word data, provision must be made for the communication of logic and control signals. The sense lines enable the DOS 350 to exercise control over the digital program. The 375, through the use of a specific test instruction (SKS) is able to determine their status and make consequent decisions. When the DOS must have immediate access to the 375, an interrupt line has to be activated. When this happens, the 375 is forced to store its present computation and proceed to a program designed to handle the interrupt situation. The system has up to eight interrupt lines each of which has a specified priority in relation to the others. Similarly, the 375 also has the ability to transmit signals to the digital section. Output control pulses are generated by the 375 and terminated on DOS 350 patch panel.

The control signal exchange between the 231 R-V and DOS 350 computers is done by means of special interface components: these are the analog comparator and the digital analog (DA) switch. The analog comparator permits the DOS 350 to be made aware of the occurrence of certain events within the analog section. This circuit generates a logical ONE on the DOS 350 patchboard when the algebraic sum of two selected input voltages exceeds zero. The DA switch is basically an analog gate; under DOS control, the path of an analog signal may be open or closed. Thus the logical program is able to accomplish numerous functions, such as changing parameters, providing automatic rescalings, etc.

While the above description of the HYDAC 2400 is extremely brief, it is hoped that it would aid the reader in understanding subsequent discussions. More detailed writings on the systems are available from numerous sources.

REFERENCES

1. "Automatic Iterative Operation on an Analog Computer"
   EAI App. Ref. Lib. April 1. 3. 4a

2. "System Response Analysis with the Analog Memory and Logic System"
   EAI App. Ref. Lib. April 1. 3. 4h

3. "Applications of Analog Storage Techniques for Hybrid Computation"
   EAI App. Ref. Lib. April 1. 3. 8h

4. "Continous Data Analysis with Analog Computers using Statistical and Regression Technique" EAI PCC Report #160, 1960

5. "Two Variable Function Generator Program for HYDAC"
   EAI App. Ref. Lib. April 1. 3. 6h

6. "The Simulation of Transport Delay with the HYDAC Computing System"
   EAI App. Ref. Lib. April 1. 3. 7h

7. "Trigonometric Resolution Program for HYDAC"
   EAI App. Ref. Lib. April 1. 3. 10h

8. "Arithmatic Operations with the HYDAC Digital Operations System"
   EAI App. Ref. Lib. April 2. 4. 1h

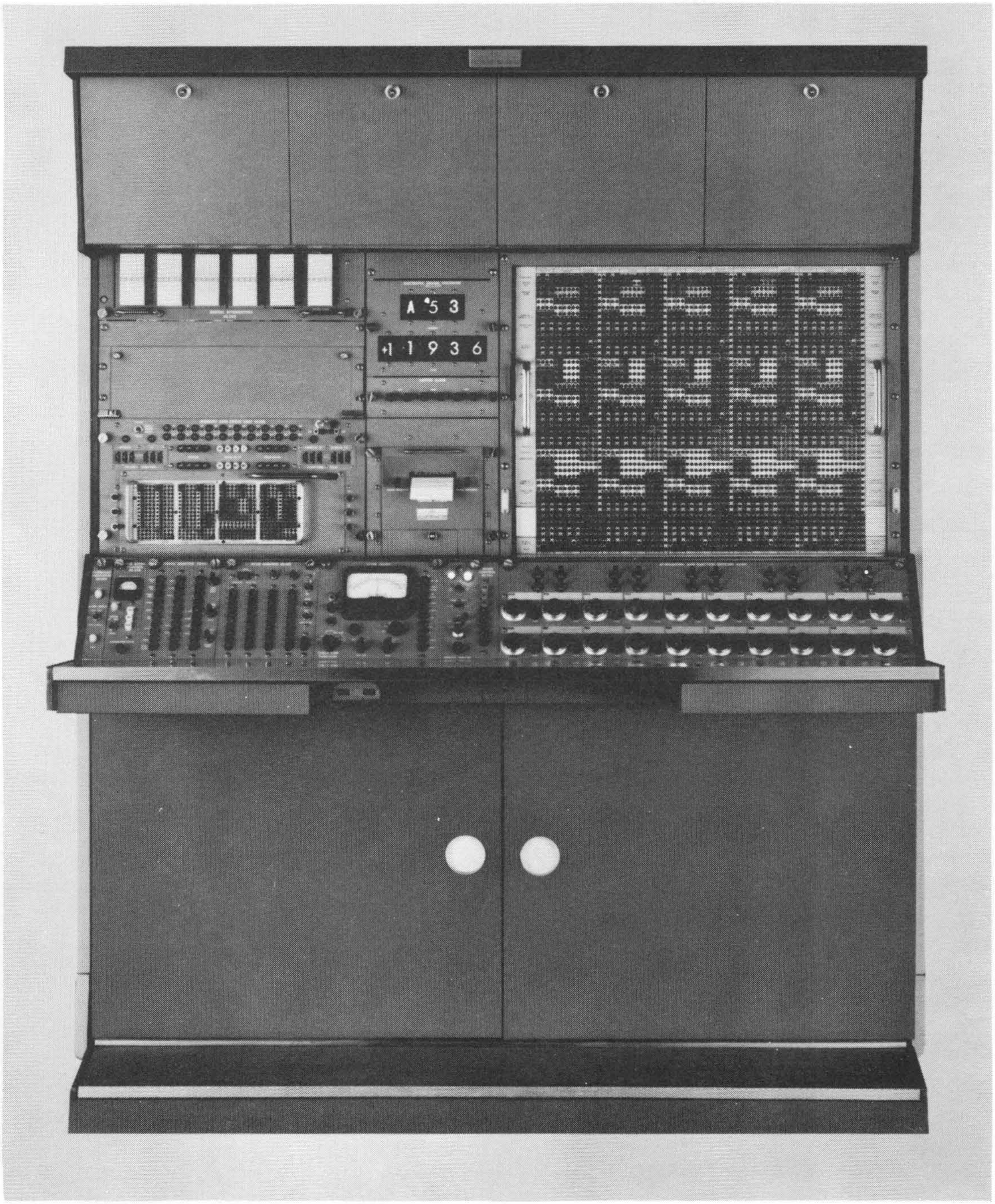
9. "HYDAC Simulation of a Terrain Avoidance Flight Control System"
   EAI App. Ref. Lib. April 3. 4. 2h

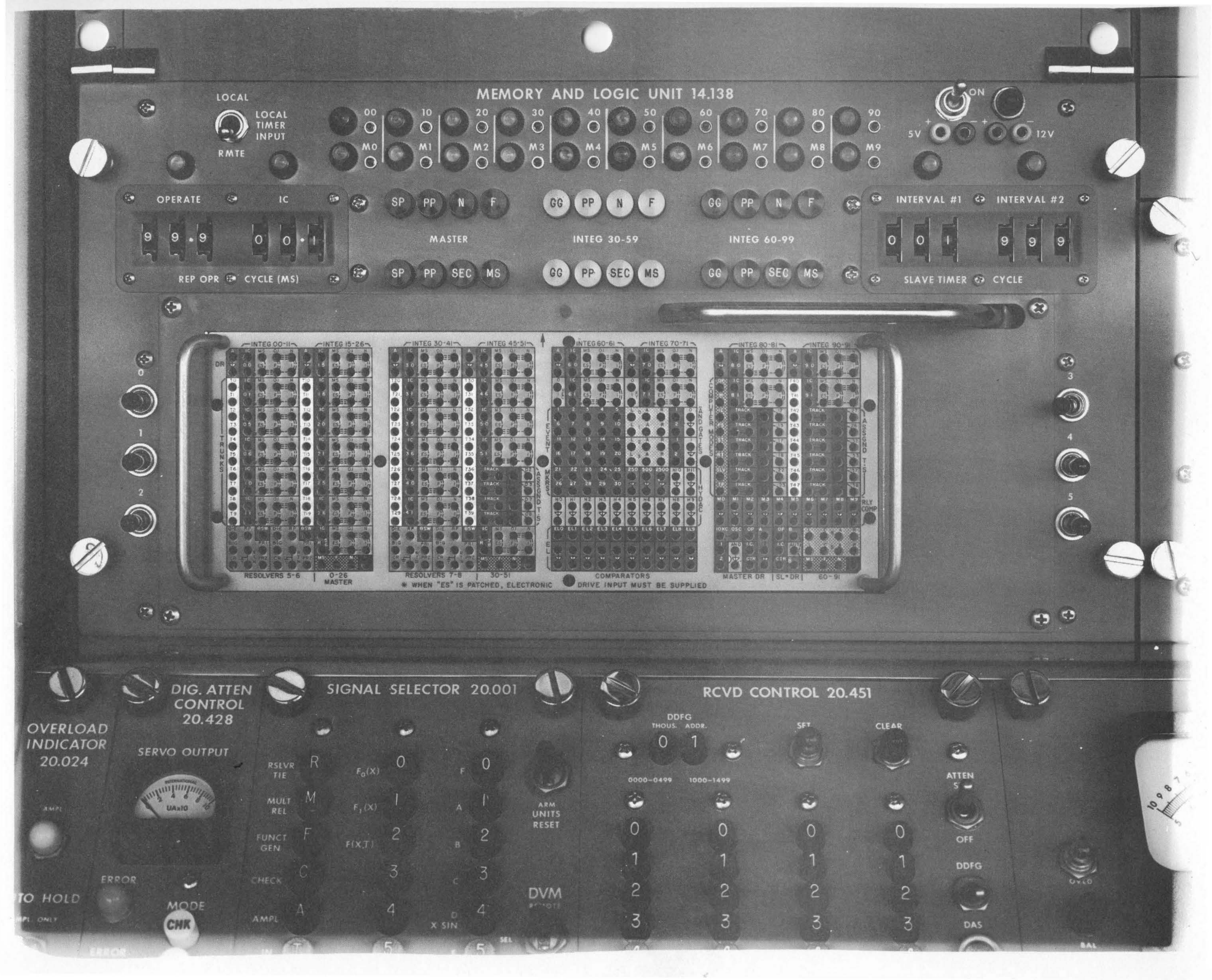
10. "Hybrid Techniques Applied to Optimization Problems"
    EAI App. Ref. Lib. April 1. 3. 2h

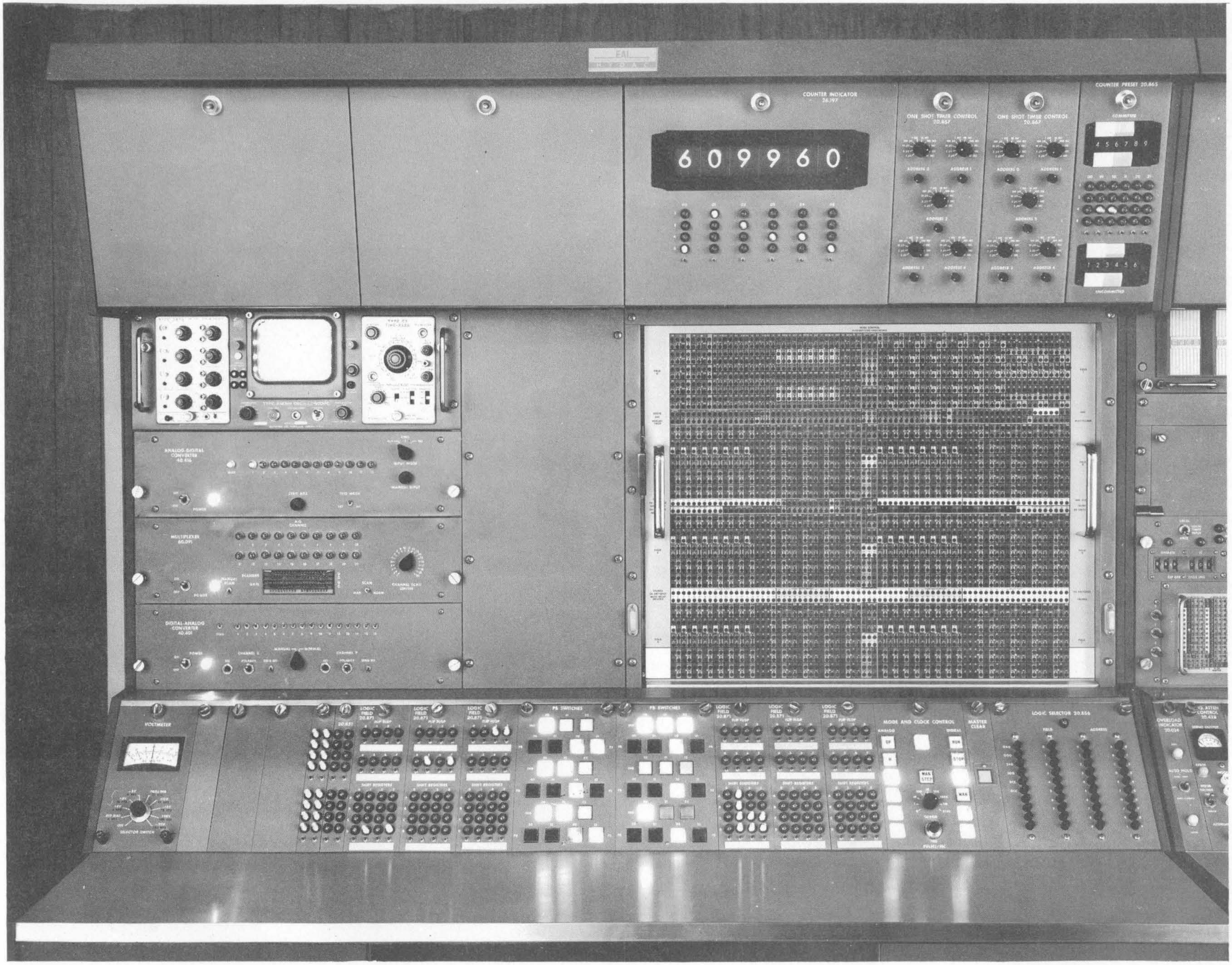
11. "Hybrid Simulation of an Aircraft Adaptive Control System"
    EAI App. Ref. Lib. April 3. 4. 5h

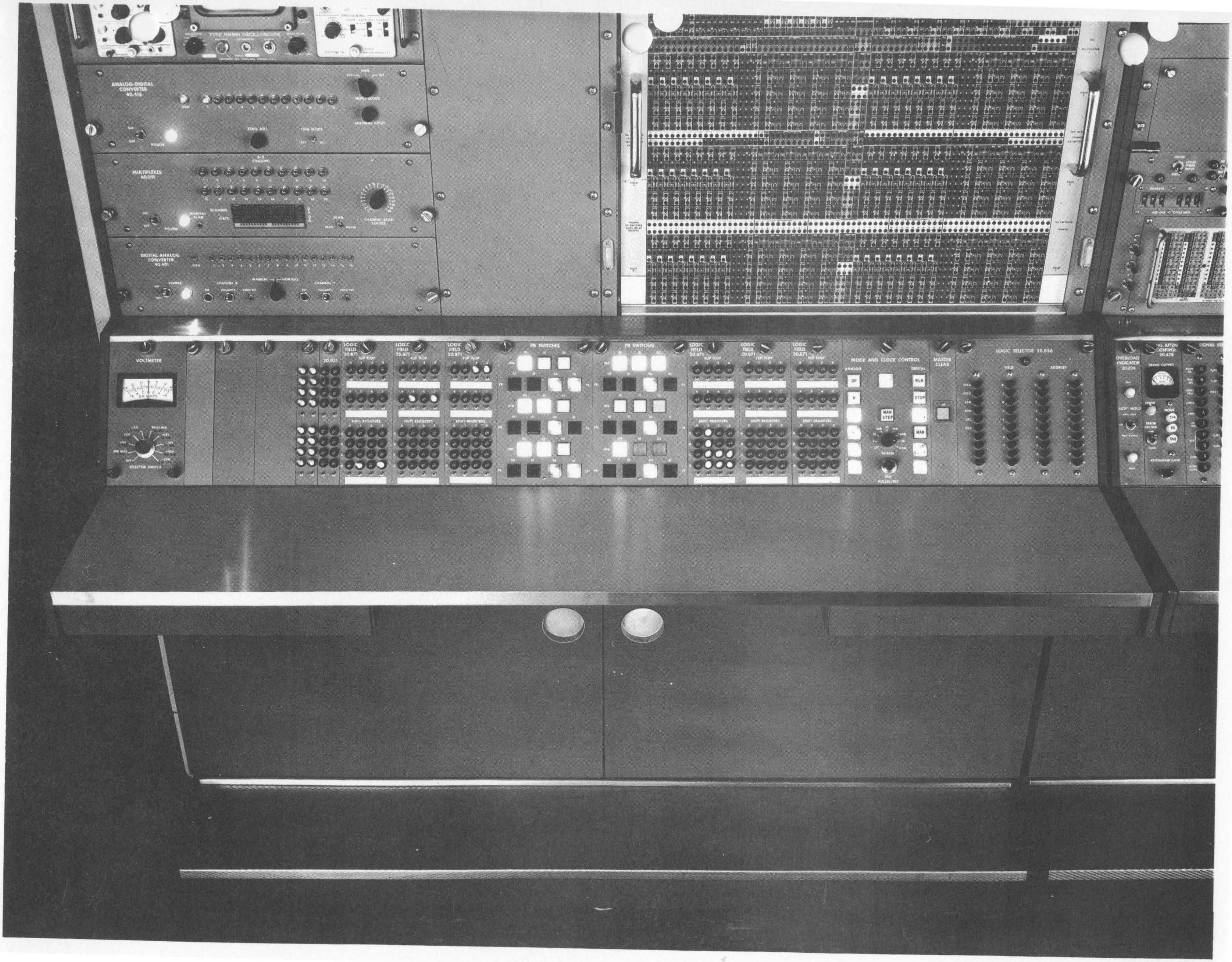
12. "Serial Solution of Partial Differential Equations on an Electronic Analog Computer" EAI PCC Report #180

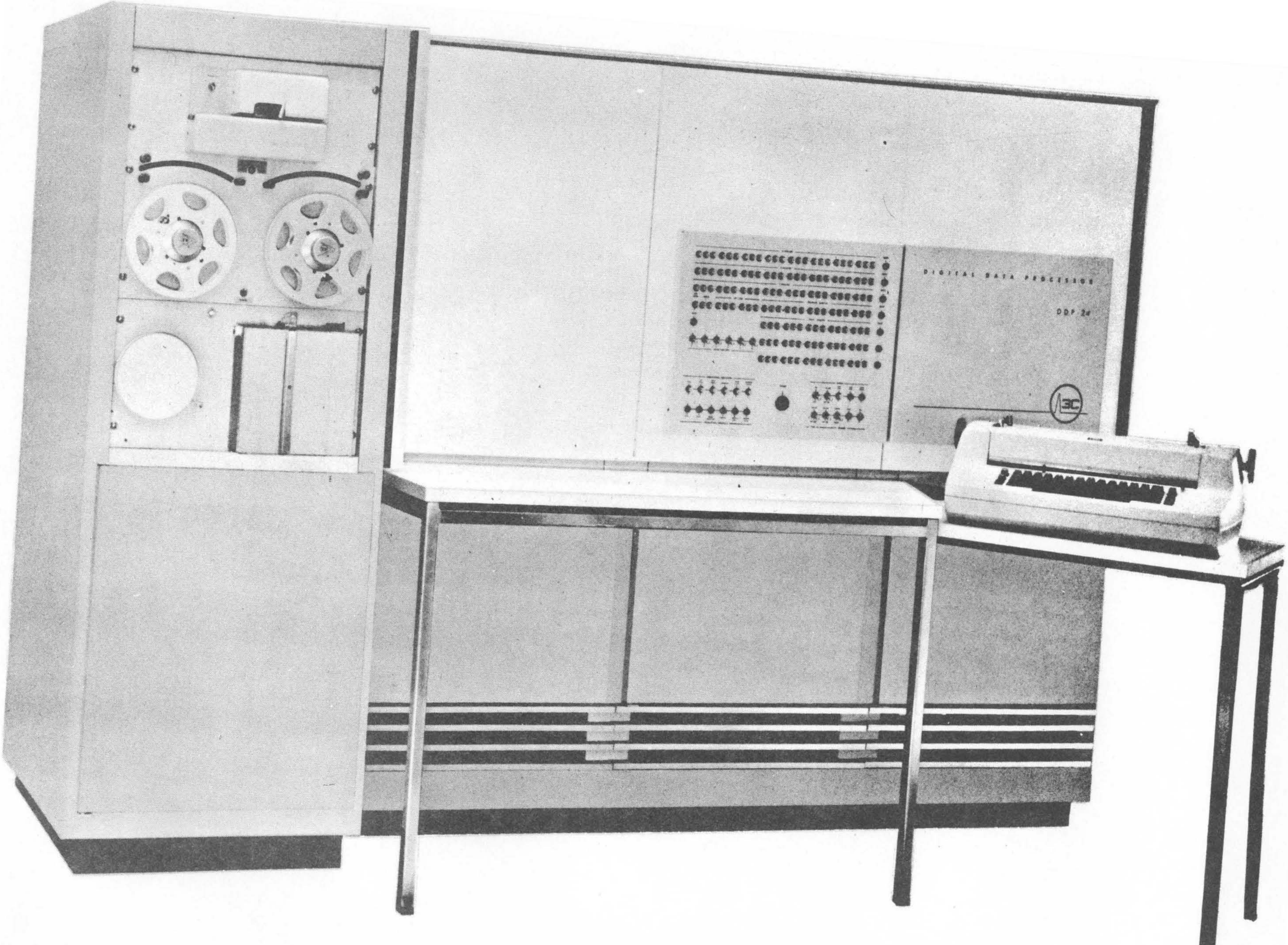
13. "Simulation of Space Vehicle with Reaction Jet Control System"
    EAI App. Ref. Lib. April 3. 4. 1h

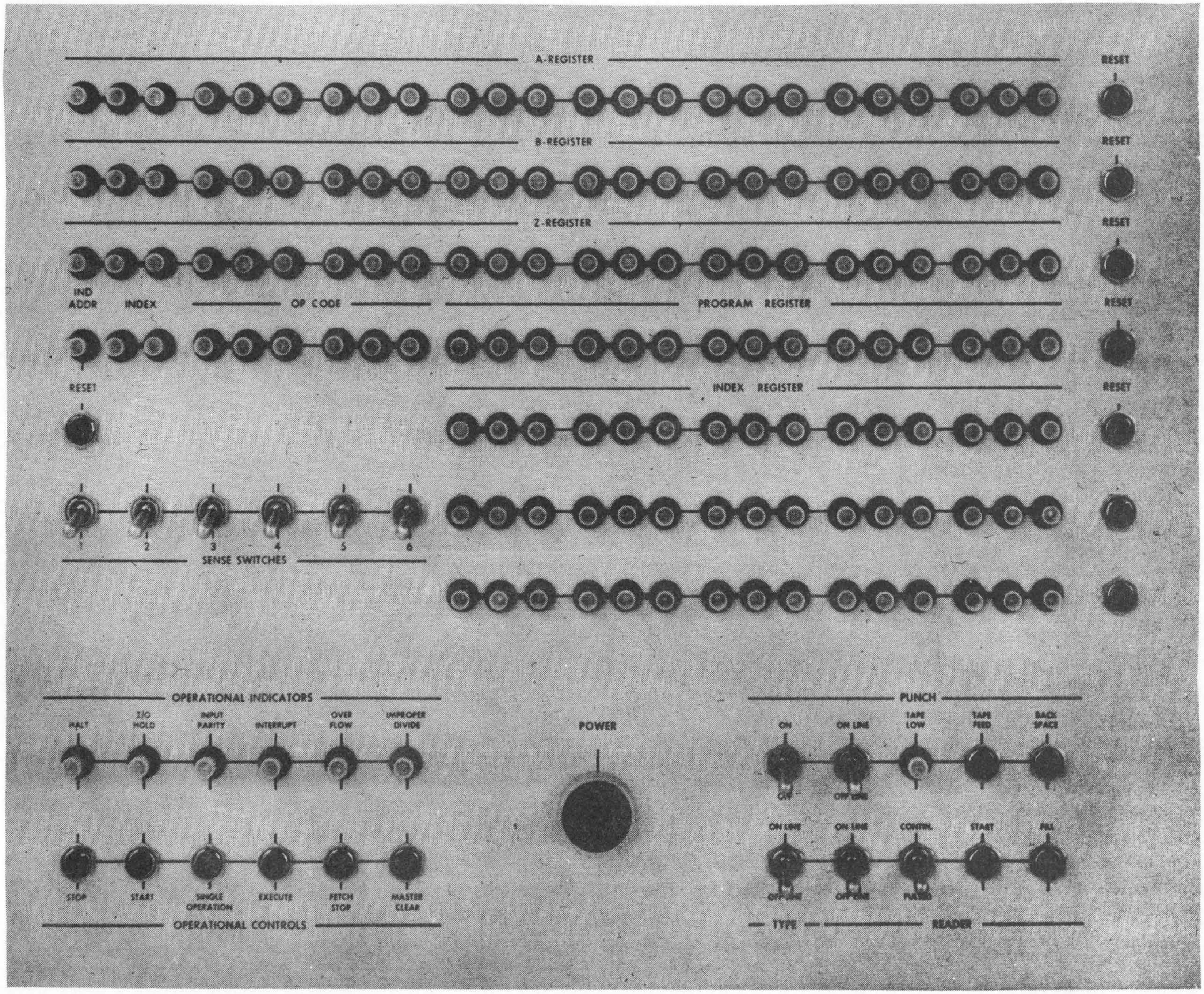

A 5 3
+1 1 9 3 6


MEMORY AND LOGIC UNIT 14.138
LOCAL
LOCAL TIMER INPUT
RMTE
00 10 20 30 40 50 60 70 80 90
M0 M1 M2 M3 M4 M5 M6 M7 M8 M9
ON
5V
12V
OPERATE
IC
REP OPR
CYCLE (MS)
SP PP N F
MASTER
SP PP SEC MS
GG PP N F
INTEG 30-59
GG PP SEC MS
GG PP N F
INTEG 60-99
GG PP SEC MS
INTERVAL #1
INTERVAL #2
SLAVE TIMER
CYCLE
INTEG 00-11
INTEG 15-26
INTEG 30-41
INTEG 45-51
INTEG 60-61
INTEG 70-71
INTEG 80-81
INTEG 90-91
TRUNKS
EVENT MKRS
ASSGND T-S
COMPUTER MODES
AND GATES
HYDAC
ASSGND T-S
RLY COMP
RESOLVERS 5-6
0-26 MASTER
RESOLVERS 7-8
30-51
COMPARATORS
MASTER DR
SL•DR
60-91
* WHEN "ES" IS PATCHED, ELECTRONIC DRIVE INPUT MUST BE SUPPLIED
OVERLOAD INDICATOR 20.024
DIG. ATTEN CONTROL 20.428
SERVO OUTPUT
UAx10
ERROR
MODE
CHK
SIGNAL SELECTOR 20.001
RSLVR TIE
MULT REL
FUNCT GEN
CHECK
AMPL
R M F C A
$F_0(X)$
$F_1(X)$
F(X,T)
X SIN
ARM UNITS RESET
DVM
RCVD CONTROL 20.451
DDFG
THOUS. ADDR.
0000-0499
1000-1499
SET
CLEAR
ATTEN
OFF
DDFG
DAS


EAL
COUNTER INDICATOR 26.197
609960
ONE SHOT TIMER CONTROL 20.667
ONE SHOT TIMER CONTROL 20.667
COUNTER PRESET 20.865
COMMITTED
UNCOMMITTED
ANALOG-DIGITAL CONVERTER 40.416
MULTIPLEXER 60.091
DIGITAL-ANALOG CONVERTER 40.401
VOLTMETER
SELECTOR SWITCH
LOGIC FIELD 20.871
FLIP FLOP
SHIFT REGISTERS
PB SWITCHES
MODE AND CLOCK CONTROL
MASTER CLEAR
LOGIC SELECTOR 20.656
FIELD
ADDRESS
OVERLOAD INDICATOR


TYPE RM561 OSCILLOSCOPE
ANALOG-DIGITAL CONVERTER 40.416
MULTIPLEXER 40.091
A-D CHANNEL
DIGITAL-ANALOG CONVERTER 40.401
VOLTMETER
SELECTOR SWITCH
20.851
LOGIC FIELD 20.871
FLIP FLOP
SHIFT REGISTERS
PB SWITCHES
MODE AND CLOCK CONTROL
ANALOG
DIGITAL
OP
RUN
STOP
MAN STEP
MAN
MASTER CLEAR
LOGIC SELECTOR 20.856
FIELD
ADDRESS
OVERLOAD INDICATOR 20.024
AUTO HOLD
SERVO OUTPUT


DIGITAL DATA PROCESSOR
DDP 24
3C


A-REGISTER
RESET
B-REGISTER
RESET
Z-REGISTER
RESET
IND ADDR
INDEX
OP CODE
PROGRAM REGISTER
RESET
RESET
INDEX REGISTER
RESET
1
2
3
4
5
6
SENSE SWITCHES
OPERATIONAL INDICATORS
HALT
Z/O HOLD
INPUT PARITY
INTERRUPT
OVER FLOW
IMPROPER DIVIDE
STOP
START
SINGLE OPERATION
EXECUTE
FETCH STOP
MASTER CLEAR
OPERATIONAL CONTROLS
POWER
PUNCH
ON
OFF
ON LINE
OFF LINE
TAPE LOW
TAPE FEED
BACK SPACE
ON LINE
OFF LINE
ON LINE
OFF LINE
CONTIN.
PULSED
START
FILL
TYPE
READER

HYBRID SIMULATION OF A RE-ENTRY PROBLEM

During the past ten years a great deal of research and development work has been conducted on various types of re-entry vehicles. Numerous techniques for guiding and controlling re-entry vehicles have been proposed. The purpose of this simulation is to evaluate the capability of a new flight control system for re-entry vehicles which by its nature is simple, reliable and inherently insures a safe re-entry. The ensuing discussion explains the general re-entry problem, the unique control system being studied, and includes a detailed discussion of the simulation equipment required and its programming.

I. General Re-entry Considerations

To obtain a better insight into the need for a simple yet reliable guidance and control system for lifting re-entry vehicles, a brief discussion of the general re-entry problem follows.

The re-entry vehicle utilized in the study is an unpowered lifting vehicle with wings, which, unlike its ballistic counterpart, is highly maneuverable and can be landed on conventional runways. (See Fig. 1)

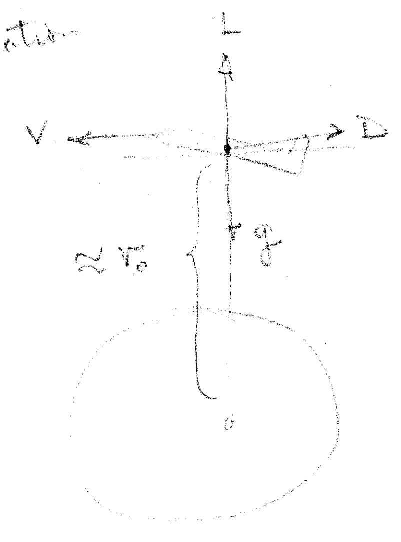
The lift and drag coefficients (C_L and C_D) of a typical high L/D lifting re-entry vehicle are shown in Fig. 2 as a function of angle of attack. A plot of lift to drag ratio (L/D or C_L/C_D) is also shown since it is of importance in determining the range of the re-entry vehicle. During the high velocity portion of re-entry flight the vehicle operates on the high C_L (larger α) side of (L/D) max. ($15^\circ < \alpha < 65^\circ$). These larger lift coefficients yield a re-entry trajectory with lower dynamic pressure, acceleration, and temperatures. Fig. 3 shows a typical uncontrolled re-entry trajectory with an angle of attack, α of 35°. Fig. 4 defines the coordinate system and symbols. During most of the vehicle flight $r_0 + h \approx r_0$ and $\gamma \approx 0$. Thus, the following approximate equations can be used:

flight path angle; see P 5

$$\ddot{h} \approx -g + \frac{V^2}{r_0} + \frac{L}{m} \qquad (1)$$

lift force

centripetal acceleration



$F_y = m a_y$

$\boxed{L - g = m\ddot{h} - \frac{v^2}{r_0}}$

$$\dot{V} \approx -\frac{D}{m} \qquad (2)$$

$F_x = m a_x$

$\boxed{-D = m\dot{V}}$

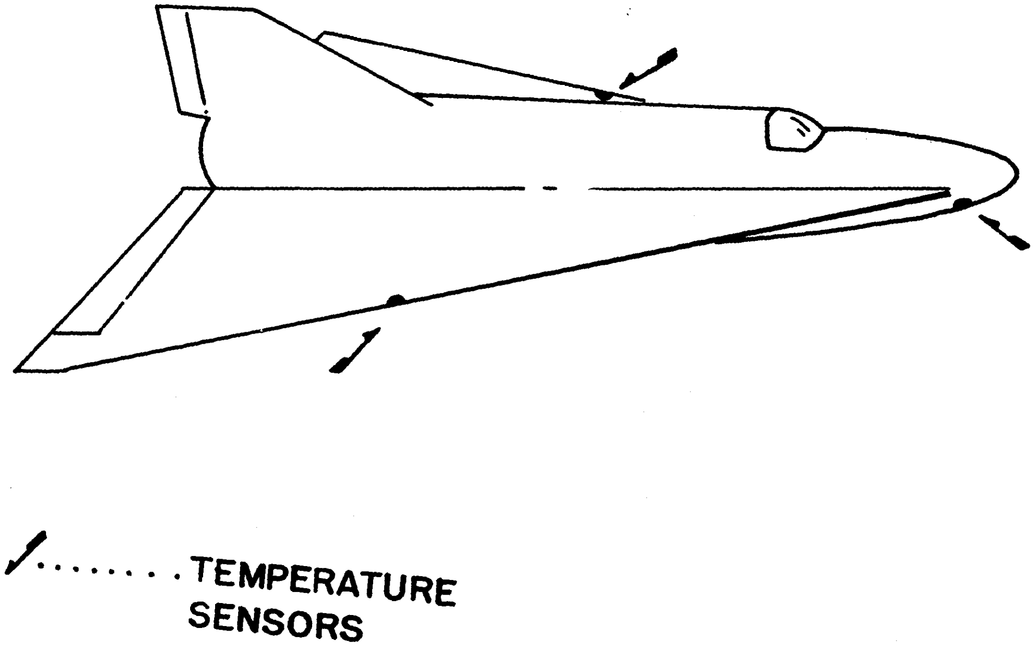


FIG.1 SKETCH OF RE-ENTRY VEHICLE

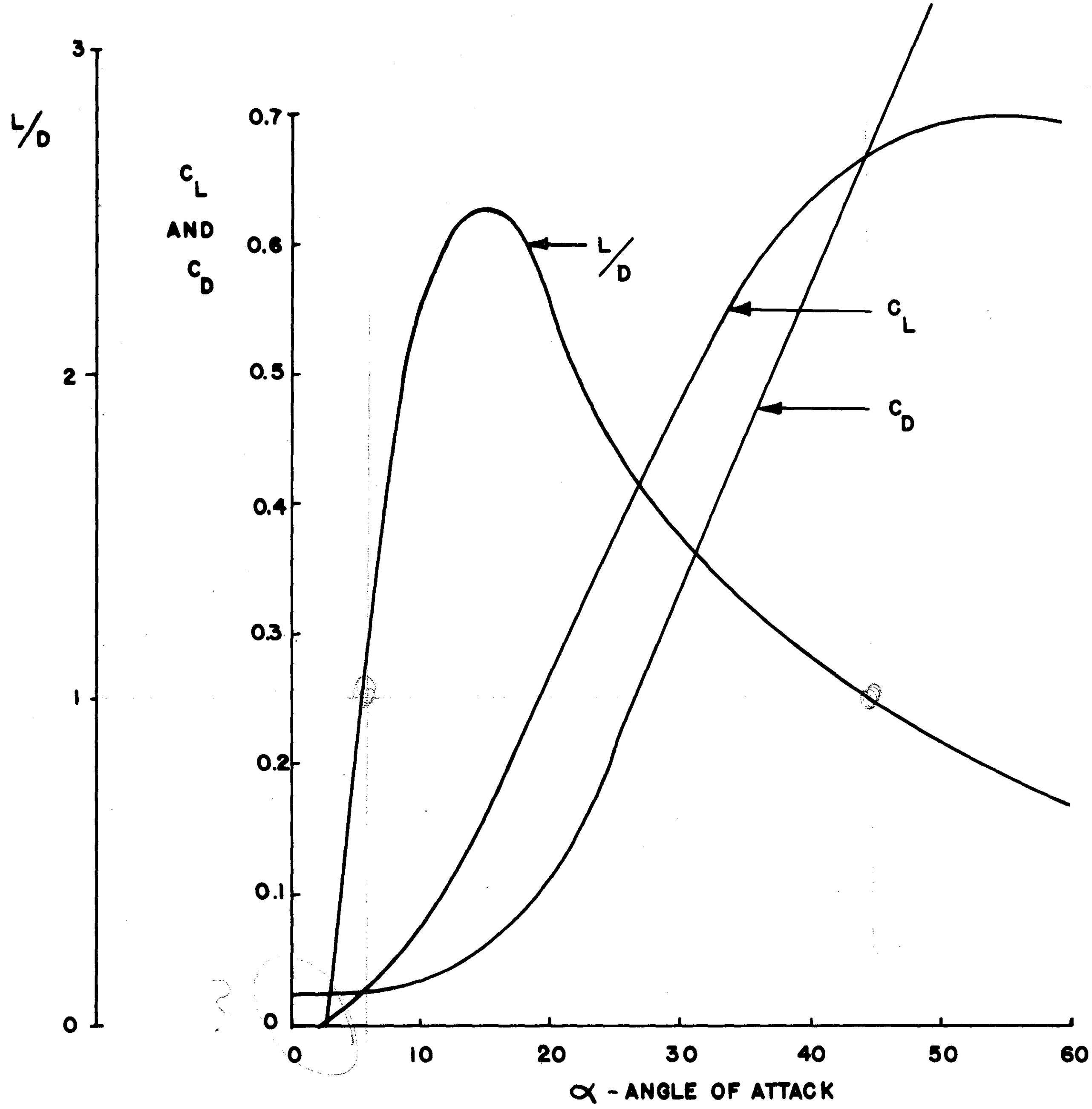


FIG.2 TYPICAL VALUES OF C_L, C_D, AND L/D FOR A LIFTING RE-ENTRY VEHICLE

3

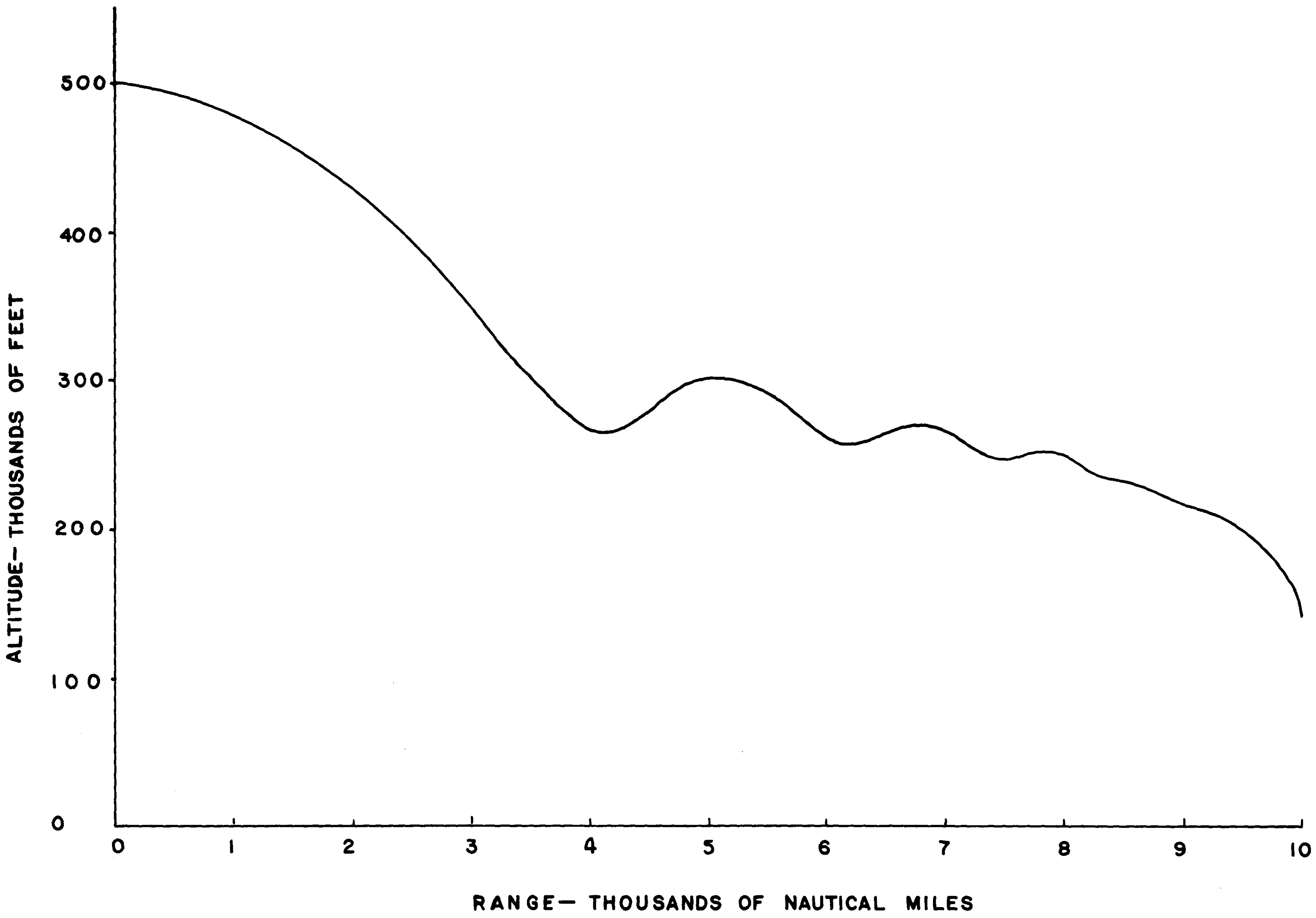
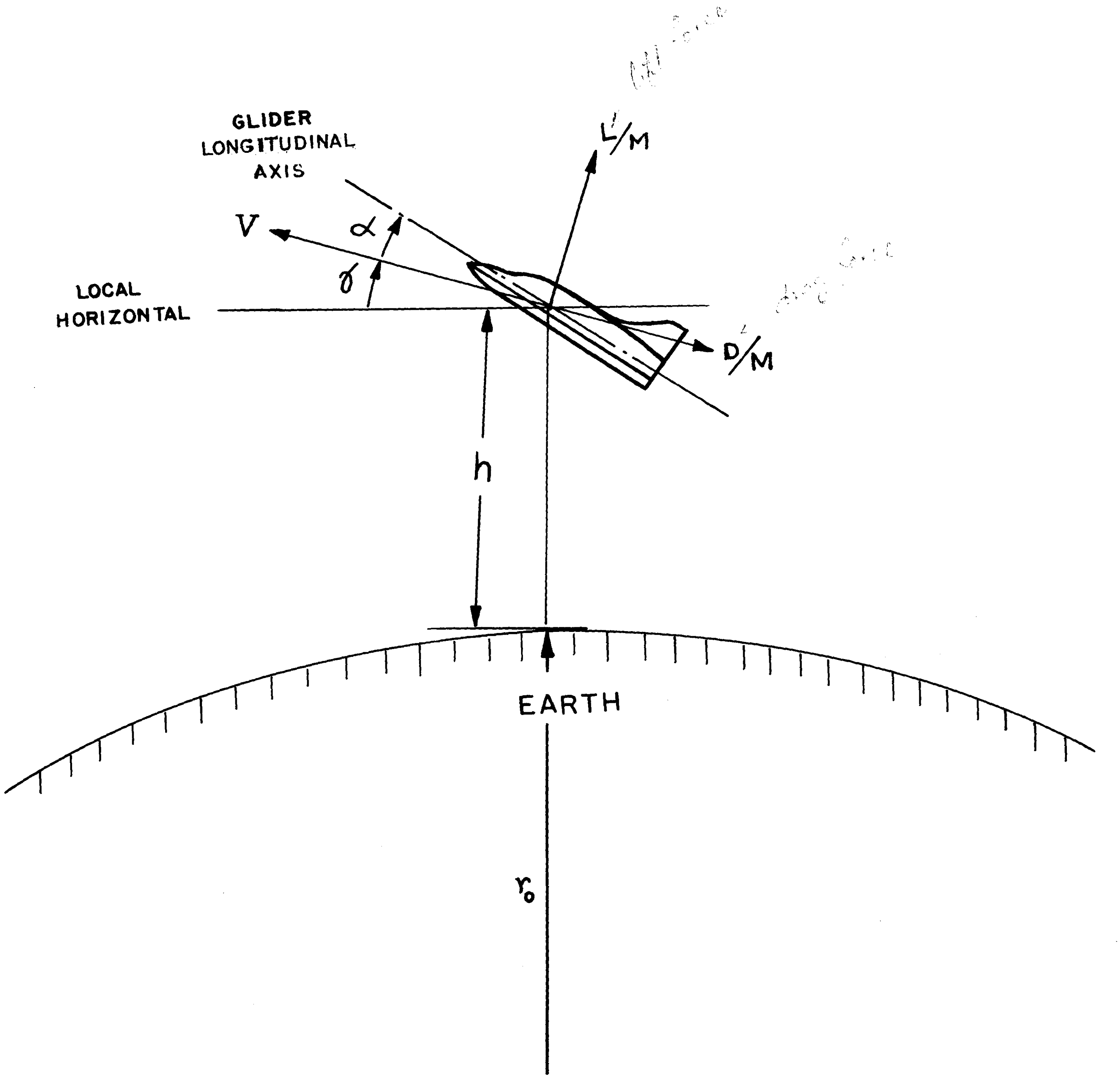


FIG.3 TYPICAL RE-ENTRY TRAJECTORY



h - ALTITUDE

r_o - RADIUS OF EARTH

$r - r_o + h \approx r_o$

V - VEHICLE VELOCITY

γ - FLIGHT PATH ANGLE (APPROXIMATELY ZERO)

α - ANGLE OF ATTACK

L/M - LIFT ACCELERATION

D/M - DRAG ACCELERATION

FIG. 4 SIMPLIFIED PLANAR RE-ENTRY COORDINATE SYSTEM

For a lifting re-entry vehicle the lift is large enough to allow the vehicle to fly along an equilibrium glide path where $\ddot{h} \approx 0$. Along the equilibrium glide path where $\ddot{h} \approx 0$,

$$\frac{L}{m} = g - \frac{V^2}{r} = q\frac{SC_L}{m} \tag{3}$$

where q is the dynamic pressure ($\frac{1}{2}\rho V^2$) and S is the wing area. Thus for a given C_L there is a unique altitude-velocity profile. Fig. 5 shows the altitude-velocity profile for two different equilibrium glide lines and the boundaries of the re-entry corridor. The lower boundary of the corridor is determined by the temperature and load limits of the vehicle and the upper boundary by the recovery ceiling. The recovery ceiling is defined as that altitude (with altitude rate, $\dot{h} = 0$) from which the vehicle can recover without exceeding the temperature and load limits of the vehicle.

One of the primary re-entry problems is to control the vehicle so that temperature and load limits are not exceeded and a smooth equilibrium glide is established. The heavy trajectory shown on Fig. 5 is a typical uncontrolled re-entry with its familiar skipping oscillations which cause the vehicle to approach dangerously close to the heat limits. The temperature rate control system being simulated, was developed to eliminate these skipping oscillations and reduce the peak temperatures during re-entry.

Another re-entry problem is to carefully manage the energy of the vehicle as it re-enters so that the desired terminal point is reached. The range capability of the re-entry vehicle can be determined quite readily in the re-entry portion of flight since the vehicle is near equilibrium glide where $\ddot{h} \approx 0$. If both sides of Equation 2 are divided by L/M or $g - \frac{V^2}{r}$ the following is obtained:

$$\frac{\dot{V}}{g - \frac{V^2}{r_0}} = -\frac{D}{L} \tag{4}$$

If $V\frac{dV}{dR}$ is substituted for $\dot{V}$:

$$dR = (L/D)\frac{r_0 V dV}{V^2 - r_0 g} \tag{5}$$

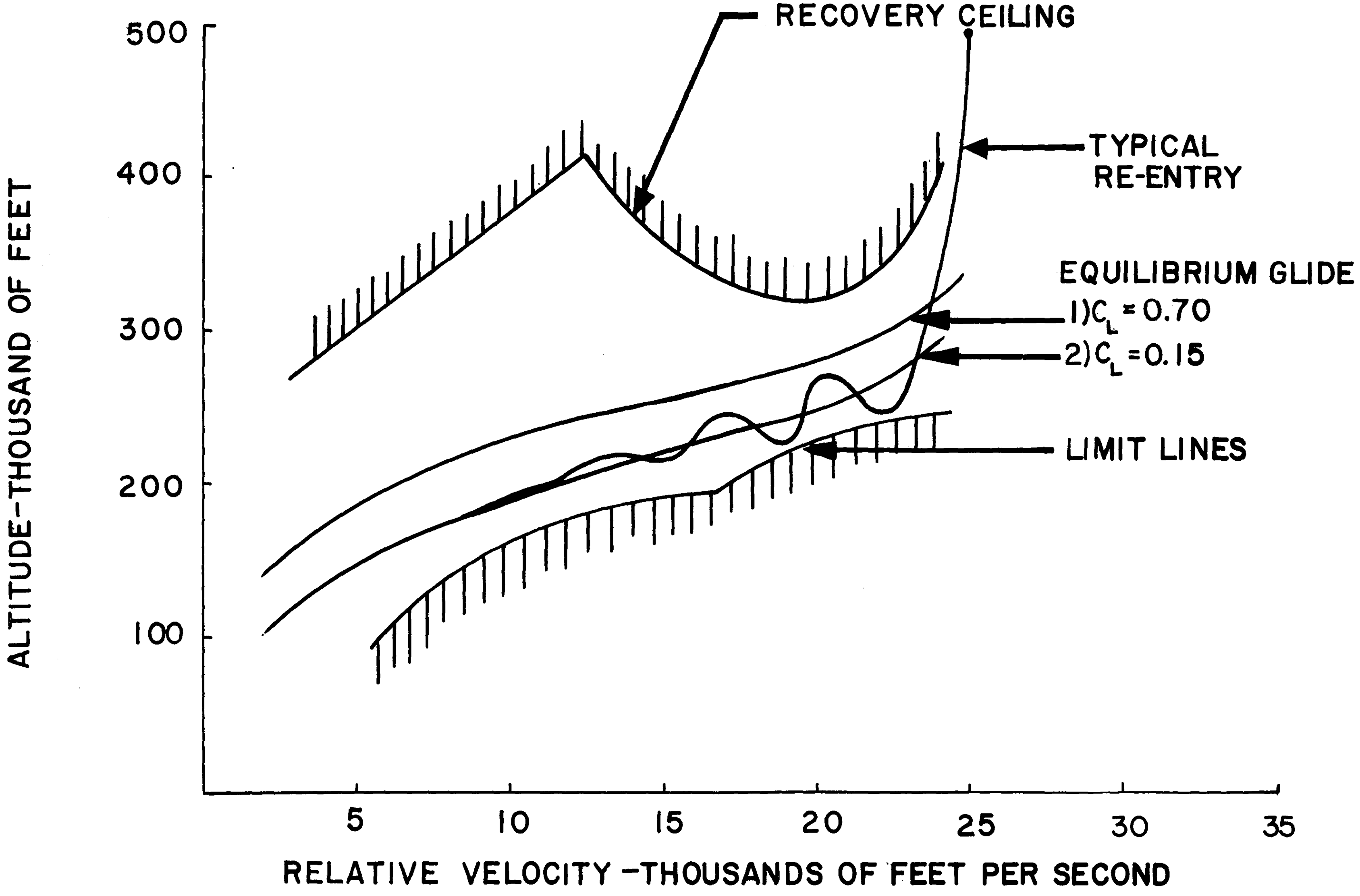


FIG.5 TYPICAL SAFE OPERATING CORRIDOR FOR A LIFTING RE-ENTRY VEHICLE

If this is integrated from an initial velocity to zero, the range is obtained:

$$R = \frac{r_0}{2}\left(\frac{L}{D}\right) \ln\left(\frac{1}{1-\frac{V^2}{r_0 g}}\right) \tag{6}$$

Thus, during the re-entry portion of flight the range is a function of velocity and L/D. Range control is thus accomplished by varying L/D, i. e. , by varying angle of attack. From Equation 6 it should also be noted that the range is very sensitive to initial velocity when the velocity is nearly equal to orbital velocity $\sqrt{r_0 g}$. For the re-entry shown in Fig. 3 the sensitivity of range to initial velocity error is approximately 80 NM/fps.

Lateral maneuverability is obtained by banking the re-entry vehicle so that the aerodynamic lift vector is rotated providing a lateral acceleration. Fig. 6 shows an energy management footprint for a typical re-entry flight. The lines of constant α and μ show what attitude must be maintained to reach a particular landing site. The dashed lines show temperature limits. This large maneuverability of a lifting re-entry vehicle requires a reliable guidance system which will perform accurately over the long re-entry and minimize errors at the desired terminal point. The temperature rate control system is being simulated to demonstrate its compatability with different types of navigation and guidance systems. As will be pointed out in the next sections the TRFCS acts as a filter to the guidance signals to insure the safety of the vehicle at all times.

II. The Statements of the Problem

A. Problem Background

The temperature rate flight control system (TRFCS), developed by the AC Spark Plug Division of the General Motors Corporation, is based upon the use of temperature sensors instead of conventional inertial instruments to provide both short-period stabilization and long-term guidance during the re-entry flight. Details of this flight control system are given in the accompanying AC Spark Plug Report. (See Reference #1) . The mathematical formulation for the simulation of the re-entry problem was furnished by AC Spark Plug.

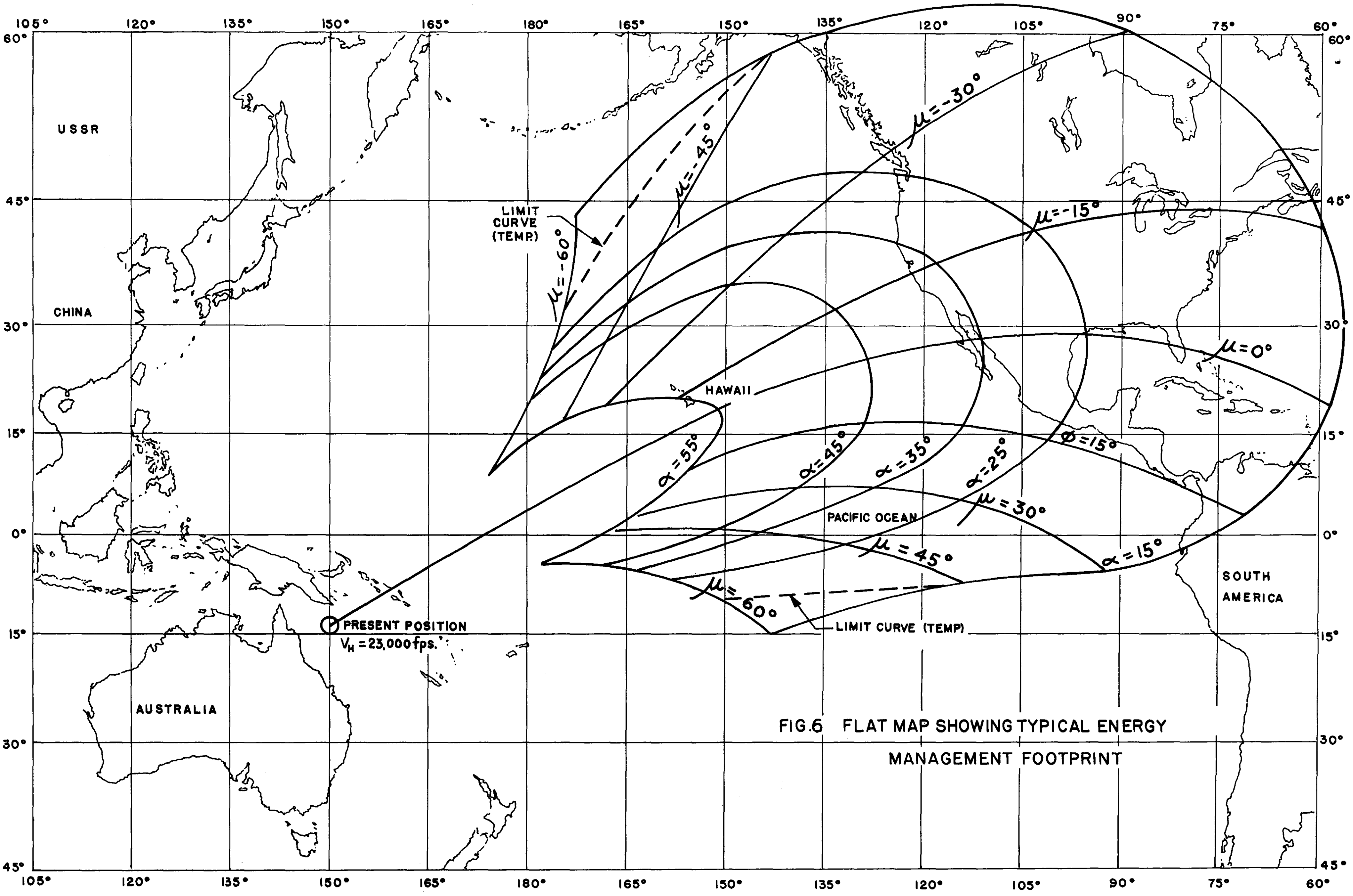


FIG.6 FLAT MAP SHOWING TYPICAL ENERGY MANAGEMENT FOOTPRINT

The new control system introduces several significant advantages:

(1) Overall vehicle safety during re-entry. This system represents an unorthodox approach to the design of an overall system of re-entry vehicles. In the standard approach the tendency has been towards a complex integrated system, while in the TRFCS, successful effort has been made to separate safety of the vehicle from the task of accurate navigation. Because of the inherent nature of temperature rate feedback and certain selected limits on the control authority, the control system minimizes skin temperature peaks. The maximum "g's" amd dynamic pressure are independent of initial conditions and maneuvers performed. This safety aspect of the TRFCS performance is entirely independent of the guidance commands and in fact, the TRFCS serves essentially as a filter for them.

(2) Simple, reliable hardware. This separation of control and guidance also results in more reliable hardware, since the failure of the necessarily complex guidance system cannot cause the complete destruction of the vehicle. Furthermore, simple thermocouple temperature sensors replace the conventional gyros and accelerometers. These sensors are used to control the flight path as well as the short-period oscillations in pitch and yaw. The only additional sensor required, besides the temperature sensors, is a vertical reference gyro, which, for the safety aspects of the re-entry, can be quite inaccurate.

(3) Both manual and automatic modes. In case of automatic guidance system failure, the TRFCS can be controlled manually. The manual flight program to be followed by the pilot, is very simple and the resulting temperature peaks, dynamic pressure, and "g" loads ---compare favorably with those obtained in the fully automatic mode.

During past years, extensive simulation studies were conducted by AC Spark Plug to obtain familiarity with the control system. A rather conventional simulation program was pursued: First analog simulations were performed to gain qualitative knowledge of the system, and to determine the practibility of this approach. Next, digital techniques were used to evaluate the accuracy of the guidance through TRFCS.

In the analog simulations, the system characteristics were split and analyzed in two independent studies. In the first, a three degree of freedom simulation of the mass center of the vehicle was combined with equations describing the short-period pitch dynamics of the vehicle. Pitch axis controls and trajectory controls in three dimensions included an approximate, simple lag representation of the lateral response of the vehicle. In the second type of simulation, the effects of lateral dynamics were obtained by simulating the dynamics of the vehicle in detail by a standard set of lateral stability equations with variable coefficients. The coefficients were varied with dynamic pressure, velocity, and stagnation temperatures of the vehicle skin, all obtained from function generators. The data for setting up the function generators came from the first type of simulation. In turn, the results from the second type of simulation were used to determine the lumped lateral response for the first type of simulation. Thus, a basis for an iterative procedure was established.

The reason for separating the simulation of the pitch dynamics and trajectory control from simulation of the lateral dynamics was the unavailability of necessary simulation equipment. The investigation of aspects of the system such as coupling between pitch and roll, could not be made with the above "split" simulation approach and awaits a complete six-degree of freedom simulation.

The next logical step was therefore to study the system's characteristics in a combined, six-degree of freedom simulation. But here arises the question of what computer or computers should be used. Past experience has shown that the conventional first-analog then digital approach is definitely not the best approach. Some of the conclusions gathered during the simulations are as follows:

(1) Repeatability of analog simulation was only marginal (50 miles in range).

(2) Slowness of digital simulation. Even for narrow ranges, determined by previous analog simulation, digital simulation was too time consuming, and therefore too expensive to optimize parameters. Reduction of the digital data also proved to be a problem.

B. Problem Objective

The main objective of the problem is to evaluate a TRFCS controlled re-entry both in automatic and manual modes of operation. To fully explore the ability of the TRFCS to control the short period attitude of the vehicle throughout the re-entry, a complete six degree of freedom simulation is required. To evaluate the ability of the TRFCS to guide the vehicle during re-entry to the desired terminal point, an accurate and repeatable simulation is required. Economy of analysis should be considered, especially in the automatic guidance studies where faster than real time simulation can be employed.

C. Computational Requirements

In order to attain the above problem objective, the following set of rigid computational requirements must be met:

(1) High accuracy in trajectory calculations for the evaluation of the guidance capability of the TRFCS.

(2) Very fast computing capability to faithfully simulate the high frequency parameters for the short-period dynamics of the vehicle.

(3) Real time and faster than real time simulation for control system evaluation. (For economical evaluation of the control system in automatic mode, the time scale should be as high as possible.).

On the basis of experience gained during past simulations, it was concluded that the simultaneous need for high accuracy and very fast computation can only be satisfied by a hybrid digital-analog computer.

Such a computer would allow the programmer to choose either analog or digital solution for different portions of the problem, trading it with fast processing for high resolution etc.

A hybrid system makes high demands on both the digital and the analog elements as well as on the control section. The tasks to be performed by the control section, however, cannot be over emphasized. Some of these tasks are the basic timing, control operations, logic decisions and conversions which preferably are parallel with other computations. Only in this way can the programmer truly utilize all elements of the hybrid to the fullest. If the digital section for "useful" computation is severely limited. To achieve these complex tasks, the "control center" should consist of programmable logic elements, such as flip-flops, counters, shift registers, parallel memory and converters. The answer to all these requirements was EAI's HYDAC 2400 system. The selection of the HYDAC system enabled the programmers to use analog and digital computational equipment judiciously and to produce the best mechanization of the problem in question.

II. Problem Mechanization

All advantages of the hybrid computer are in vain, unless very careful consideration is given to the programming of the physical system under study. This phase of the simulation transforms the essentially general purpose computer into a simulator of the specific physical system.

A. Allocation of Tasks

The first step towards a successful hybrid program is the allocation of tasks on the computer. The underlying philosophy is to subdivide the physical system into sections, and assign these to various parts of the computer, where their speed and accuracy needs are best satisfied.

As shown in Fig. 7, the physical system consists of four sections, three of them (the vehicle dynamics, the TRFCS, and the temperature sensor simulation) constitute the attitude control loop, while the vehicle dynamics, together with the guidance system and TRFCS, form the long period loop.

The assignment of these sections to various elements of the HYDAC 2400 is shown in Fig. 8. The attitude control loop, consisting of the vehicle rotational dynamics, the TRFCS and the short period sensor equations, are programmed on the analog section. In addition, the displays and cockpit simulator are tied into the analog since continuous analog signals are required. The translational equations of motion, long period heat sensor equations and guidance equations are programmed on the 375 because of the stringent accuracy requirement. The DOS 350 provides the master timing, data conversion, function generation, and reaction jet control logic.

The DOS 350 timing and control is essential because of the operational differences between the analog and digital sections. The analog is a parallel continuous computer with computing time independent of problem size. The digital is a serial, discrete interval computer with computing time directly dependent on the size of the problem. The DOS, through its timing and controls, synchronizes the calculations on each computer and controls the flow of information between sections. Function generation and the reaction control jet logic are ideally suited to the DOS since these operations can be performed rapidly in parallel with the 375-general purpose digital computer, so that the digital computation time is minimized.

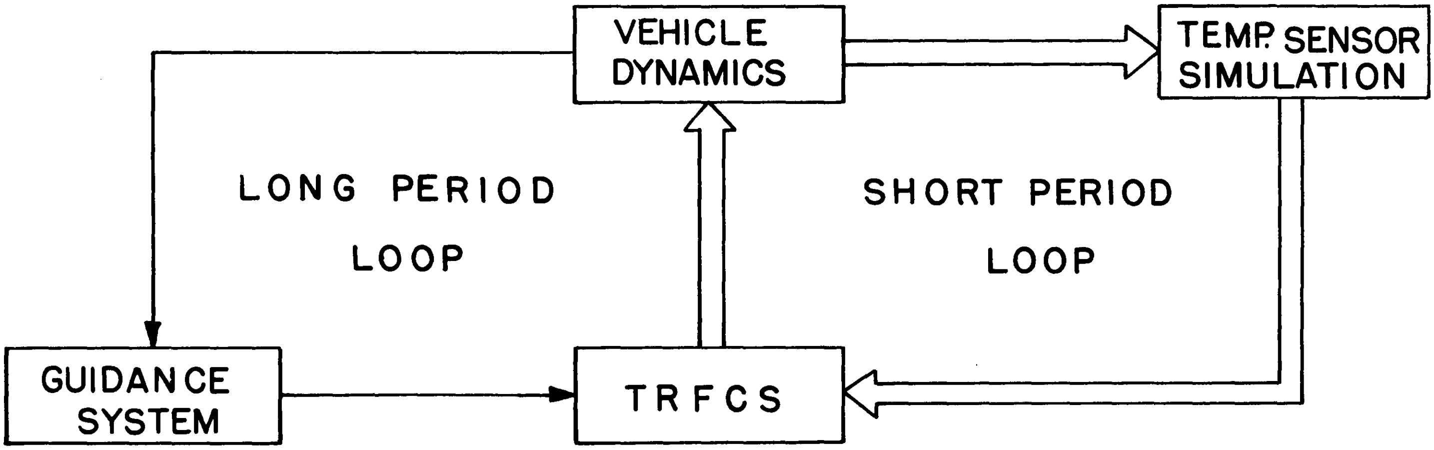


FIG.7 THE PHYSICAL SYSTEM

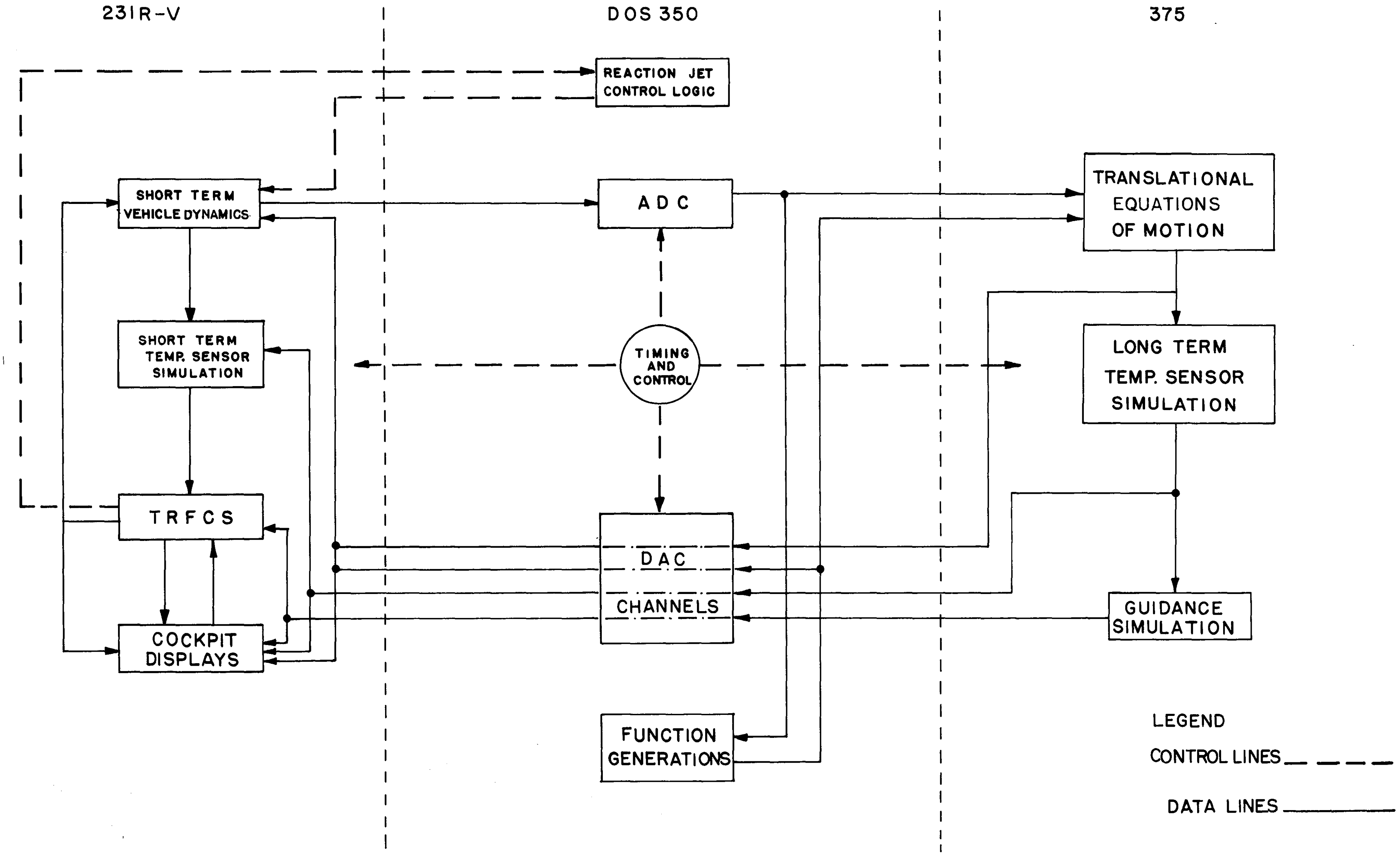


FIG.8 BLOCK DIAGRAM OF COMPLETE SYSTEM

#5

B. DOS 350 Program

Fig. 9 shows a detailed block diagram of the DOS. The DOS performs the five following functions: 1.) Timing 2.) Mode Control, 3.) Data Transfer, 4.) Function Generation, and 5.) Reaction jet control logic. These functions are described in detail in the following sections.

1. Timing

Timing is required on a hybrid computer for the following reasons:

(1) To make the mathematical time step used in the integration in the digital program correspond to the physical time scale used on the analog computer; i.e., the digital and analog sections must run in synchronism. This synchronism is accomplished by sending a periodic master time pulse T_1 which initiates the calculations for each time step in the digital computer.

(2) To time information transfers between the analog section and the digital section. Not all transfers are at the same rate, since the serial memories of the DOS are used for function generation of aerodynamic moment and force coefficients. These variables must be transferred at a high rate, since the functions are used in the short period rotational dynamics of the system. On the other hand, those variables relating to long term trajectory variations are transferred to the digital section at a lower sampling rate. Timing is clearly needed to control these two different transfer rates.

Timing is controlled from the DOS with a Master Timer which controls the sequence of events occurring per cycle of operation. The master 500 timer is a BCD counter driven by the SM8 signals occurring at the rate of ~~20~~pps (approximately one pulse every 2 ms.) Fig. 10 shows a block diagram of the master timer.

As shown schematically in Fig. 11, events are scheduled with respect to the master timing signal T_1, (every 25 SM8's), the transfers AD and DA being initiated at the specified times TAD and TDA respectively. At the end of one computation cycle the digital program jumps to the main executive loop and waits there until T_1, is received. In this manner, the digital section can be run synchronously with the analog. For instance, if T_1 occurs every 50 ms and the time scale desired is 2.0 times real time, the time step to be used in the digital program is .1 sec.

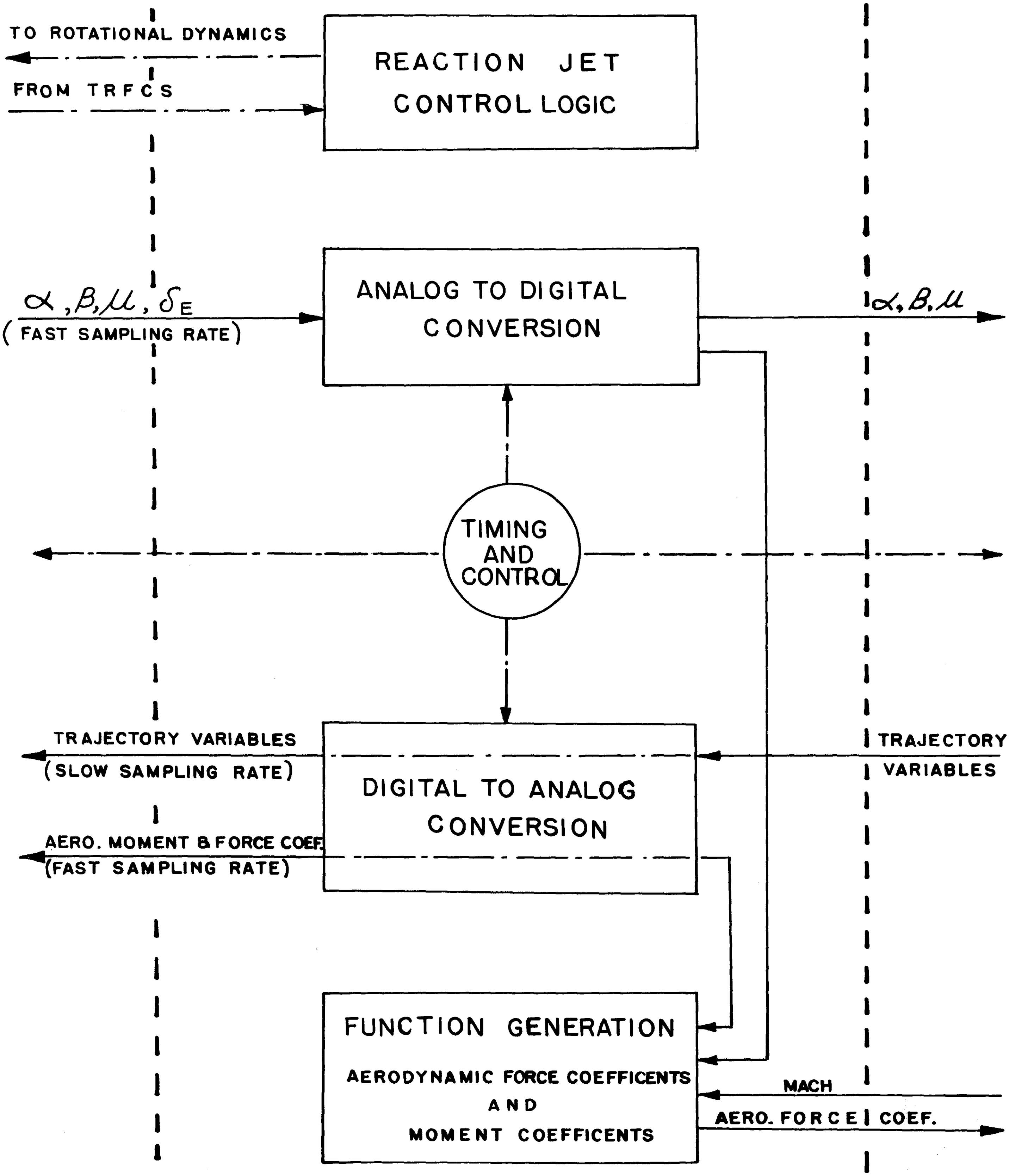


FIG.9 DOS 350 BLOCK DIAGRAM

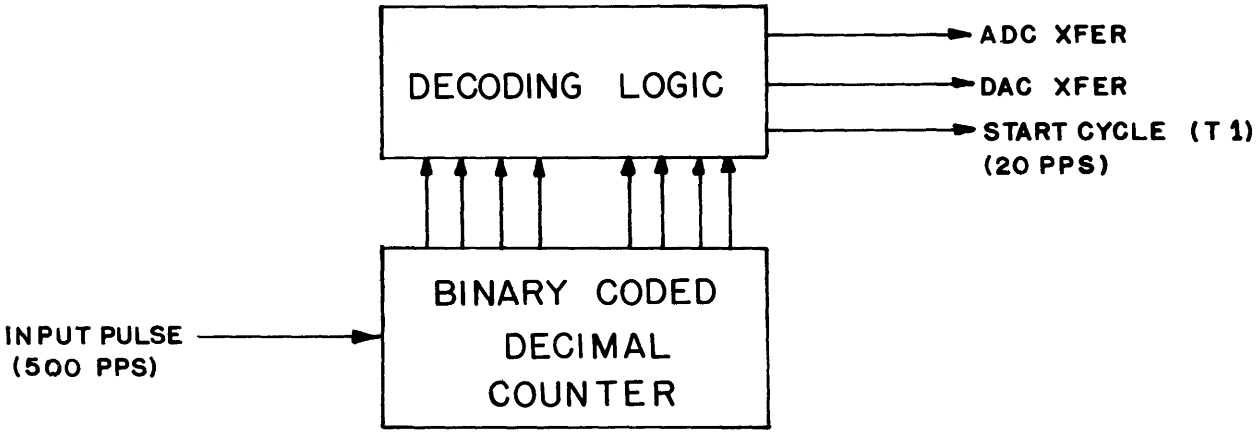


FIG.10 MASTER TIMER — BLOCK DIAGRAM

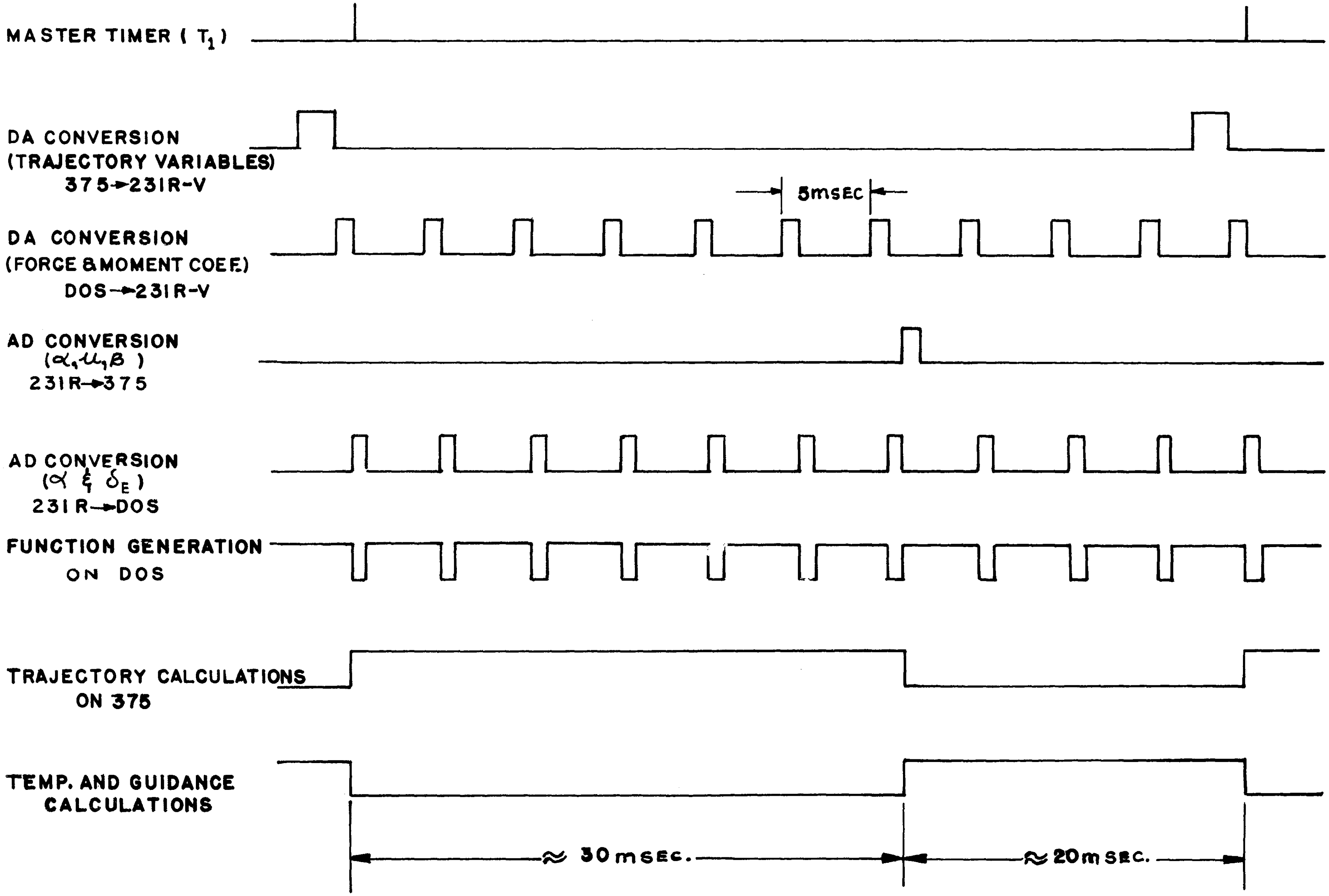


FIG.11 GENERAL TIMING OF SIMULATION

2. Mode Control

A very important function of the DOS 350 is to control the modes of operation of the system. Communication between the DOS and the 375 occurs through 16 sense lines which are set from the DOS and sensed on the digital section. In the other direction, eight flip-flops on the DOS can be set from the 375 with special OCP instructions (output control pulses). Of these, four can also be reset from the digital console.

All modes are controlled by pushbuttons from the DOS 350 and the following list summarizes the state and function of the analog and digital sections when the indicated pushbutton is depressed.

(1) IC (initial condition)

a. 231R in IC
b. 375 in IC
In IC the 375 goes through all computations with the exception of the integration routine.

(2) TIC (type in initial conditions)

a. 231R in IC
b. 375 is ready to accept new initialization data from the typewriter. From this mode the program returns automatically to the IC loop.

(3) TYTI (type titles)

a. 231R in IC
b. The 375 types out title block and line headings for the 26 variables chosen for print out.

(4) TRA (transfer only)

a. 231R in IC
b. 375 in IC

This mode is for single stepping through the DA and AD transfers and was found very useful for problem checkout. The Master Timer is stopped and now incremented only manually by pushbutton action. The data is transferred in both directions, one word at a time, each time the pushbutton TEST CONV is depressed. After completion of the DA transfer the digital program jumps to the executive waiting loop (EXWL) where it waits for another master timing signal from the appropriate pushbutton.

(5) DA Test Pattern

a. 231R in IC
b. The 375 goes directly to the DA transfer and back to the EXWL, skipping all calculations. A fixed block of data consisting of positive and negative maximum values (corresponding to ± 100v on the analog) is transferred continuously. This was found very convenient for a quick check of the DA conversions.

(6) OP (operate)

a. 231R in operate
b. 375 in operate
This is the normal mode of operation of the system

(7) TS (time scale)

a. 231R in operate
b. 375 in operate
This pushbutton changes time scales, there being two arbitrary time scales available , i. e. , real time and twenty times real time.

(8) HOLD
a. 231R in hold
b. 375 in hold waiting loop
The 375 stays in a waiting loop in the executive program until another mode is selected.

(9) DUMP

a. 231R in hold during actual dump operation otherwise in IC or operate as previously selected.
b. 375 goes to output routine at periodic intervals determined by TP the printout time interval. The digital jumps to the output routine and types out the present values of the 26 variables. The DUMP command can be given in either IC, Hold or OP and the system resumes in which ever mode it

was at the time of execution.

3. Data Transfers

The data transfer for this problem is very demanding since two basic sampling rates are required, one for the short period aerodynamic functions and the other for the long period trajectory variables. The data transfers will be discussed in two parts: 1.) the analog-to-digital conversion and 2.) the digital-to-analog conversion.

a. Analog-to-Digital Conversion

Two variables, α, and δ_E are converted every 5 milliseconds since they are used on the DOS for function generation. Once every 50 milliseconds α, β, and μ are converted and transferred to the 375 for use in the long period trajectory calculations. Fig. 11 in the previous section shows these two different types of conversions.

A block diagram of the ADC data transfer is shown in Fig. 13. Upon command from the master timing program on the DOS the ADC control logic increments the multiplexer to the proper channel and sends a convert signal to the AD converter. When the conversion is complete, the converted data is either loaded into a serial memory on the DOS for use in the function generation program or sent on to the 375 for use in the digital calculations. The data transfer from the DOS to the 375 is accomplished by loading the data into the buffer register and setting the parallel input channel ready flip flop on the digital computer. This flip-flop is enabled from the DOS by a pulse on the TE_i line prior to each A-D transfer. The 375 then inputs the data through its parallel input channel. It is thus seen that the DOS controls all the A-D conversion, thus minimizing digital computing time on the 375.

b. Digital-to-Analog Conversion

Updated values of the aerodynamic force and moment coefficients which are generated on the DOS are transferred to the analog every 5 milliseconds. The trajectory variables (altitude, temperature, guidance errors, etc.) are transferred from the 375 to the analog once every 50 milliseconds. Fig. 11 in the previous section shows the timing of these transfers.

A block diagram of the DAC data transfer is shown in Fig. 12. The digital-to-analog transfers are initiated by the DOS control logic upon command from the master timing. If data is to be transferred from the 375 the DOS sets the output channel ready flip-flop on the 375. This flip-flop is enabled from the DOS by a pulse on the TE_o line prior to each D-A transfer. The 375 will then output

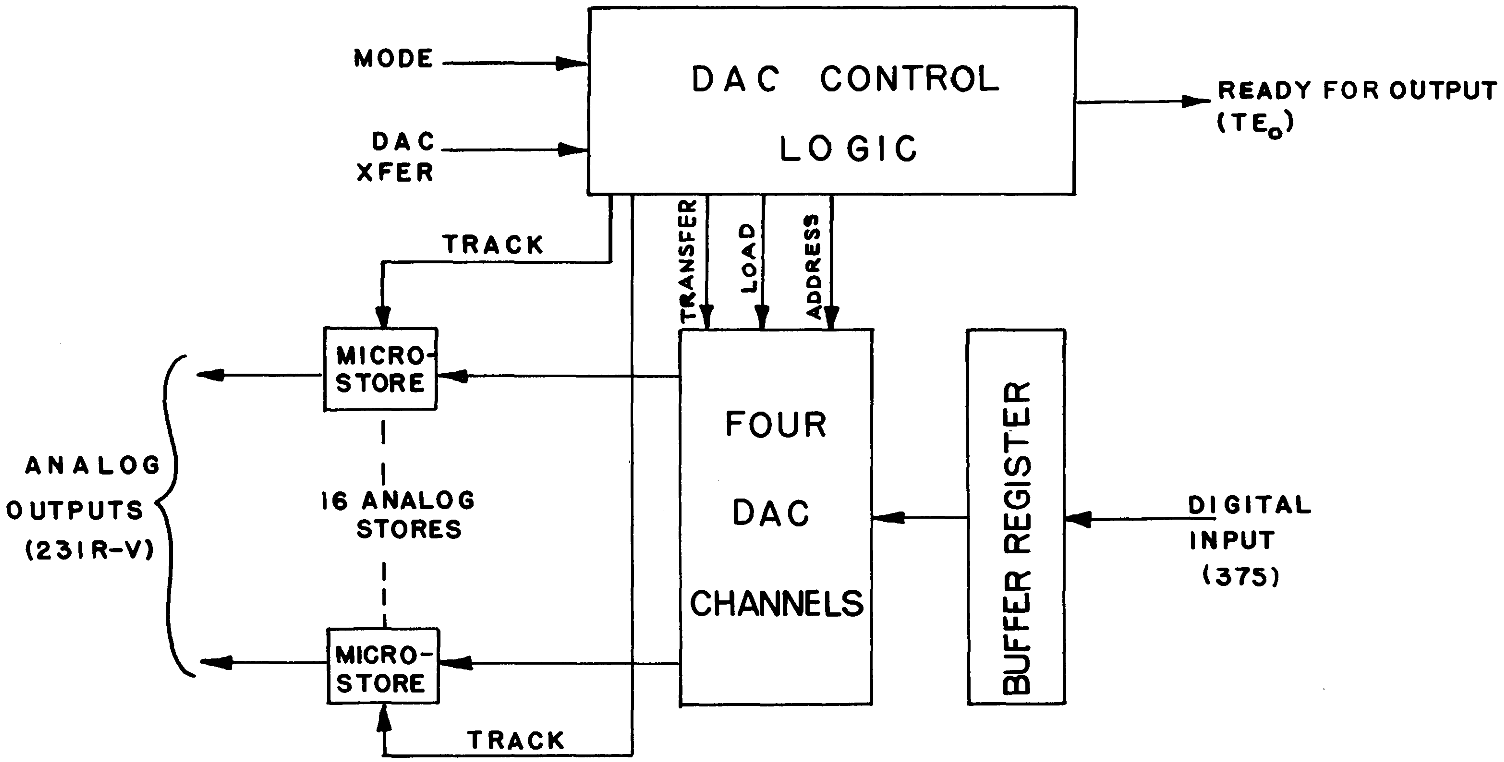


FIG.12 DAC DATA TRANSFER-BLOCK DIAGRAM

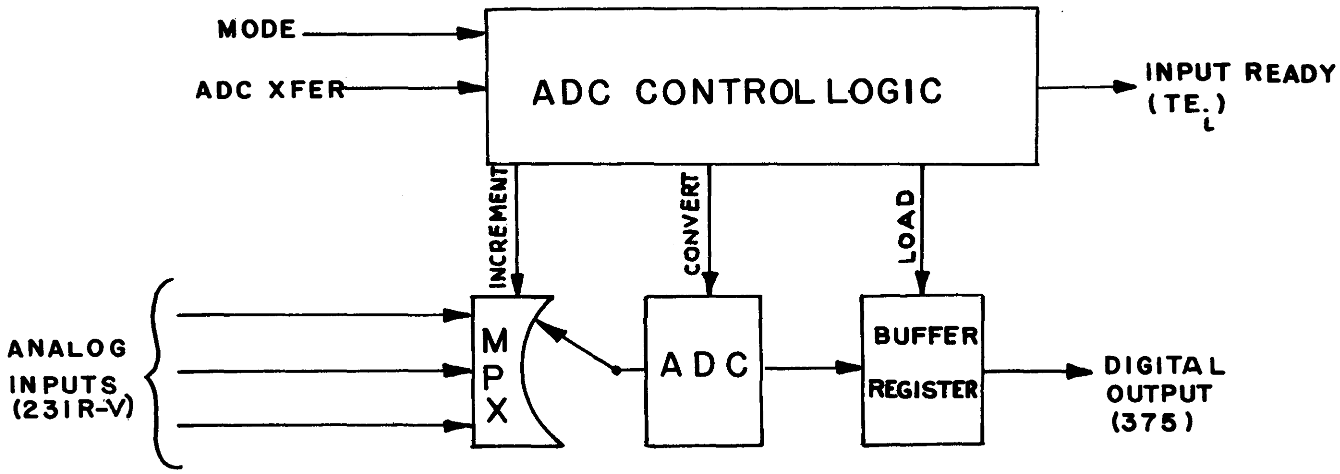


FIG.13 ADC DATA TRANSFER—BLOCK DIAGRAM

information to the buffer register and the DAC logic on the DOS loads it into the proper DA converter. To minimize digital computer time on the 375, four data words are loaded into the four DAC's (under the control of the DOS). While this data is being converted and then demultiplexed on the analog under control of the DOS, the 375 is formating the next four words to be transferred. This process is repeated until all sixteen words are transferred. Using this technique, the total processing time consumed for the transfer operation (except for formating) is only 200μsecs. The force and moment coefficients are transferred by loading the four DAC's from the serial memories which store the latest computed values of these coefficients. The data is then converted and demultiplexed on the analog and then the last four coefficients loaded. The DOS controls these conversions so that they occur at the end of each function generation period and therefore do not interfere with the conversions from the 375 to the analog.

4. Function Generation

The present simulation requires the generation of eight aerodynamic force and moment coefficients. Among these are four functions of one variable $(C_{\eta_\beta}(\alpha), C_{l_\beta}(\alpha), C_{\eta_{\delta_a}}(\alpha), C_{l_{\delta_a}}(\alpha))$, and four functions of two variables, $(C_m(\alpha, \delta_E), C_L(\alpha, M), C_D(\alpha, M), C_{Y_\beta}(\alpha, M))$. α, M, & δ_E are angle of attack, mach number, and elevator deflection respectively. These functions are generated on the DOS for the following reasons:

(1) The functions of two variables are extremely difficult to generate in the analog section and would at best require a number of sums and products of functions of one variable. The functions of one variable could be generated on the analog but were programmed on the DOS because of the ease of setup and the speed at which the functions could be changed to study other vehicle configurations.

(2) The functions are also very difficult to generate on the 375 because of the fast sampling rate required. The sampling rate on the functions should be at least 10 samples per second since they are used in the short period attitude loop of the vehicle. In the real time simulation the functions should therefore be sampled at least every 100 milliseconds and in the twenty times real time mode the functions should be sampled at least every 5 milliseconds. To meet this requirement and still use the 375, the 375 would have to be interrupted to compute these functions a great number of times during the major computation cycle. (In this problem the major computation cycle is 50 milliseconds). It is estimated that the computation time required to compute these functions on the 375 is about 4 milliseconds; therefore, if the program is stopped every 5 milliseconds for

function generation, only one millisecond of the five millisecond interval can be spent on the solution of the long period problem. Hence the major computation cycle would be 5 times longer and twenty times real time runs would be impossible.

By use of the DOS, the function generation at sample rates of 10 per sec per function can be accomplished in parallel with the digital calculations in the 375.

Fig. 14 shows a general block diagram of the function generation technique. The program for this problem utilizes 2-SM8's in series and allows 32 curves of 16 points each to be stored. Linear interpolation between points is utilized, and for the functions of two variables several curves are used with linear interpolation between them. For the present problem the four functions of utilize 1 curve each and the remaining four functions of two variables are generated with 7 curves each.

Each variable is sampled once per 2 SM8 cycle, i.e., once every 4 msec. In a 20 times real time run this would correspond to a sampling rate higher than 10 per second.

5. Reaction Jet Control System

Since portions of the flight are outside the atmosphere in regions where the dynamic pressure is too small to make use of aerodynamic surfaces for control, a reaction jet system is required. Fig.'s 15 and 16 show the block diagram of the digital controller, the jet configurations, the sign conventions and the combination of jets required to execute various commands. To make the reaction jet control easier, a pulse width modulation scheme is used which makes the moment proportional to the error signal. To prevent continuous pulsing of the jets even for small errors a deadzone is built into the controller.

Fig.'s 17 and 18 show the circuitry required in the roll axis control to test the magnitude and sign of the roll error and to provide the pulse width modulation. Fig. 19 shows the logic required to decode the commands so that the proper jets are used.

The logic operations required in the simulation of the control system are performed on the DOS in parallel to all other operations in the digital section. The simulation of such a system by conventional analog techniques is a formidable task (dozens of switches and relays would be required). The use of the 375 for such an operation would result in a significant increase in digital computation time because of the large number of logic operations and the fast sampling rate required.

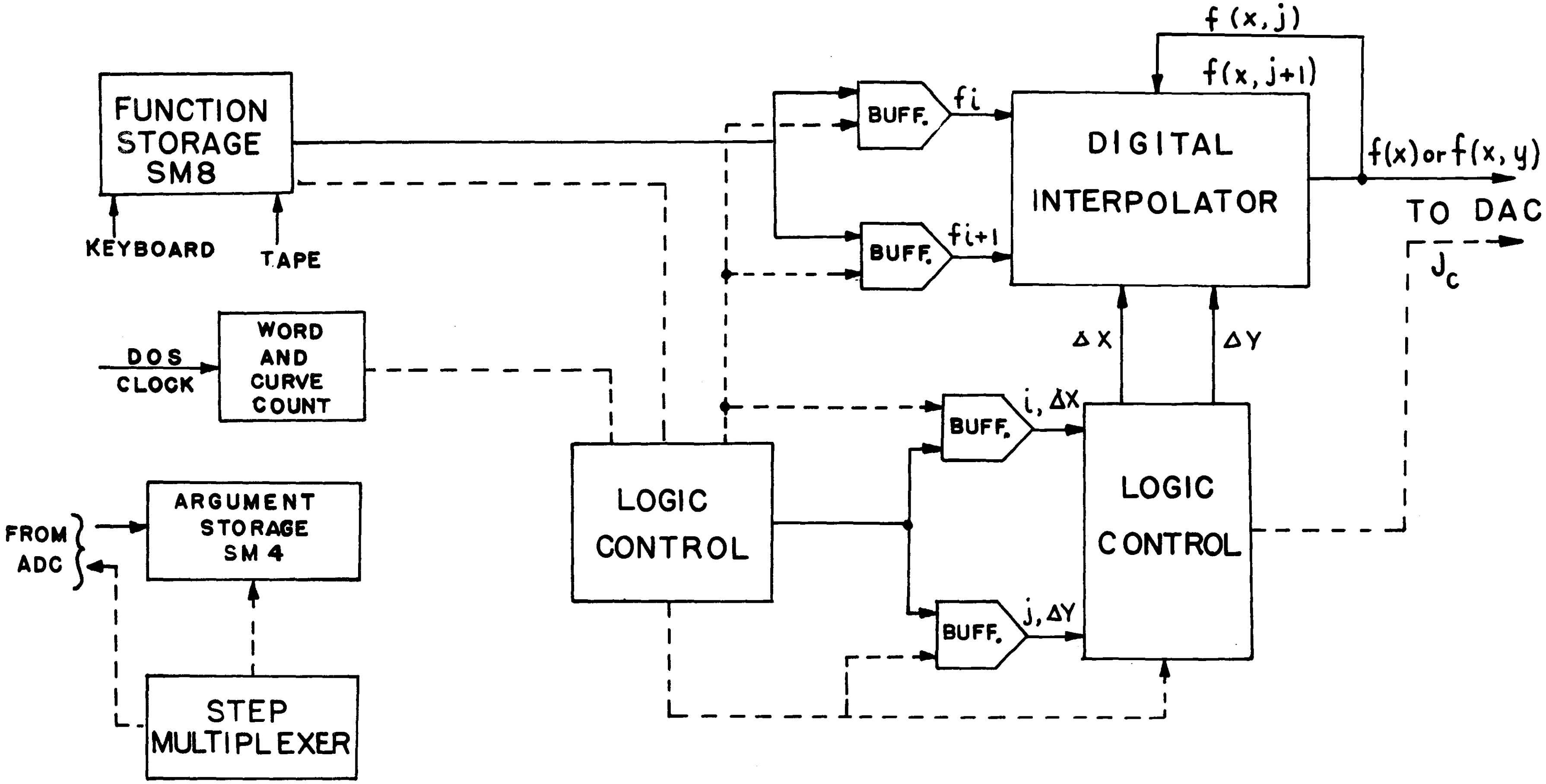


FIG. 14 BLOCK DIAGRAM OF TWO VARIABLE FUNCTION GENERATION

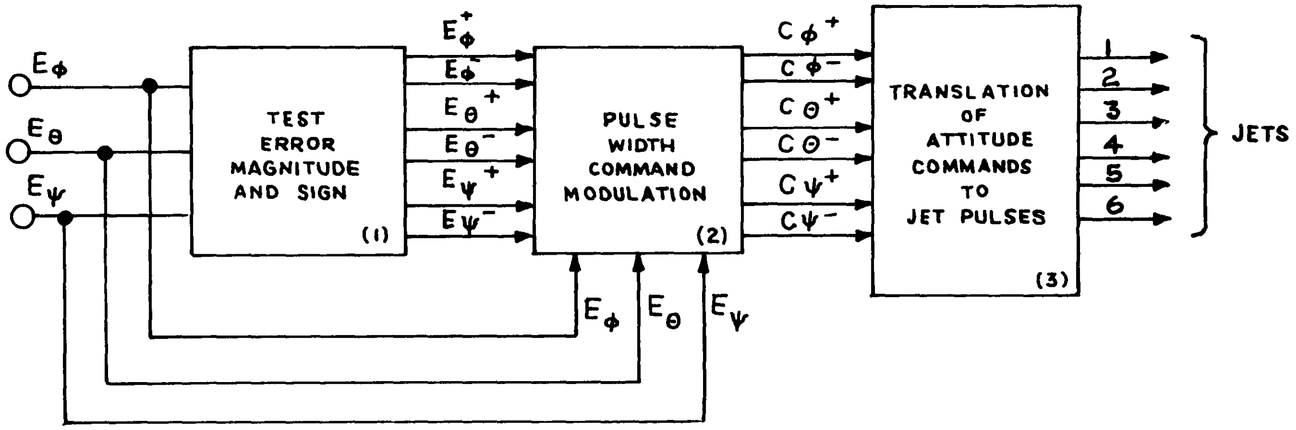
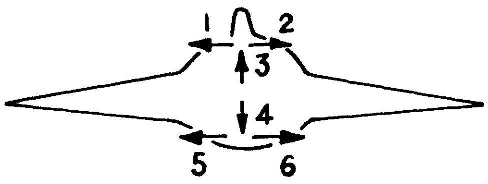
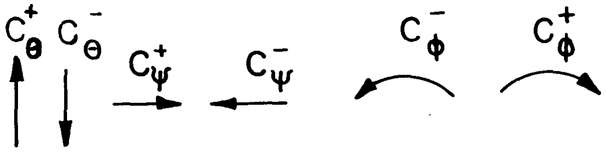


FIG.15 BLOCK DIAGRAM OF DIGITAL CONTROLLER



ARRANGEMENT OF JETS



COMMAND CONFIGURATION

COMMAND	JETS	COMMAND	JETS
C_θ^+ PITCH UP	3	C_ψ^+, C_ϕ^-	6
C_θ^- PITCH DOWN	4	C_ψ^+, C_ϕ^+	2
C_ψ^+ YAW RIGHT	2, 6	C_ψ^-, C_ϕ^-	1
C_ψ^- YAW LEFT	1, 5	C_ψ^-, C_ϕ^+	5
C_ϕ^+ ROLL RIGHT	2, 5		
C_ϕ^- ROLL LEFT	1, 6		

FIG.16 RELATIONSHIP BETWEEN COMMANDS AND JETS

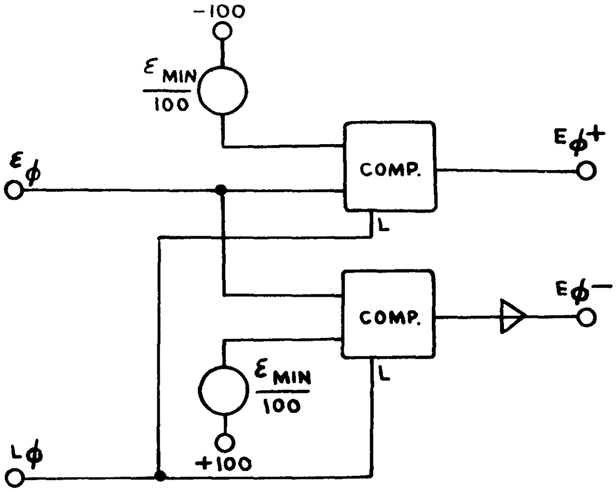


FIG.17 CIRCUIT FOR TESTING ROLL ERROR MAGNITUDE AND SIGN

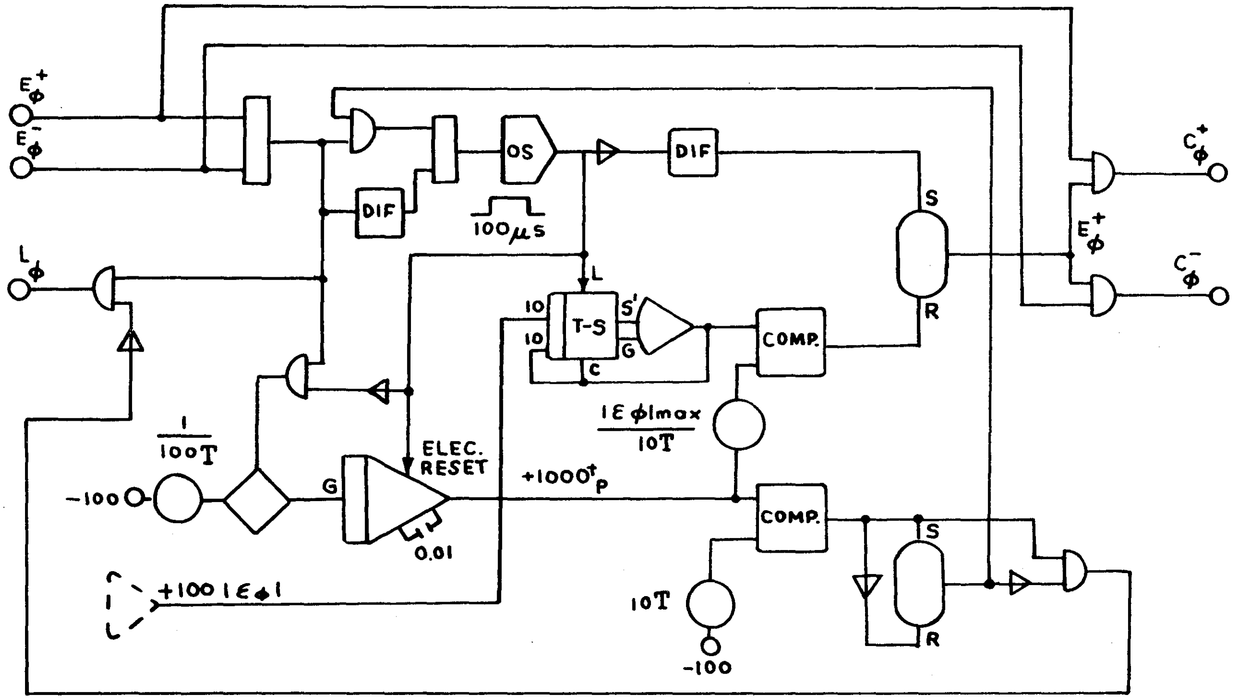


FIG.18 PULSE WIDTH COMMAND MODULATION CIRCUIT FOR ROLL ANGLE

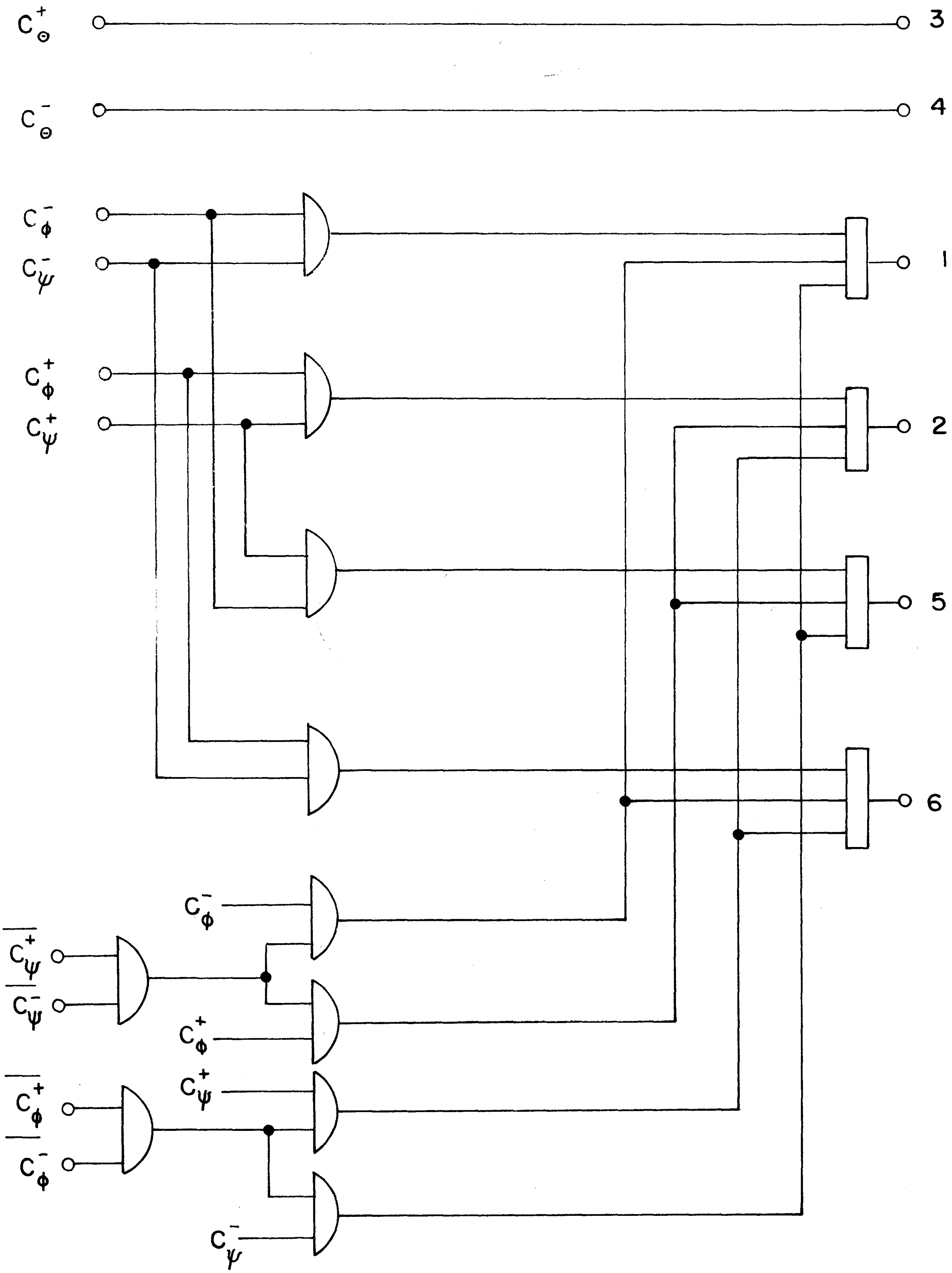


FIG.19 COMMANDS—TO—JETS LOGIC

C. Digital Calculation on the 375

The digital program was written to make the digital calculation time as small as possible and to stay withing a memory capacity of 4, 000 words. The material which follows gives a description of the equations which are solved on the 375 and the details of the digital computer programming.

1. Summary of the Digital Calculations

Fig. 20 shows the block diagram of the system of equations to be solved on the 375. A complete summary of the equations and symbols is given in Appendix A.

The three degree-of-freedom translational equations of motion are solved in a local horizontal coordinate system with axes along the north, east, and radial directions. (See Fig. 21) The gravity and altitude calculations are based upon an oblate model of the earth, and the U. S. Standard Atmosphere, 1962, was stored in table form on the 375. Most of the equations are conventional and quite straighforward, with the exception of the heat transfer, temperature, and guidance equations which will be discussed in more detail.

The heat transfer and temperature equations are as follows:

$$\dot{q}_{cs} = \frac{2.70893}{\sqrt{R}} \sqrt{q} \left(\frac{V}{10^4}\right)^2$$

$$\dot{q}_{RS} = 144.9 \, R \, \rho^{1.57} \left(\frac{V}{10^4}\right)^{17}$$

$$\dot{q}_c = \dot{q}_{cs} \cos^{1.5} \alpha$$

$$\dot{q}_R = \dot{q}_{RS} \cos^6 \alpha$$

$$T_s = \left[\frac{1}{\epsilon \sigma_B} \left(\dot{q}_{RS} + \dot{q}_{cs}\right)\right]^{1/4}$$

$$T = \left[\frac{1}{\epsilon \sigma_B} \left(\dot{q}_R + \dot{q}_c\right)\right]^{1/4}$$

$$\dot{T}_s = \frac{T_s}{4}\left(3\frac{\dot{V}}{V} + \frac{\beta e \dot{r}}{2}\right)$$

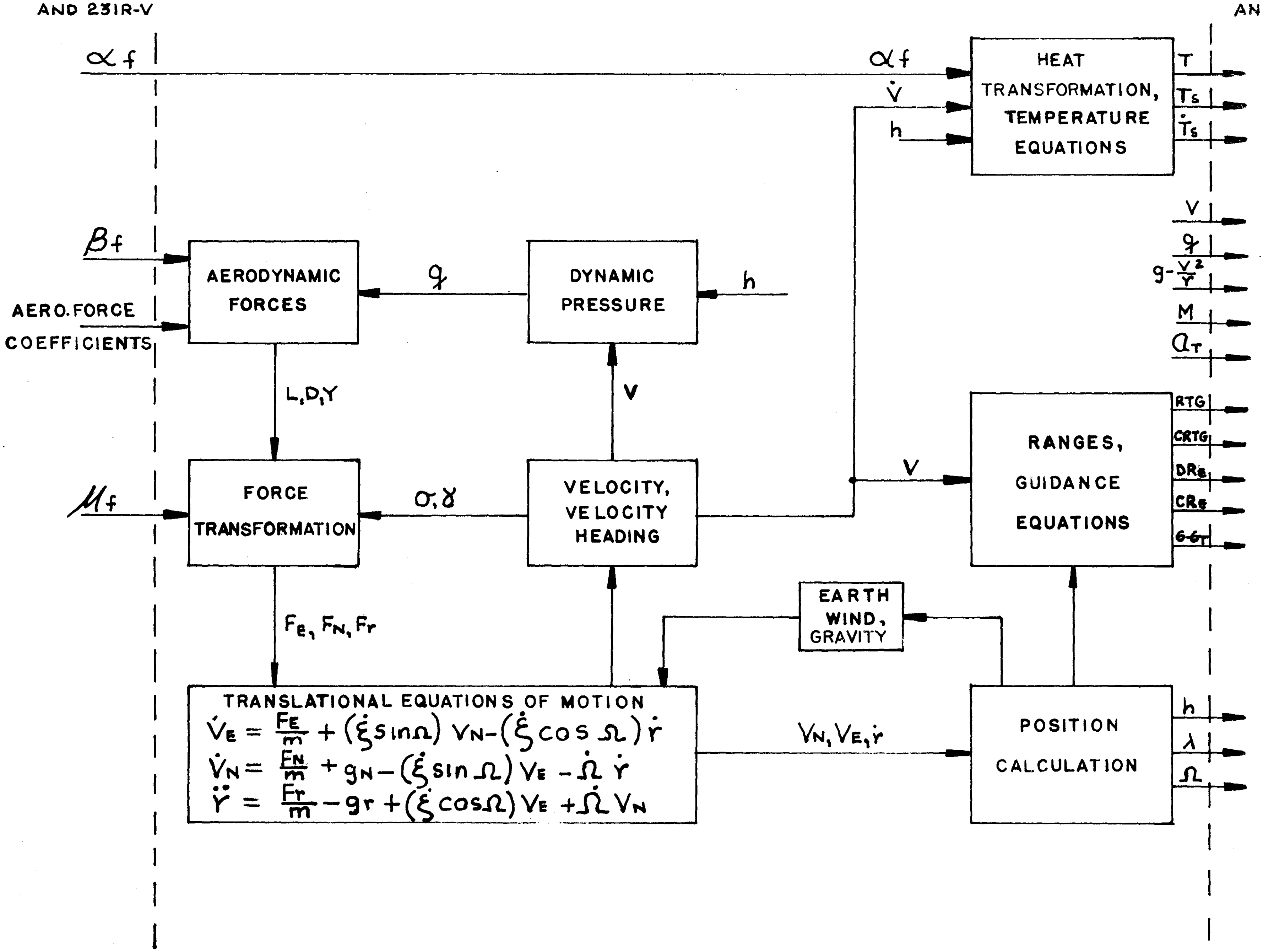
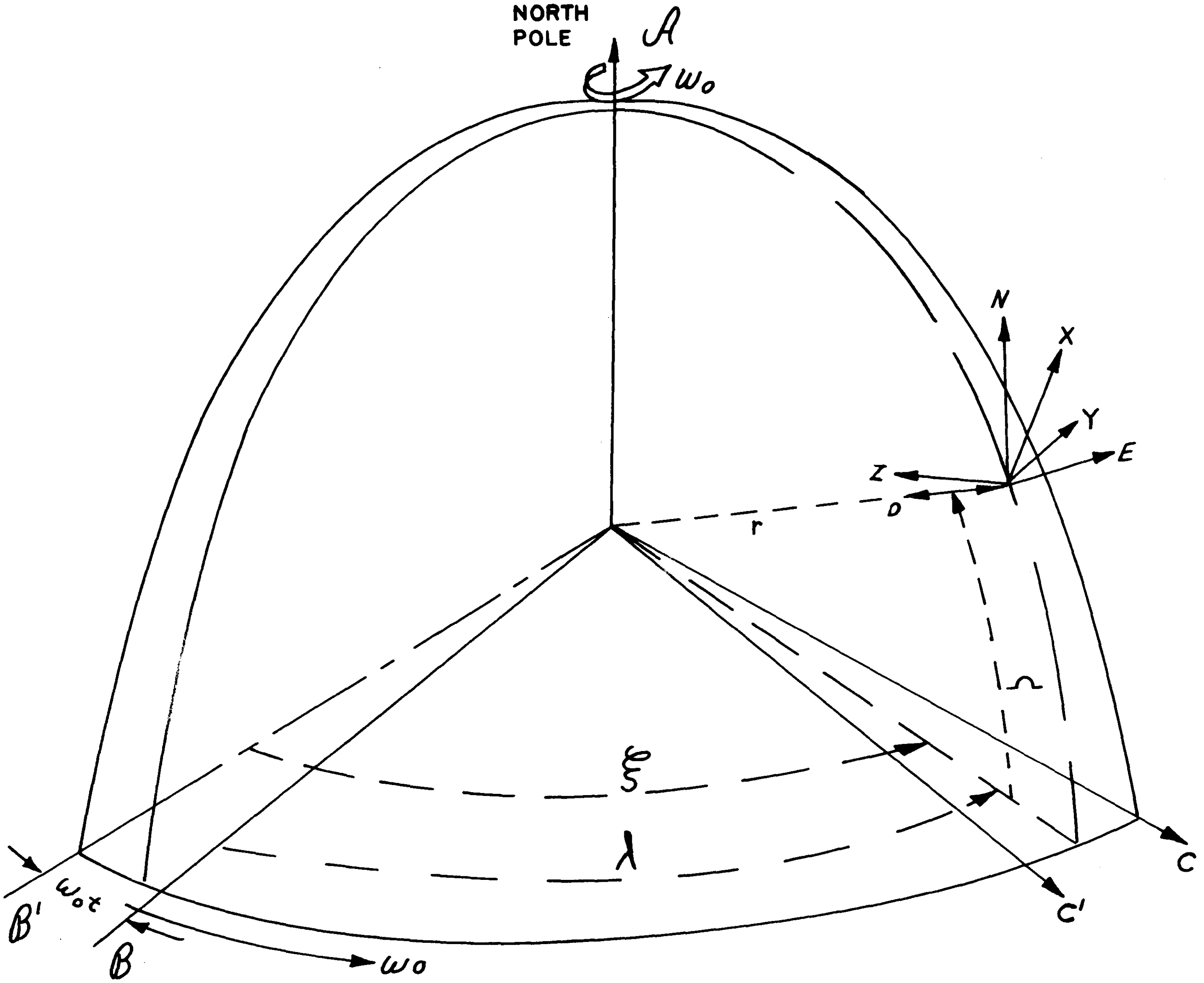


FIG. 20 BLOCK DIAGRAM OF DIGITAL PROGRAM



AB'C' : SPACE FIXED REFERENCE FRAME

ABC : EARTH FIXED REFERENCE FRAME

NED : LOCAL HORIZONTAL REFERENCE FRAME

XYZ : BODY REFERENCE FRAME

$\lambda \Omega r$: SPHERICAL COORDINATES RELATIVE TO EARTH FIXED REFERENCE FRAME

FIG.21 ORIENTATION OF SPACE FIXED, EARTH FIXED, AND BODY REFERENCE FRAMES USED IN SIX DEGREE OF FREEDOM SIMULATION

$$q_{TS} = \int_0^t (\dot{q}_{RS} + \dot{q}_{CS})\, dt$$

$$\beta_e = \beta_e(h) = \frac{1}{\rho}\frac{d\rho}{dh}$$

β_e is calculated from the stored density table by use of a second order curve fit technique to determine the derivative. These equations were calculated in the digital section because of the large numbers of complex functions required (i.e., sine, cosine, log, exponential, square roots, etc.). However they are easily computed on the 375 by use of library subroutines. Fortunately these terms were also slow-varying, trajectory dependent quantities so that the slower serial computation on the digital is acceptable. The temperature rate equations involving angular rates are calculated on the analog because of their rapidly varying characteristics.

Because of their critical accuracy requirements, the guidance calculations are also performed on the digital section. These can be separated into two parts; (1) the determination of the range-to-go, (RTG), heading angle to target, (σ_T), cross range-to-go, (CRTG), and heading error, and (2) the generation of down range and cross range errors (DR_E and CR_E) which are sent to the TRFCS and subsequently controlled to zero. The equations for range-to-go, heading angle, cross-range-to-go and heading error, which are determined from spherical trigonometric relations, are given in Appendix A. Fig. 22 illustrates the physical significance of these quantities. The down range and cross range errors used for guidance are generated as follows:

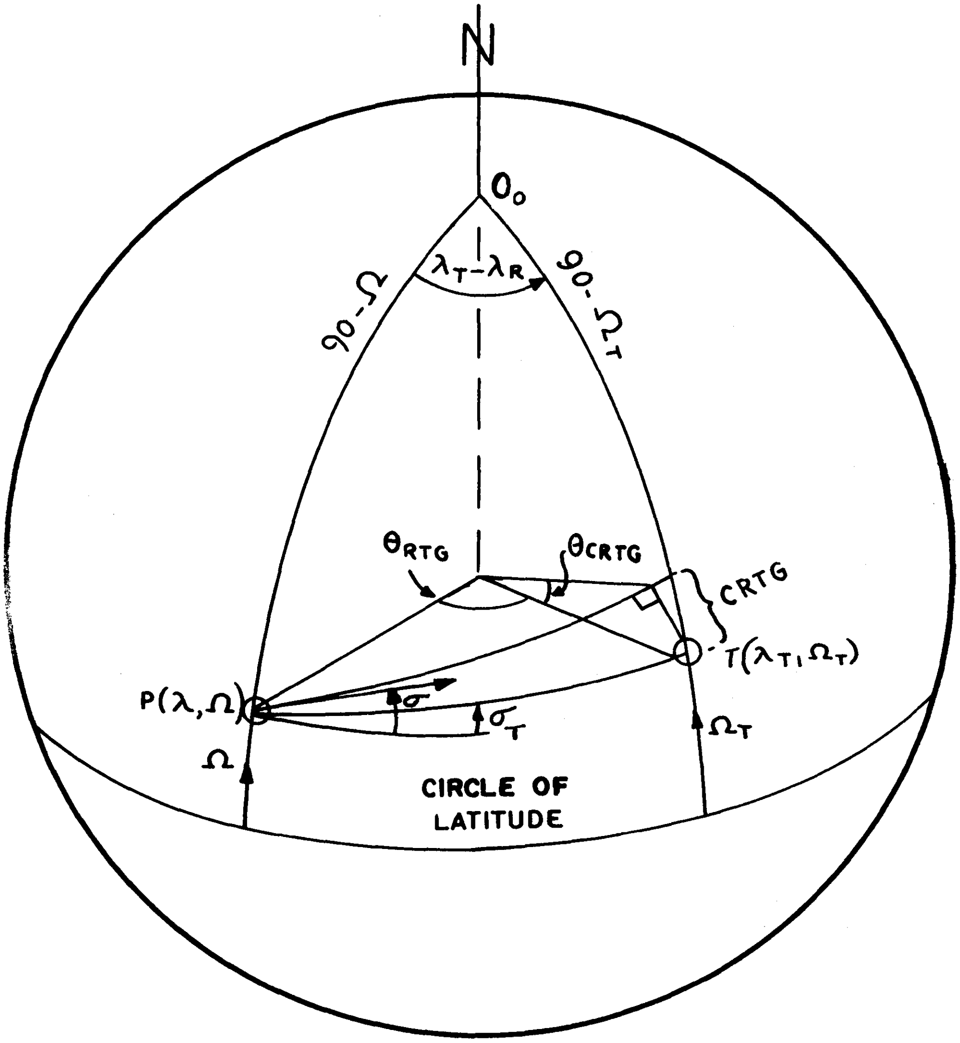
$$DR_E = RTG - RTG_d$$

$$CR_E = CRTG - CRTG_d$$

$$RTG_d = f_1(V)$$

$$CRTG_d = f_2(V)$$

f_1 (V) and f_2 (V) are the values of RTG and CRTG for a nominal re-entry. They are stored as a function of the relative velocity.



HEADING ERROR: $\sigma_T - \sigma$

FIG.22 DEFINITIONS OF λ, Ω AND RANGE ANGLES

2. General Description of Digital Program

In addition to solving the translational, temperature, and guidance equations, the digital program must accept mode control and timing commands from the DOS, scale variables as they are transferred in and out of the digital section, and finally, provide various input-output functions as described in the DOS section.

Fig. 23 shows the flow diagram of the digital program. The executive program is discussed in detail in the section that follows. The dashed lines in the diagram link together the calculations which are performed when the IC mode of operation is selected. All operations are performed except the Runge-Kutta integration shown in block 9. When the operate mode is selected the integration loop is entered and three passes around the loop are made (see description of numerical integration which follows). At the end of three passes, the positions and velocities, temperatures and guidance functions are updated and the analog-to-digital and digital-to-analog transfers are made. The executive loop is then re-entered and the cycle continued until a different mode is selected. The following section discusses this executive loop in detail

3. Digital Executive Program

The main media of communication, through which the 375 receives the mode commands and timing from the DOS 350 is the digital executive program. (See Fig. 24). This program is essentially a chain of test instructions, through which the digital section interprets the DOS 350 commands and executes them by jumping to the respective portion of the stored program.

Since some of the symbols are not identified in Fig. 24, a short explanation of the executive program follows. The first test determines whether T is greater than TF, where T is the current time (time elapsed sinc the beginning of this simulation run ·) and TF is final time indicating the end of the run. If final time is reached, results are outputed and the digital section is halted. When re-started, the digital program begins with the next decision. The next decision determines whether fast or slow time scaling is requested. If fast time scale is selected, the increment size for integration DLT (delta t), becomes 1 sec; otherwise it is 0.1 sec. The subsequent two test instructions simply switch the digital program into TIC or TYTI modes if so commanded by the mode control logic. (See Section B for a definition of these modes). The T > TT decision determines if it is time for the periodic dump of pre-selected digital parameters. The variable TT holds the time for the next dump, say 20, 30, 40 etc. seconds. TP is the print-out time interval and increments TT when the dump time is reached. If the dump switch is on and T > TT the output dump is performed, if not, the program proceeds to the next decision. The next test, keeps the entire digital program under the timing control of the master timer. The 375 cannot proceed further until the next

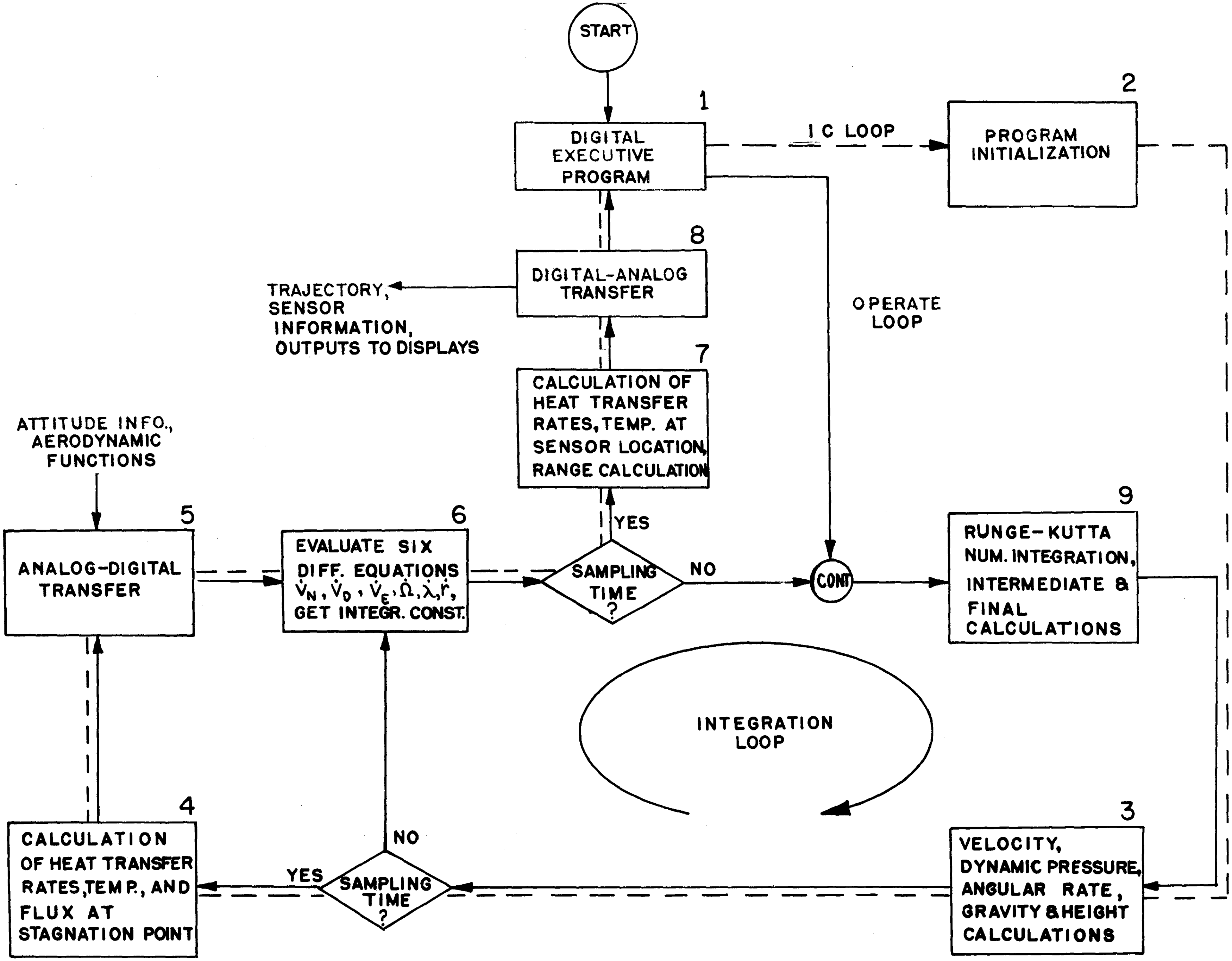


FIG.23 FLOW DIAGRAM OF DIGITAL PROGRAM

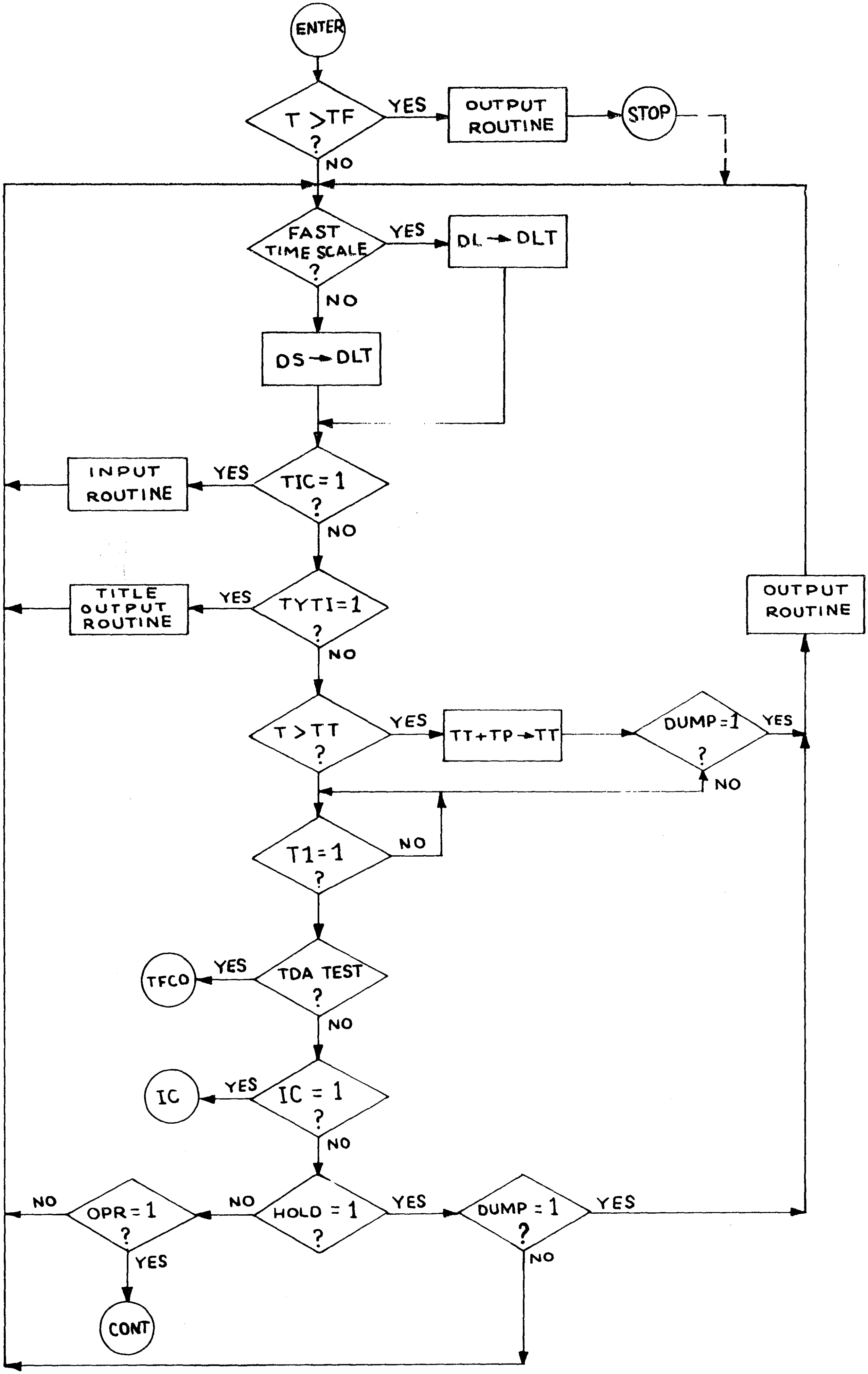


FIG.24 DIGITAL EXECUTIVE PROGRAM

T1 pulse indicates the beginning of the next computational cycle. Further testing takes place only after the T1 pulse arrives. The TDA test enables the programmer to place the digital program into a loop where only DA transfer is performed. This mode serves as a convenient test for DA conversion. (This instruction may be removed from the program after the check-out phase is over). The last few tests are self-evident and require no explanation with the exception of the seemingly superfluous second test for dump. The test for dump in hold mode makes it possible to dump any time by putting the computer in hold mode prior to the dump request, in addition to or in lieu of periodic dumps. In this manner the programmer can determine parameter values with digital accuracy at any time during the simulation.

4. Numerical Integration

Unquestionably one of the most complex and time-consuming parts of the digital program is the solution of the six simultaneous differential equations which provide the position (altitude, latitude, and longitude) and the velocity (radius rate, velocity east, velocity north) of the vehicle.

The numerical techneque selected for this calculation was the fourth order Runge-Kutta method. The basic method as applied to a single differential equations is described briefly.

Let $\frac{dy}{dx} = f(x,y)$ represent any first order equation and

$$K_1 = h\,f(x_n, y_n)$$

$$K_2 = h\,f\left(x_n + \frac{h}{2}, y_n + \frac{K_1}{2}\right)$$

$$K_3 = h\,f\left(x_n + \frac{h}{2}, y_n + \frac{K_2}{2}\right)$$

$$K_4 = h\,f(x_n + h, y_n + K_3)$$

$$\Delta y = \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4)$$

then $x_{n+1} = x_n + h$ and $y_{n+1} = y_n + \Delta y$

The increment for the second interval is computed in a similar manner by means of the same formulae

A computer flow diagram for this scheme is shown in Fig. 25. The loop on the left side of this diagram corresponds to the constant calculations (k_1, k_2 k_3, k_4),while theloop on the right hand side of the drawing updates the independent variable and starts the calculation of the next increment. When the number of increments computed (n) is equal to the number desired (nf), the computation ends. Note that x_s and y_s, the starting values for the current increment calculation, must not be destroyed during this process until the new starting values are produced, since all formulae are dependent upon them.

The preceding technique was extended to a system of six equations in an obvious manner.

While preparing the digital program and effort was made to combine the calculations due to the integration with other necessary calculations to minimize the processing time as well as the memory space requirements. This approach makes it difficult to trace the integration on the flow diagram of the digital program (Fig. 23). For example, the Runge-Kutta constants are determined in block 6 and the variables are incremented for the calculation of the next constant in block 9 (such as $V_E + \frac{K_1}{2} \rightarrow V_E$) etc., and the equations are evaluated with the incremented variables in blocks 3, 4, 6, and 7. The final calculation of the variables at is performed in block 9.

The question may be asked by the competent reader: Why was this particular scheme chosen from many others available? In order to answer this question, the selection criteria is listed below.

(1) Self-starting method.

It was desirable to choose a self-starting method to simplify programming and reduce memory requirement. (If the method is not self-starting, another method is needed to calculate the first few points of the solution. This virtually doubles the integration program).

(2) Accuracy.

The error of the fourth order Runge-Kutta method is of the order of h^5, and provides a wide enough range for h within which the accuracy of the calculation is acceptable. (During the simulation runs this reasoning proved to be a valid one). The accuracy of calculation using the second or third order method was considered to be insufficient or marginal.

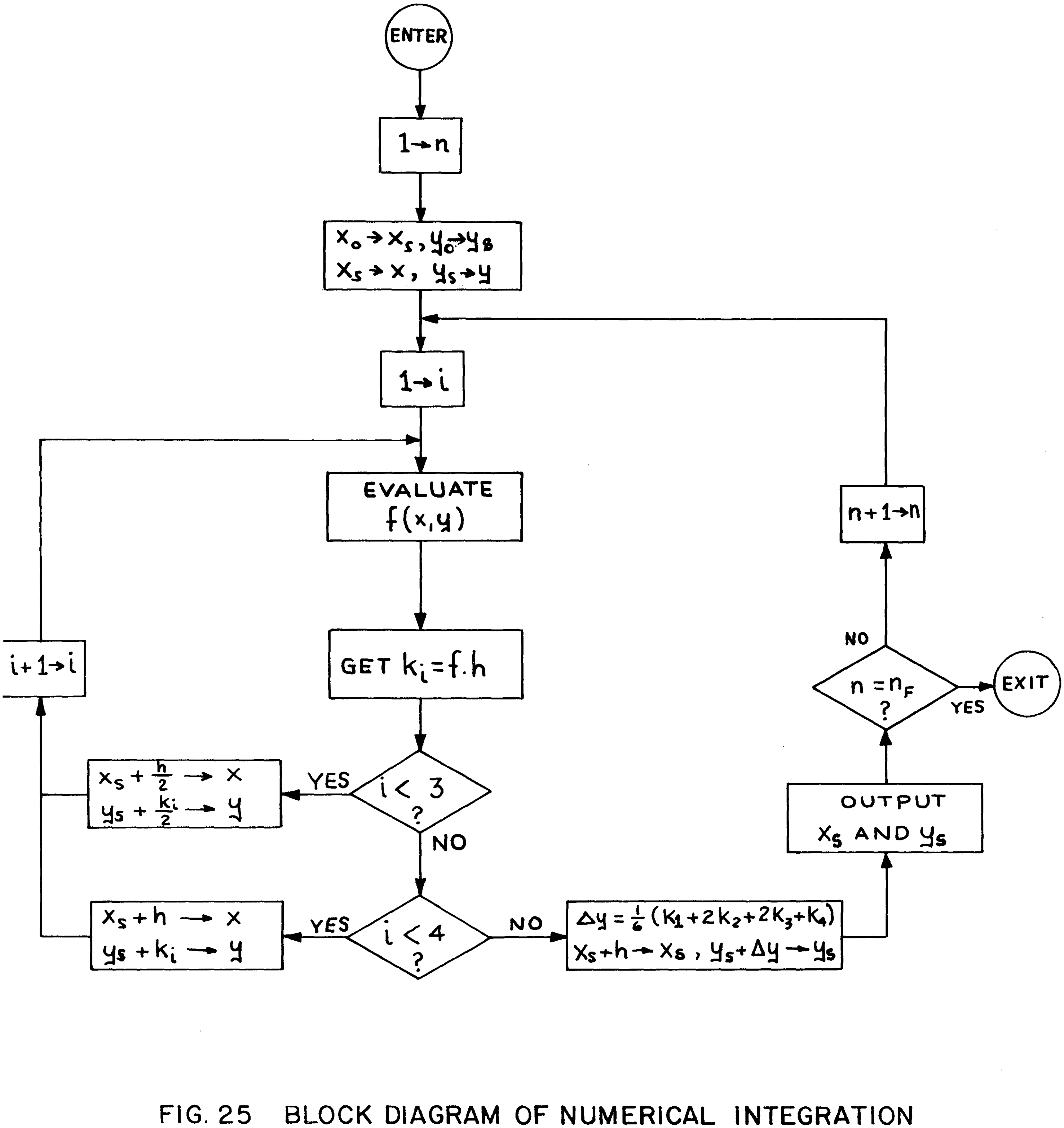


FIG. 25 BLOCK DIAGRAM OF NUMERICAL INTEGRATION
(FOURTH ORDER RUNGE-KUTTA METHOD)

(3) Large time steps in integration.

While processing time associated with this method is considerable, the accuracy is good enough to allow larger time steps. The increase in computation time using Runge-Kutta is more than off set by the increase in allowable step size.

(4) Ease of changing step size.

In the selected method the length of the step can be modified at any time in the course of the computation without additional labor. This was considered to be a substantial advantage since the simulation is frequently called upon to change time scale, thus requiring a different time step for the integration.

5. Utility Programs

In order to satisfy the various data handling needs (type in, type out, punch tape, convert binary to decimal, etc.) an excessive amount of digital programming must be done. Fortunately, all these programs are already available, tested, and clearly described in the 375's "software package". This is no small feat if one considers that the above programs, together with the numerous subroutines, normally add up to about 75% of all digital programming required. Further more, as it will be pointed out later, several programs are available to make the debugging and updating procedures efficient and fast.

D. Analog Section

The analog computer is the one link of the HYDAC 2400 system ideally suited for control system simulation by virtue of its capability for high speed, parallel computation and its input/output flexibility. Output data can be displayed in a multitude of forms such as X-Y plots, strip chart plots, oscilloscope displays, auxiliary meters, etc. Special purpose input equipment can easily be adapted for compatible operation with the analog computer.

1. General Mechanization

The block diagram of Fig. 26 delineates the mechanization of the analog program. A listing of the equations simulated in the analog section is given in Appendix A. The symbols used in the equations are defined in Appendix B.

As previously mentioned the TRFCS and rotational dynamics were programmed on the analog due to their rapidly varying characteristics. When the vehicle is in the atmosphere, the vehicle attitude is controlled by aerodynamic control surfaces. The aerodynamic control moments are calculated from the surface deflections and the moment coefficients generated on the DOS 350. Out of the atmosphere, the vehicle attitude is controlled by a reaction jet control system.

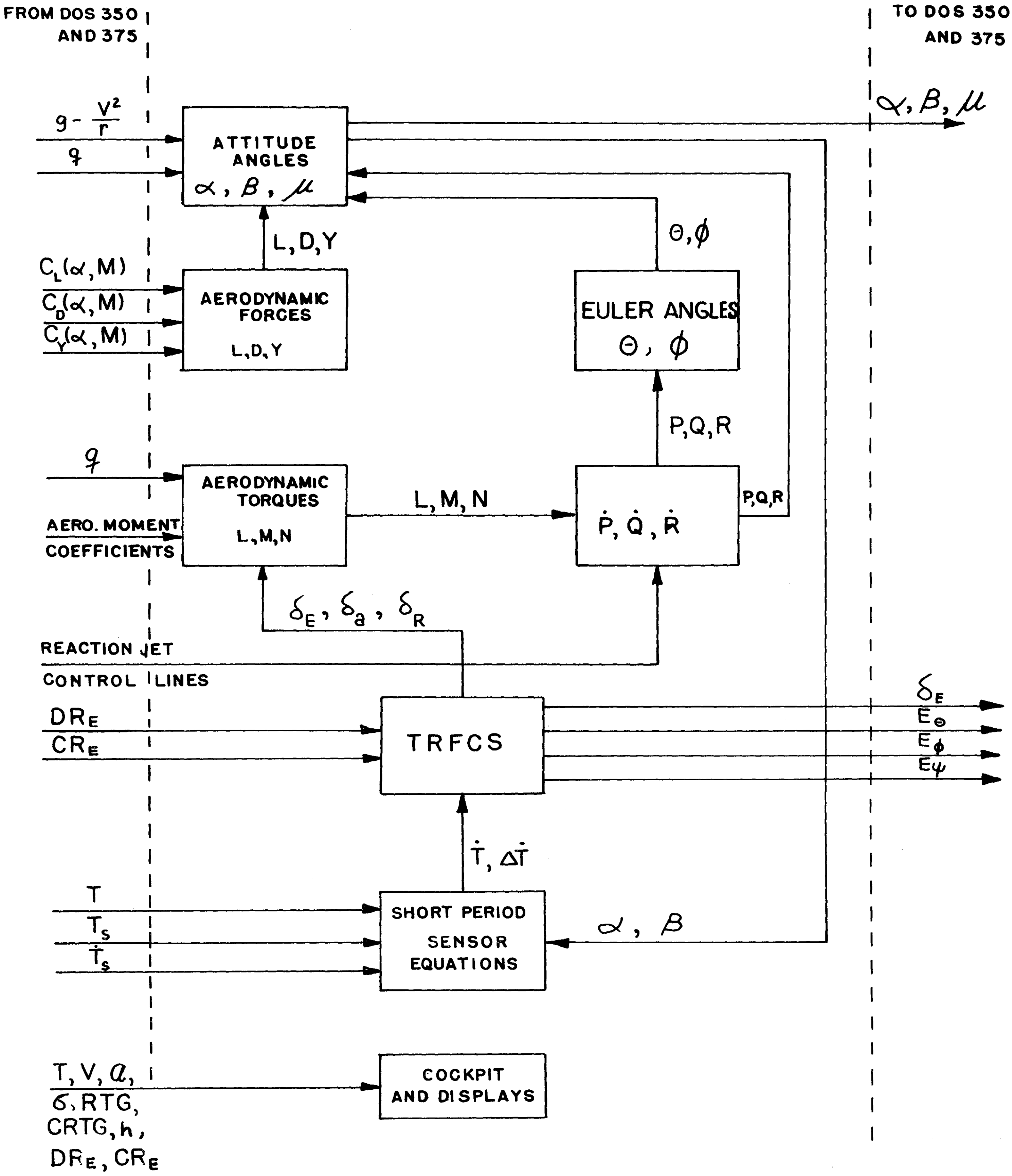


FIG.26 BLOCK DIAGRAM OF ANALOG SECTION

The jet pulse logic is generated on the DOS 350 and transferred to the analog where the moments are produced and fed into the angular acceleration equations. The angular rates, generated from the angular acceleration equations, are used to calculate the Euler angles θ and ϕ . . These angles are used to resolve gravity into the body axis for use in the $\dot{\alpha}$ and $\dot{\beta}$ equations shown below.

$$\dot{\alpha} = \left(\frac{g_r}{V} - \frac{V_I^2}{rV}\right)(\cos\theta\cos\phi\cos\alpha + \sin\theta\sin\alpha) - \frac{L}{mV} + Q - (P\cos\alpha + R\sin\alpha)\beta$$

$$\dot{\beta} = \left(\frac{g_r}{V} - \frac{V_I^2}{rV}\right)(\cos\theta\sin\phi) - R\cos\alpha + P\sin\alpha + \frac{Y}{mV} + \frac{D}{mV}\beta - \left(g_r - \frac{V_I^2}{r}\right)(\cos\theta\cos\phi\sin\alpha - \sin\theta\cos\alpha)\beta$$

These seemingly redundant force equations $\dot{\alpha}$ and $\dot{\beta}$ are computed on the analog since they are required in the short period sensor equations shown below. The aerodynamic force coefficients in the $\dot{\alpha}$ and $\dot{\beta}$ equations are generated on the DOS 350. The geometric definitions of α and β are shown in Fig. 27.

The short period sensor equations are:

$$\dot{T} = \dot{T}_S(1 - .1875\alpha^2) - .375\,\alpha\dot{\alpha}T_S$$

$$\Delta\dot{T} = .021(T_S\dot{\beta} + \dot{T}_S\beta)$$

$\dot{T}$ is the temperature rate at the nose sensor; and $\Delta\dot{T}$ is the temperature rate differential between the two wing sensors (See Fig. 1). The

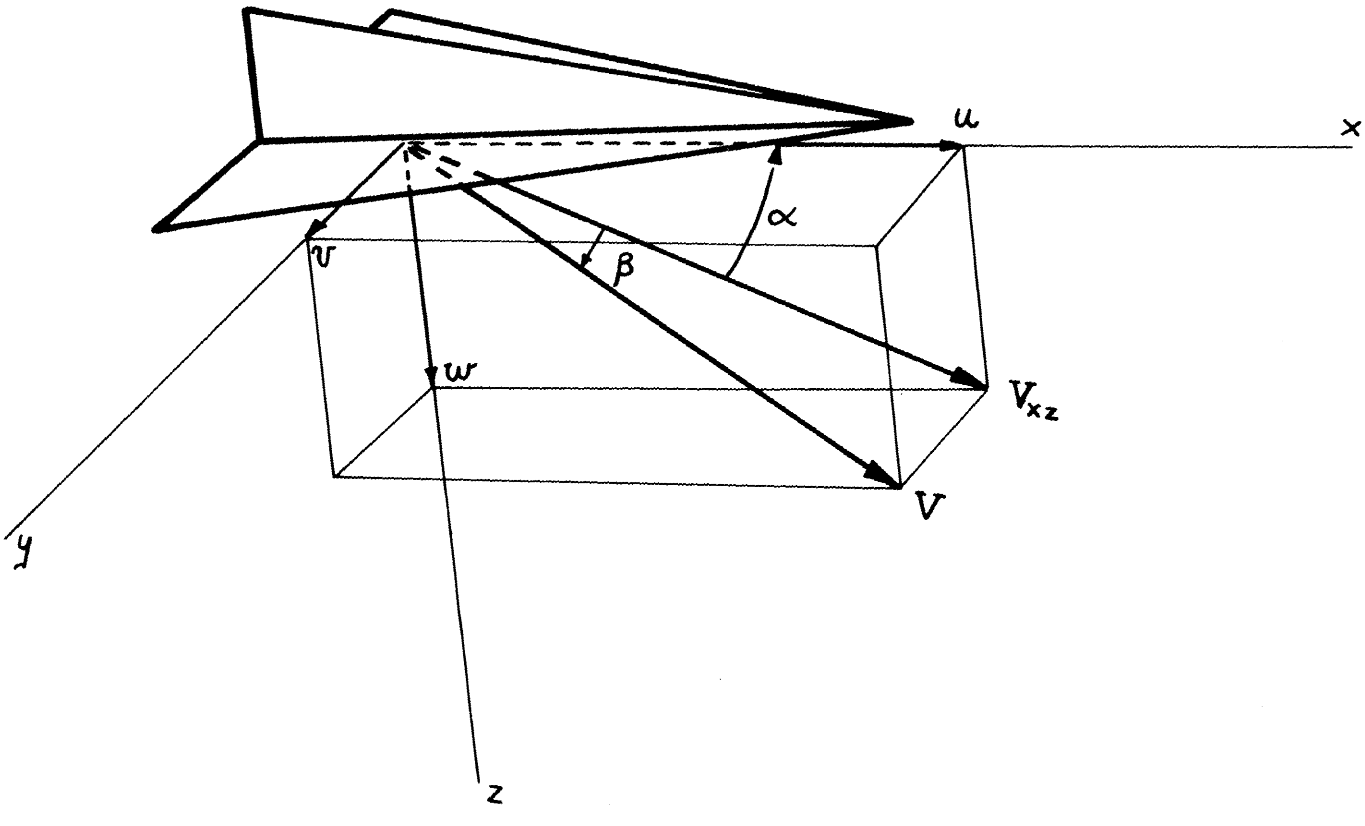


FIG.27 DEFINITION OF α AND β ANGLES

stagnation point temperature information T_s and $\dot{T}_s$ which is used in these sensor equations is calculated in the digital section. The $\dot{T}$ and $\Delta\dot{T}$ are used in the TRFCS control equations shown below:

$$\delta_E = f_1(\dot{T})$$

$$\delta_a = K_1(\mu - \mu_c) + K_2 P + K_3 \delta_R$$

$$\delta_R = f_2(\Delta\dot{T}) + K_4 \delta_a$$

$$\mu_c = f_3(\dot{T})$$

The $\dot{T}$ and $\Delta\dot{T}$ terms supply damping to the control equations alleviating the heating problems associated with an undamped trajectory. Closed loop guidance is achieved by adjusting the pitch axis controls with a compensated down range error, DR_E, and adjusting the μ_c to compensate for the cross range error, CR_E. The pilot can manually control the range by adjusting his temperature rate profile to eliminate the displayed down range and cross range errors.

2. Cockpit Simulator and Display Equipment

A cockpit simulator is utilized to evaluate the TRFCS in the manual mode, and is trunked directly to the analog computer. Computer outputs drive display meters on the TRFCS CONTROL PANEL* which monitor the following parameters:

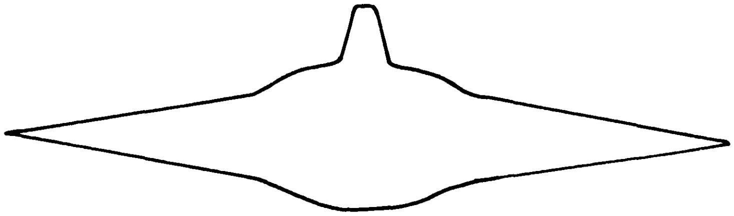
(1) temperatures
(2) temperature rate
(3) velocity
(4) acceleration
(5) pitch damper
(6) yaw damper
(7) bank angle
(8) heading angle
(9) elevator trim
(10) range
(11) range error

* Refer to Fig.'s 28 and 29 of Reference #1.

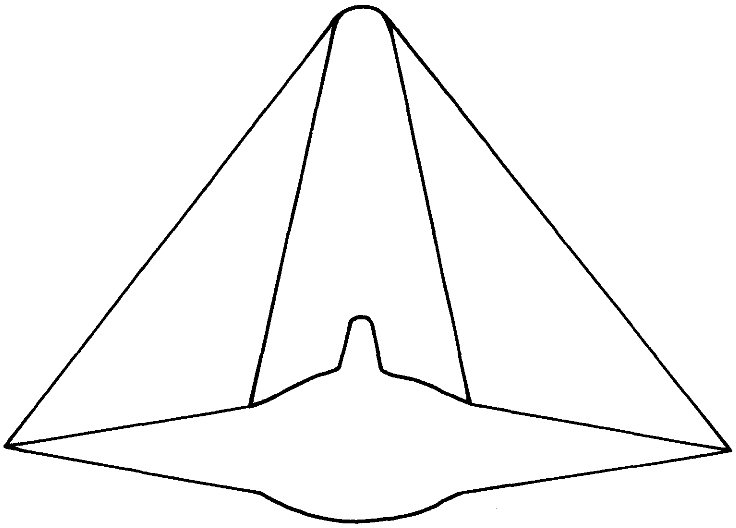
(12) cross range
(13) cross range error
(14) altitude

As seen in the analog program block diagram , many of these parameters are transferred from the digital section. Some of the parameters, though not necessary for control, indicate trajectory status and therefore maintain the pilots confidence in his control information. The above parameters and other pertinent data are recorded on strip charts and X-Y plots for permanent record of each flight.

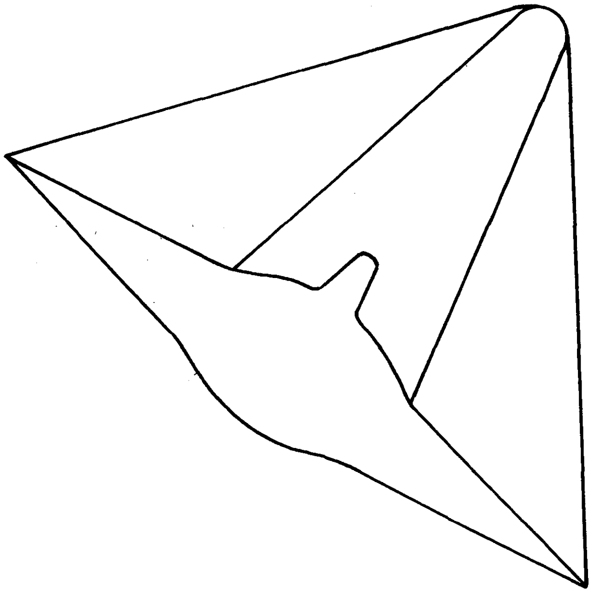
Vehicle attitudes commanded by the flight control system are displayed on a large oscilloscope. An illustration of the display at various attitudes is shown in Fig. 28.



ANGLE OF ATTACK = 0   BANK ANGLE = 0



ANGLE OF ATTACK = 45°   BANK ANGLE = 0



ANGLE OF ATTACK = 45°   BANK ANGLE = 30°

FIG.28 OSCILLOSCOPE DISPLAY OF RE-ENTRY VEHICLE

REFERENCES

1. Stalony-Dobrzanski, J., "Temperature Rate Flight Control System," Lecture given at University of California, Los Angeles, August 1963
2. Stalony-Dobrzanski, J., "Application of Temperature Rate to Manual Flight Control of Re-entry Vehicles and Energy Management, "Proceedings National Aerospace Electronics Conference, Dayton, Ohio, 1962
3. Chapman, Dean R., "An Analysis of the Corridor and Guidance Requirements for Supercircular Entry Into Planetary Atmosphere," NASA TR R-55, 1959
4. Frederickson, A. A., "Analog Computer Mechanization for Guidance Law Studies," The Boeing Company, Report #D2-8117, November 1960
5. Lees, Lester, Hartwig, F. W., and Cohen, C. B, "Use of Aerodynamic Lift During Entry Into the Earth's Atmosphere," Jet Propulsion, Vol. 29, No. 9, September 1959, p. 633
6. Wisneski, M. L. " An Analog Computer Simulation for X-20 Glide Phase Guidance Studies," The Boeing Company, Report #D2-90234, August 1962

THE COMPUTATIONAL SPECTRUM OF HYBRID APPLICATIONS

I. Introduction

Several different approaches might be considered in classifying hybrid applications. The physicist would probably like to classify them according to basic physical law, such as conservation of momentum or the second law of thermodynamics. On the other hand, mathematicians would prefer to use as argument the various classes of equations, such as transcendental, or hyperbolic partial differential. These approaches adequately classify scientific "calculations" but would prove wholly impractical for that portion of scientific computation which is being considered here -- simulation. A given problem in systems engineering contains too many physical laws, too many types of mathematics for the above approaches. Therefore, this reduces to two other arguments -- the individual physical system (or application) and the computational nature of the simulation. The former would be useful, but would bear little relation to the types of computational equipment involved. Since the practicing engineer must give prime consideration to equipment available, the economics involved in the application computational nature will be considered the most appropriate manner of classifying the spectrum of hybrid applications.

II. The Computational Spectrum

Now, consider the extent of the applications spectrum as shown in Fig. 1. Pure analog computation is on the left and pure digital on the right. Moving from left to right, the computational nature of the application will require more and more digital techniques. The spectrum can be classified into eight distinct groups, each of which are associated with certain classes of applications and equipment. These groups can be listed as:

1. High Speed Sequential/Iterative Computation
2. Data Analysis
3. Function Generation
4. Partial Differential Equations
5. Optimization
6. Digital Subsystems
7. Dynamics/Extensive Algebra
8. Dynamics/High Resolution

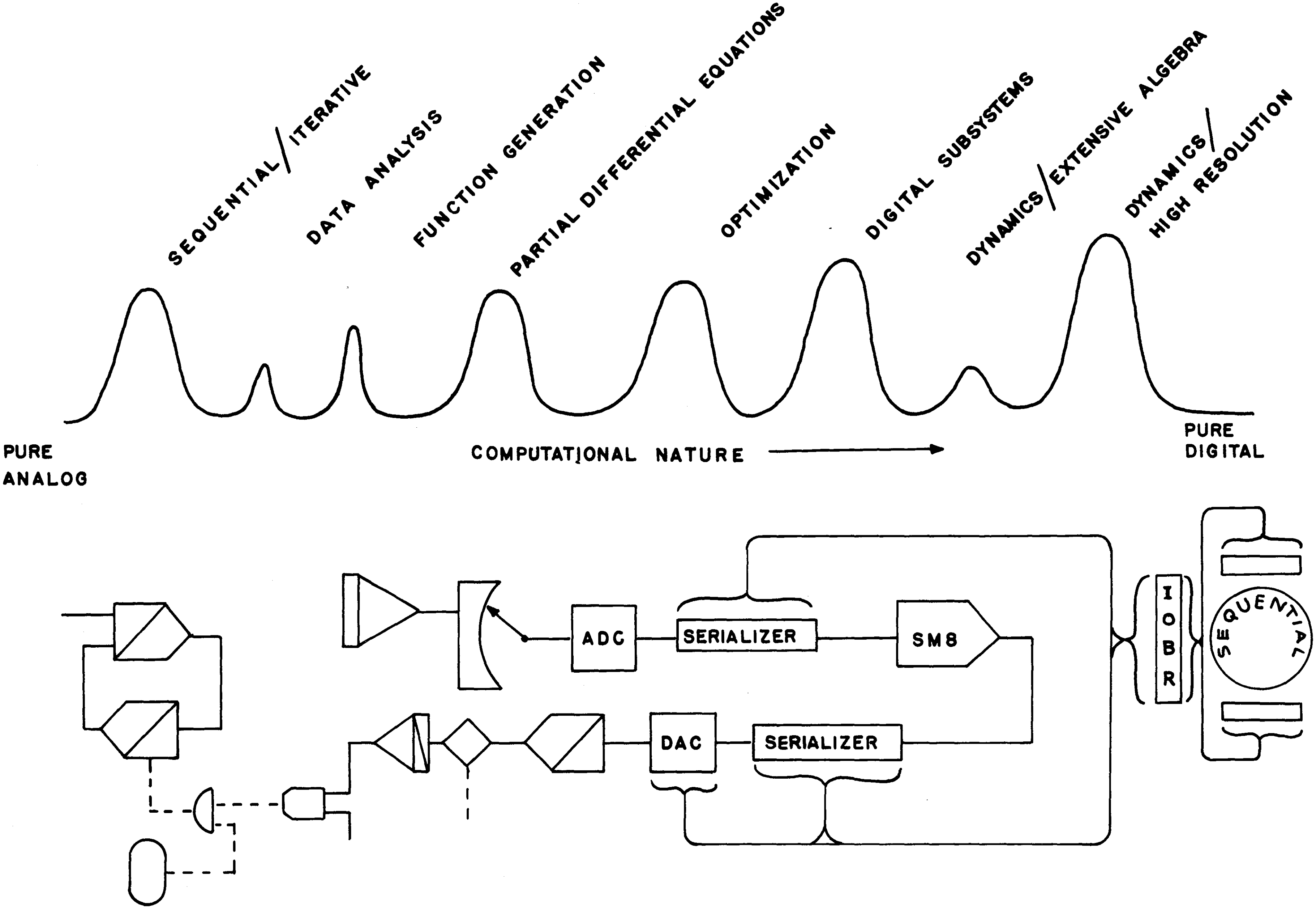


FIG. I THE APPLICATIONS SPECTRUM

Group 1 corresponds to applications involving electronic mode control, point storage, one bit communication (comparators and DA switches) and a small amount of logic, as is found in the 231R-V. As these problems grow to involve many parameters or more elaborate measurement, the extensive logic and one bit of communication of the DOS 350 becomes necessary, and the HYDAC 2000 becomes the appropriate computing system.

Groups 2 through 5 require the addition of serial memory, whole word communication, and serial arithmetic as found in the more fully expanded DOS 350. These applications form the bulk of the requirement for an analog and a parallel digital section -- the HYDAC 2000.

Finally, groups 6 through 8 require the additional capability of a serial digital section, and become the applications for the full HYDAC 2400 system. As described earlier, the DOS 350 is the control center of the system. Therefore, the serial section is relieved to a large extent and can perform, at minimum iteration times, the calculation of algebraic equations and differential equations at high resolution.

A. High Speed Sequential/Iterative Computation

The analog computer with a small amount of logic and point storage is in a position to automate many of the time consuming tasks of the past as well as open up new dimensions of system investigation. (See Fig. 2)

1. Parameter surveys, the first step in non-linear optimization, can be completely automated with the use of electronic mode control, comparators, D/A switches, track/store accumulators, counters, and gates. Nested accumulation can provide automatic scan of many parameters, the slowest changing under control of tape-set pots, while interpolated plots of variables vs. parameters, or vice versa, can be rapidly generated by use of strobing techniques.

2. Monte Carlo methods are closely akin to the preceding, where highest speed accumulation can be a series of runs with noise inputs. Here the outputs are likely to be means or variances plotted against parameters. As an example of 1 and 2, refer to Fig. 3.

This is a block diagram of a high speed, two parameter survey of a system studied in the presence of noise. The system simulation might

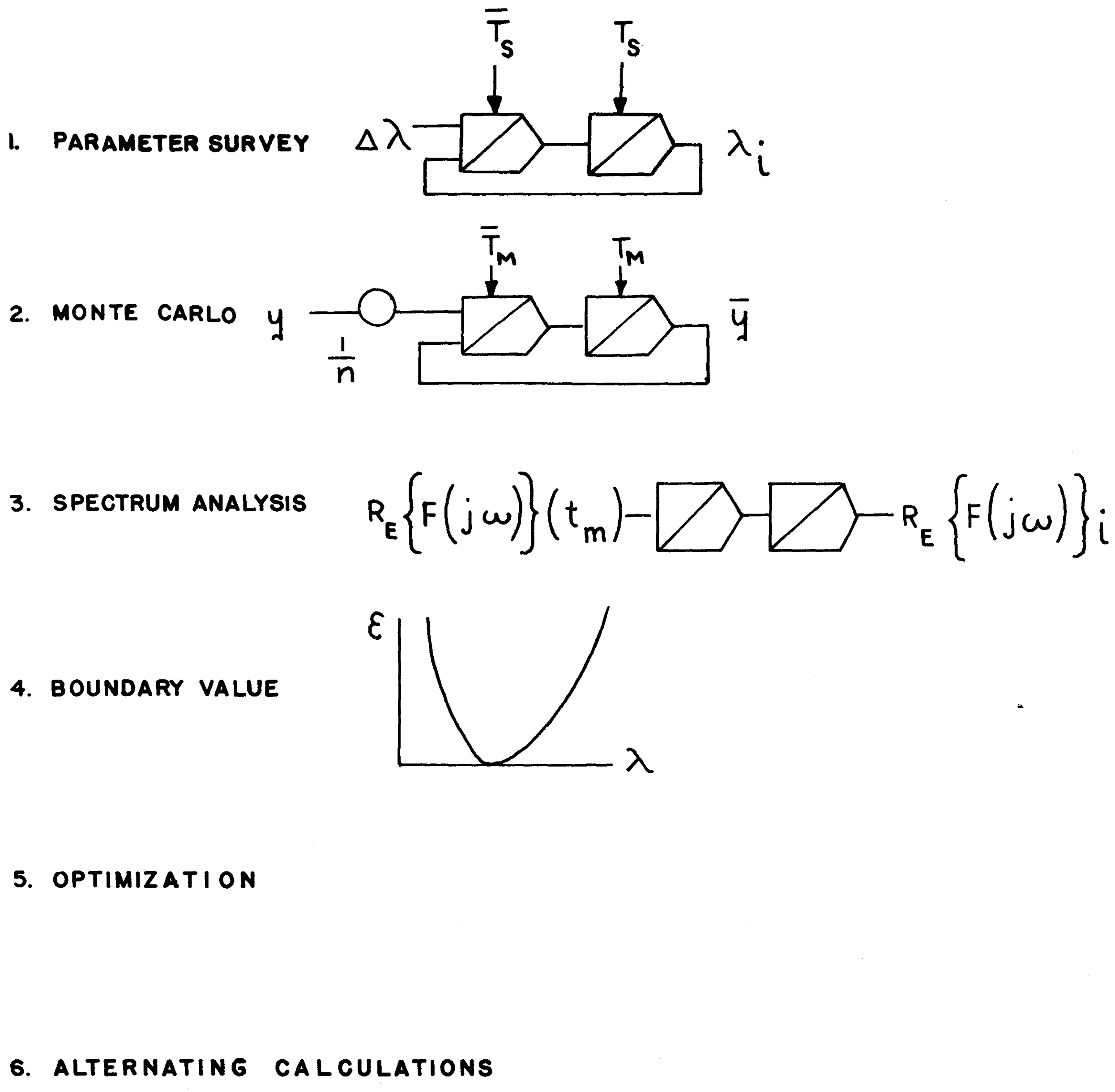


FIG.2 HIGH SPEED SEQUENTIAL/ ITERATIVE COMPUTATION

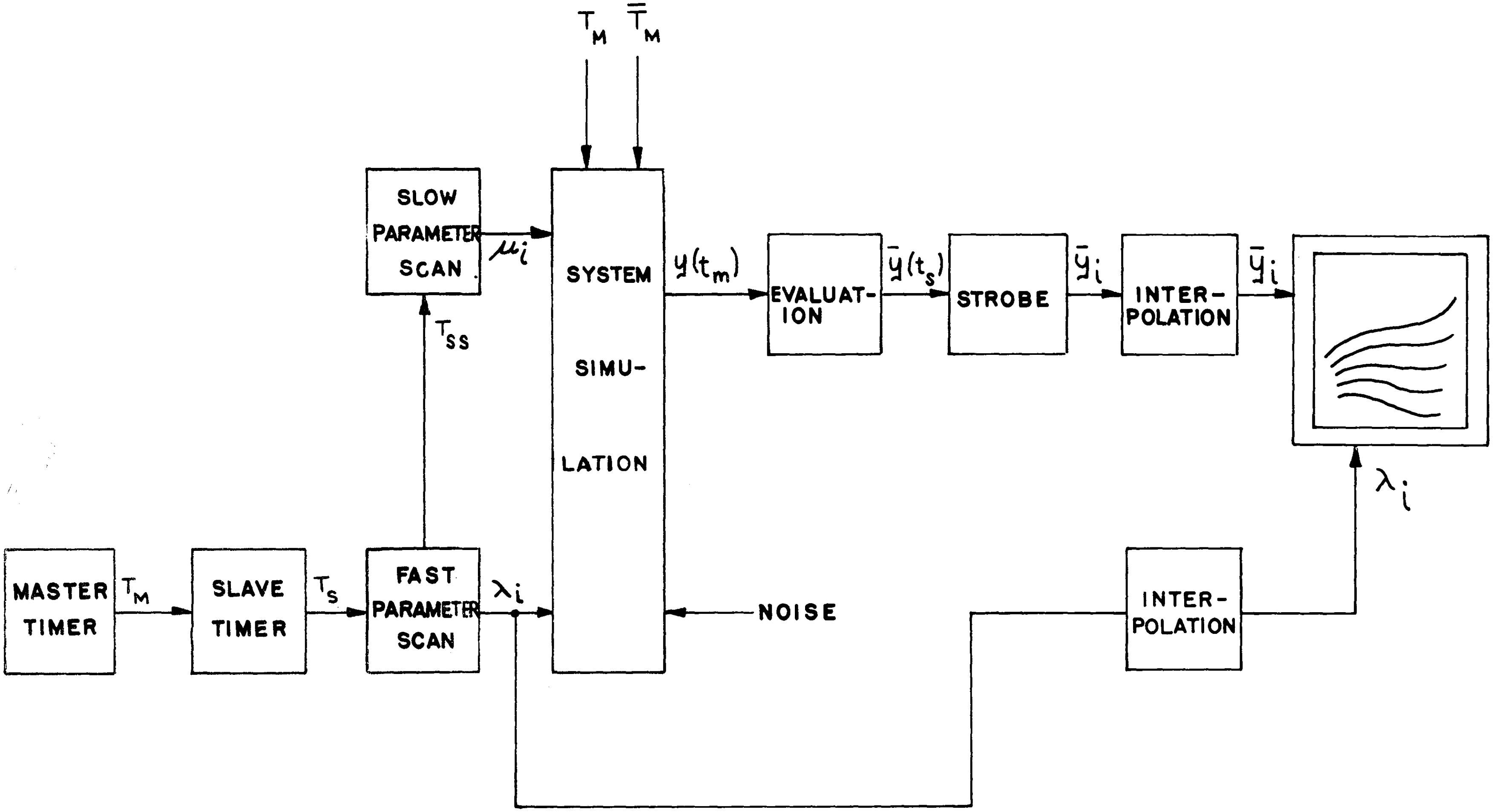


FIG.3 MONTE CARLO PARAMETER SURVEY — BLOCK DIAGRAM

be a mechanization of a homing missile, where the integrators are under control of the Master Timer signals. One run might last 20 milliseconds. An ensemble of 50 runs is then counted by the Slave Timer, whose output (T_S) provides an advancing pulse to the Fast Parameter Scan accumulator. The accumulator, or ratchet circuit, is the basic circuit behind analog counters, scanners, and evaluators, and is mechanized with a pair of track/store amplifiers and a comparator. This Fast Parameter Scan circuit provides a series of levels, λ_i, which can be forcing functions, initial conditions, or multiplier inputs in the simulation. The other output, T_{SS}, is a digital signal for advancing the Slow Scanner. Thus, the λ_i, for example, can be 10 values of a loop gain, changing value every second, and the μ_i 10 values of dead zone changing value every 10 seconds. Evaluation in this example might be measurement of the mean miss-distance over the 50 Monte Carlo runs, denoted as $\bar{y}(t_S)$. This fast moving information must be strobed by a track/store unit followed by a store/track unit. The output of the latter is a staircase, each level corresponding to the results of a given pair of the parameters λ_i μ_i. Finally, an integrator and two amplifier circuit provides a straight line interpolation between levels of μ_i, when the abscissa, say λ_i is similarly treated. The result is a family of curves, each curve representing response as a function of the parameter λ_i for one value μ_i, and plotted out in about ten seconds.

3. Spectrum Analysis is also a related topic, where now frequency is scanned with Nyquist or Describing Function plots as a result.

4. Boundary Value problems offer a prime application for iterative techniques. Often, as with the tubular reactor, the relevant partial differential equation can be written as an ordinary differential equation with a final value constraint. In this example, the initial derivative is guessed and the final value measured. The steepest descents algorithm operates on the resulting error and points the way to a better guess. Results of this series of iterations can become inputs to a real time control problem.

5. Optimization over one or two parameters can be handled with this type of equipment, as can the related curve fitting problem. Errors are completed as the difference between output and a criteria function, or a least squares formulation, and operated on by the steepest descents algorithm. Manual and semi-automatic iterations are appropriate if there are more than two degrees of freedom. Here, point storage provides a visual display of the error, which is to be minimized by successive manual adjustments of parameters.

6. Alternating Computation affords a convenient technique in certain cases, where the results of one computation will provide initial conditions for the next computation, and vice versa.

B. Data Analysis

The intent of this category is not to compete with well developed digital programs that now exist. Rather, it is the "quick look" aspects that are of interest, as so much of magnetic tape data analysis involves culling the good data from the bad. Also, there are a lot of simple tasks around that are wasteful of general purpose digital computer time. This category breaks into two areas: Simple tasks performed efficiently, and difficult tasks performed approximately, as indicated on Fig. 4.

1. Logical Operations - defined as simple tasks of counting, measuring, formatting. Typical operations are:

a. Counting number of times a magnetic tape signal crosses a threshold, or exists within given ranges, versus precise time marks. Comparators, flip-flops, gates and counters are the major tools.

b. Peak reading within given intervals requires a small amount of serial memory, and a printer or punched paper tape in addition to A/D - D/A conversion.

c. Formatting for IBM compatible tape, depending on rate requirements, can require a fair amount of logic, along with short, medium, and long delay lines for quick access bulk intermediate storage.

2. Continuous Data Analysis - defined as analog techniques for obtaining statistical parameters and functions. The Exponentially-Mapped-Past approximation, and high speed control of integrators are the major techniques. Electronic mode control and logic are required for integration over precise intervals and control of magnetic tape. Possible parameters and functions are:

a. Statistical parameters such as mean, variance, standard deviation, correlation coefficient.

b. Statistical functions such as power spectral density, Fourier analysis, auto- and cross-correlation, LaGuerre function analysis.

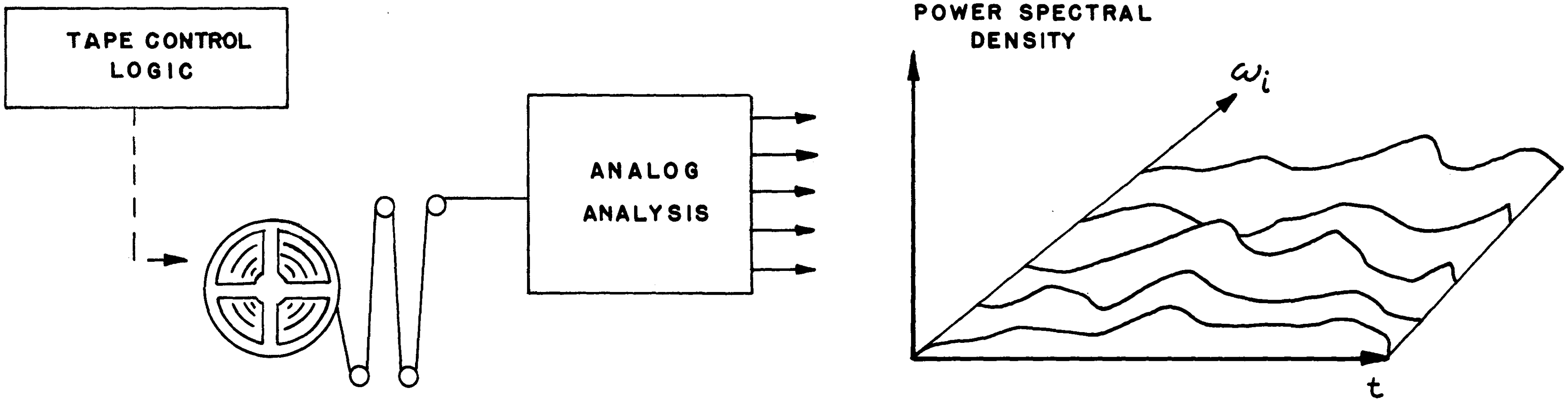


FIG.4 CONTINUOUS MEASURE OF POWER SPECTRAL DENSITY

c. Linear and non-linear regression.

3. Communication control can vary from simple latching of comparators, through tape command for an A/D or D/A conversion, to free running A/D - D/A conversion.

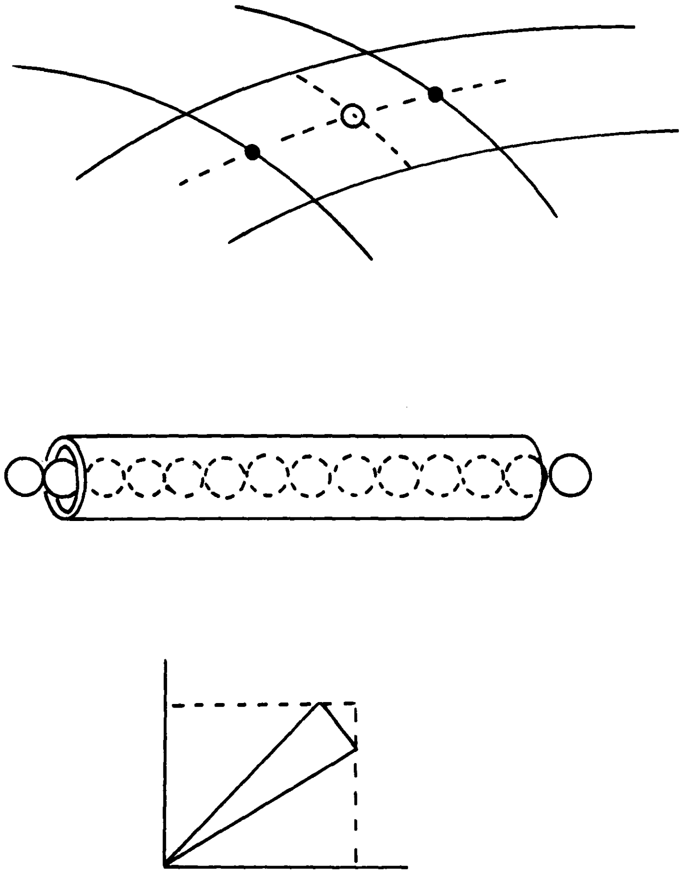
C. Function Generation

In the past the analog computer has adequately handled most problems in function generation, through tapped potentiometer, diode, and integration techniques. But increasingly, simulations are requiring more extensive sets of functions with one, two and three variables, increased fidelity of transport delay, and more stringent resolution requirements, requiring digital techniques. Fig. 5 indicates the 3 basic categories.

1. Empirical Functions, as required in two and three dimensional terrain avoidance, aerodynamic vehicles, etc. Serial memory, logic, serial arithmetic, and conversion provide the most efficient approach, particularly when the physical system includes on-board radar. When a serial digital section is required for reasons such as algebra or resolution, function generation can be done in this unit when it is consistent with appropriate iteration times. It is not always appropriately handled in the serial digital section for the following reason: When the serial section becomes involved in generating functions for the short period portions of the problem, total digital computation for one iteration can become excessive. Here it is often appropriate to perform at least the short period function generation on the parallel digital section. Conversion control in its simplest form would normally permit a planned sequence of data transfer every iteration cycle. This sequence would have to communicate with both core and serial memory in the case of mixed function generation, thus versatility in the control circuits can be quite important. The reader is referred to Chapter 3 for an example of function generation using the parallel digital section.

2. Transport Delay is conveniently and appropriately handled by serial memory, using the "marbles in a pipe" analogy. Time delay-frequency products of 30-60, and variable transport delay, are conveniently available. Conversion control would give a precession rate proportional to an analog variable, i.e., flow.

3. Cordic Logic can provide very fast generation of trigonometric functions at a very fast rate using logic, serial memory, A/D-D/A conversion, and serial arithmetic. One forward or reverse resolution may be performed every 500 microseconds, 2 resolvers may be implemented at a 1 kc word rate, etc. Fig. 6 gives a diagram illustrating the basic definition

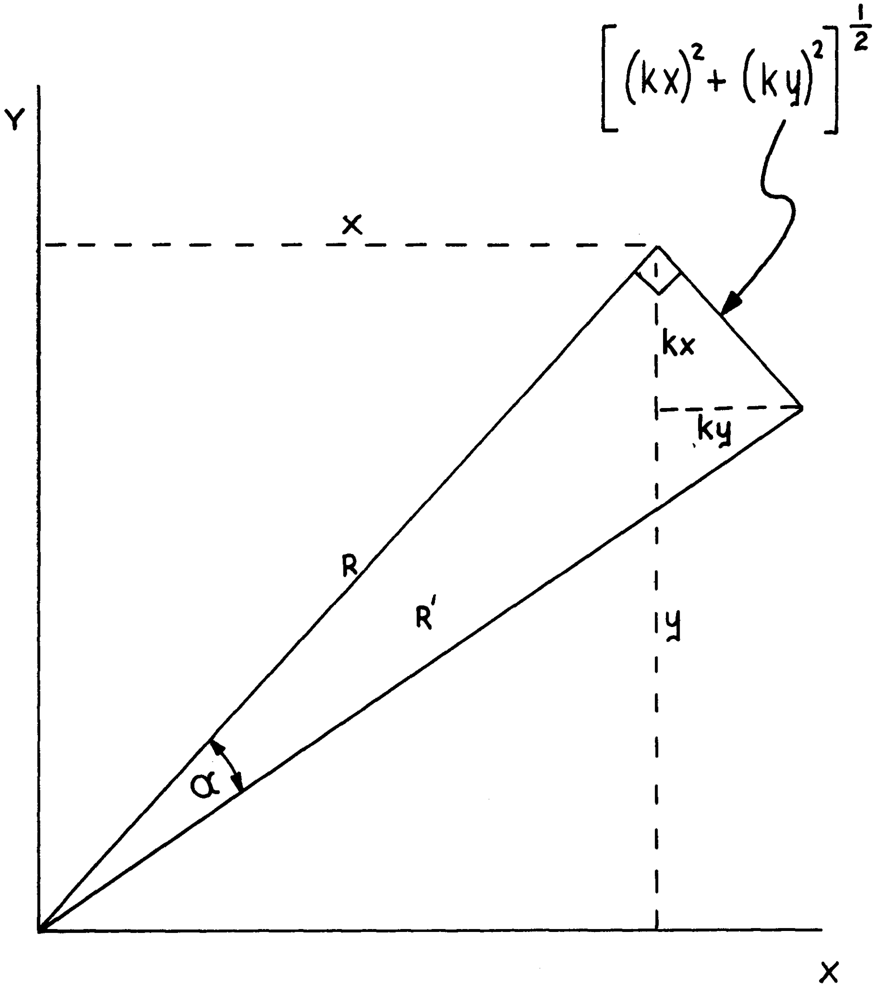


1. EMPIRICAL FUNCTIONS

2. TRANSPORT DELAY

3. CORDIC LOGIC

FIG.5 FUNCTION GENERATION



$$x' = x + ky \qquad 1.$$
$$Y' = y - ky$$

$$\tan\alpha = k = \frac{\left[(kx)^2+(ky^2)^2\right]^{\frac{1}{2}}}{\left[(x^2)+(y)^2\right]^{\frac{1}{2}}} \qquad 2.$$

$$R' = R\left(1+k^2\right)^{\frac{1}{2}} \qquad 3.$$

FIG.6 DIAGRAM OF VECTOR ROTATION SCHEME USED BY CORDIC LOGIC that is used; equation 1. Equations 2 and 3 follow from 1, and indicate the relationships used in the technique.

a. Cartesian-To-Polar Conversion

Sampled values of x and y are loaded into their respective registers, and k selected so that y is reduced. The number y can now be reduced to within one bit of zero in 16 steps, or 128 microseconds. This leaves a number proportional to R in the X register. The angle is obtained by accumulating values of $\pm\alpha_i$ from delay line storage, determined by successive values of k.

b. Polar-To-Cartesian Conversion

The vector is rotated in successive steps by choosing the sign of k such that the original angle is reduced to zero in sixteen steps. The computation involved is essentially the inverse of the above case.

D. Partial Differential Equations

The analog all-parallel approach and the digital all-serial approach both show disadvantages in economical solution of PDE's. The all parallel approach requires too much equipment for solution for more than 10-20 nodes. The approach generally used on the digital computer, the implicit solution, requires solution of a large set of simultaneous algebraic equations for every time step, and is thus expensive. Various serial-parallel approaches show promise in providing accurate fine-mesh solutions in reasonably short times.

1. Serial solution reverses the order of the all-parallel technique. Time is discretized and the space variable made continuous, allowing simple iteration to meet a final boundary condition. (See Fig. 7). A function must be stored for the previous time step, implying the use of electronic mode control, logic A/D-D/A conversion, and serial memory. The tradeoff now is time against mesh size. Conversion control must allow full rate sampling synchronized with the operate period.

2. Hybrid Implicit Solution provides an "analog subroutine" for the lengthy matrix inversion required with the implicit numerical solution. The high speed arithmetic unit and core memory provides numerical integration. Conversion control can provide a normal A/D-D/A sequence every iteration cycle.

$$\boxed{K\frac{\partial^2\Theta}{\partial x^2} = \rho c\frac{\partial\Theta}{\partial t}} \quad \Longrightarrow \quad \boxed{\frac{d^2\Theta^n}{dx^2} = \frac{\rho c}{K\Delta t}\left(\Theta^n - \Theta^{n-1}\right)}$$

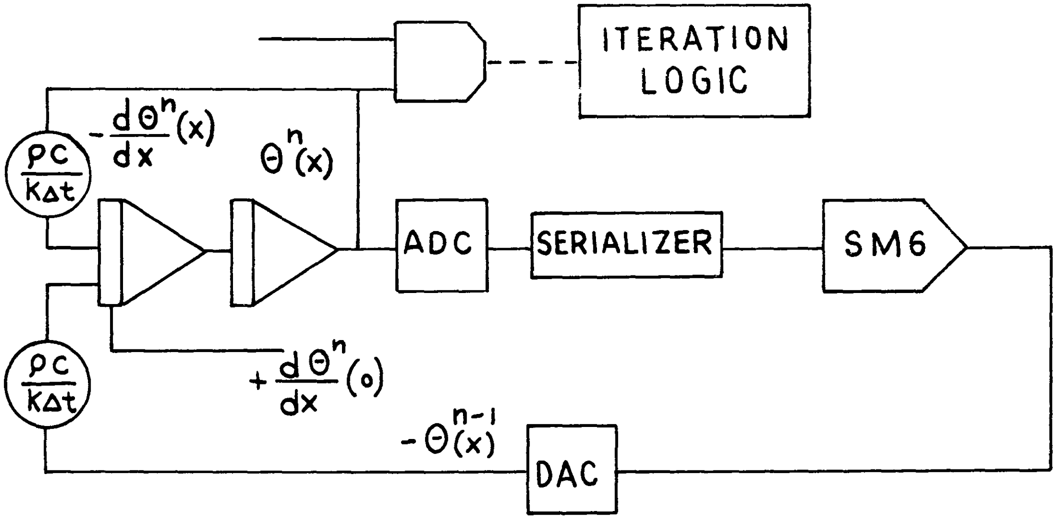


FIG.7 SERIAL SOLUTION OF DIFFUSION EQUATION

E. Optimization

This topic may be considered to cover two allied areas - model development and adaptive control. The mathematical techniques are similar, making heavy use of steepest descents.

1. Multiparameter Optimization becomes an extensive problem in logic and control. In the general case of non-linear systems, coarse parameter surveys can first be made to assure an absolute and not a relative extremum. Fig. 8 is a diagram of the steepest descent path and the quantized steepest descent path for a two parameter space. Mechanization of a quantized steepest descents formulation, assuming an appropriate criteria function, can be accomplished with electronic mode control, logic, comparators, and D/A switches. This can yield the speeds required to make optimization over many parameters a feasible matter, and a program versatile enough to readily adapt to any simulation. See Fig. 9 for a typical flow chart for an optimization program.

2. Function optimization requires the addition of A/D-D/A conversion and serial memory to iterate on functions until certain criteria are met. This is similar to the solution of integral equations, where a kernel function is guessed and integrated iteratively until satisfying the equation.

3. Model development is becoming an increasingly important area of activity. Human factors, biomedical systems, hydrologic systems, econometric models - all are characterized by the need for synthesizing known output data into the appropriate mathematical models. Thus the known output becomes the criteria function, leading to the proper parameter set by use of the steepest descents technique.

4. Adaptive Control again uses similar techniques, with the reference model as a method of generating criteria functions. This implies a continuous series of optimizations as external conditions change.

F. Digital Subsystems

One of the most obvious applications of hybrid computing is that of including digital devices as part of a dynamic model. Sometimes

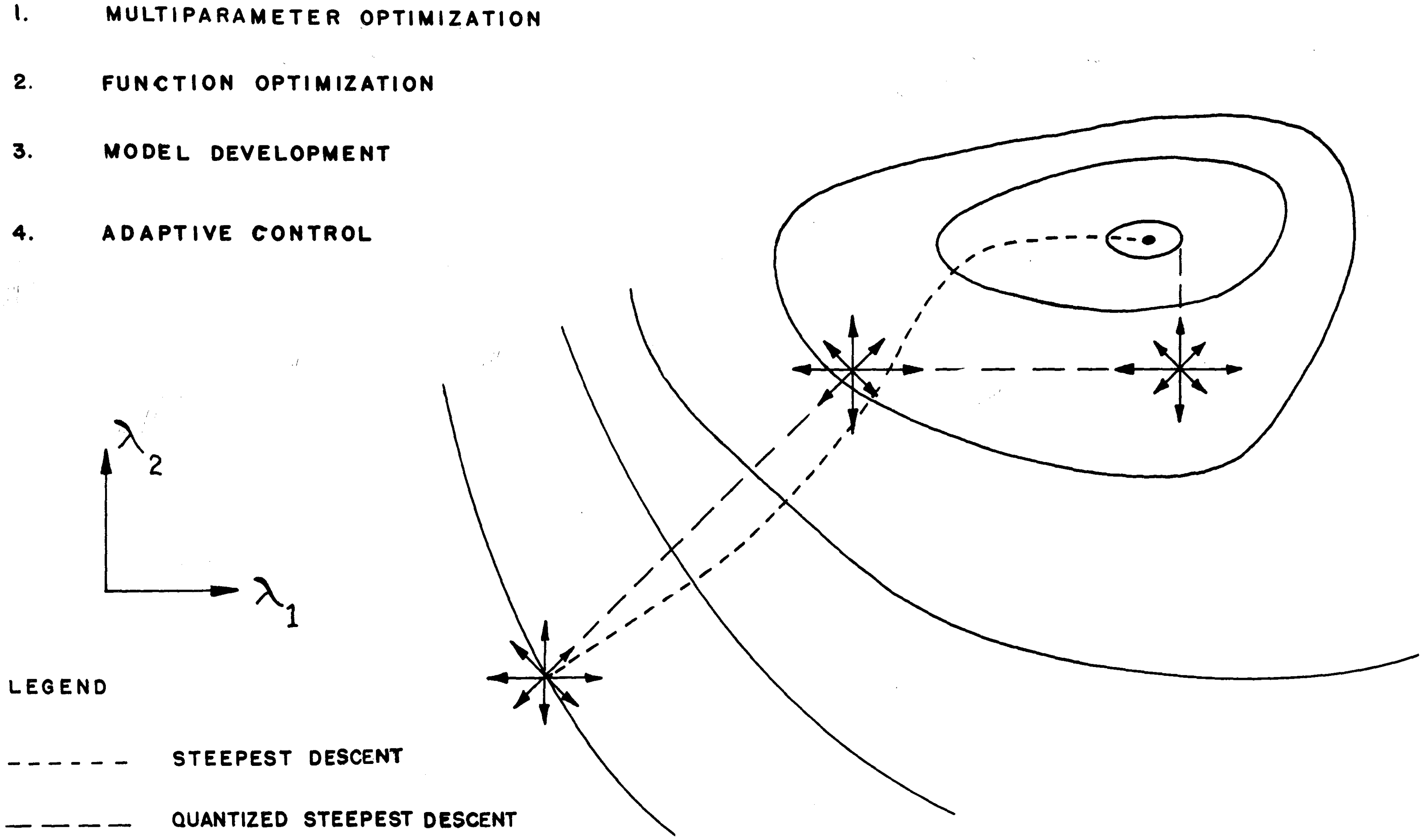


FIG.8 OPTIMIZATION METHODS

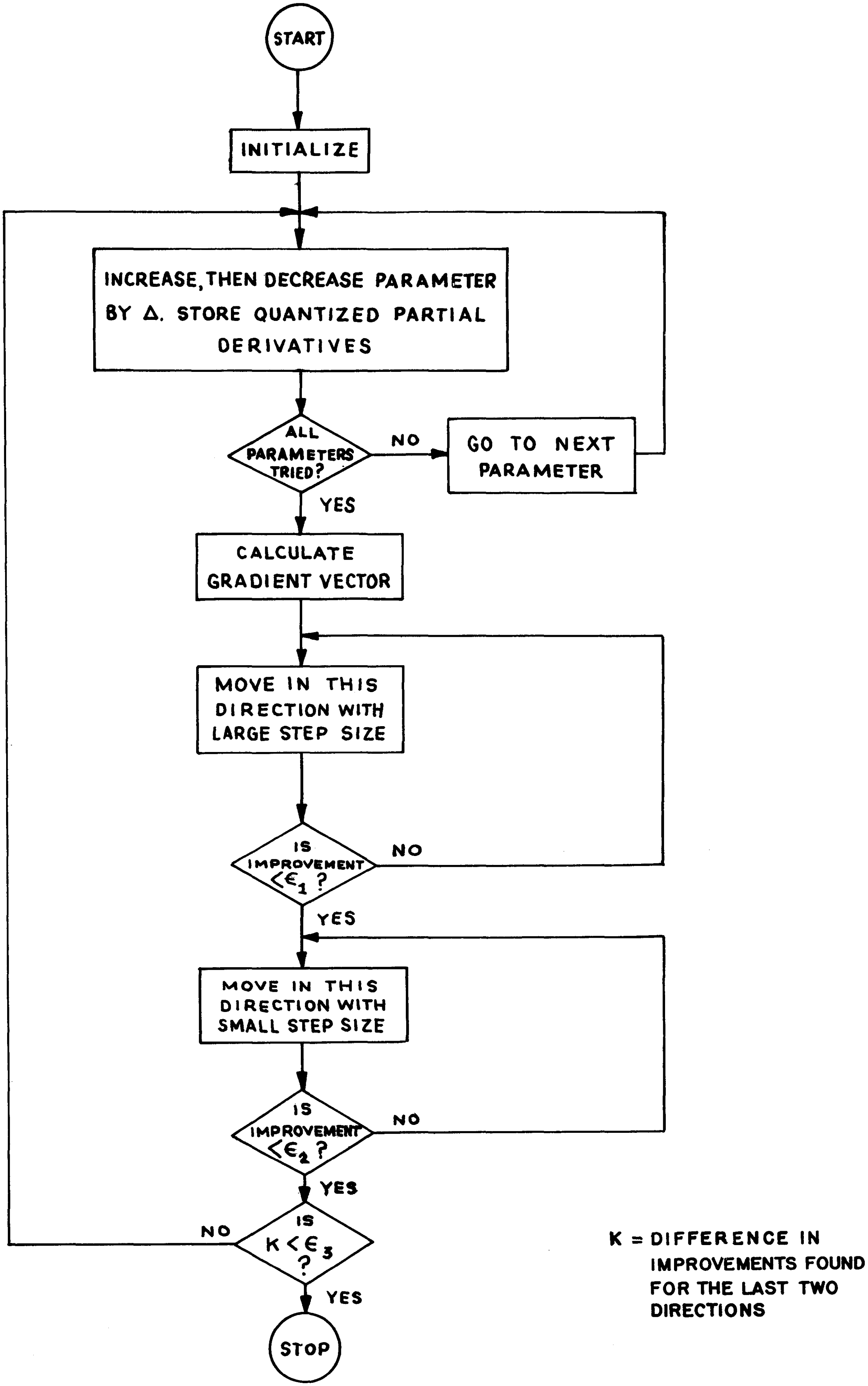


FIG. 9 FLOW CHART FOR AN OPTIMIZATION PROGRAM

thought of as sampled data control systems, they run the gamut from a small digital controller in a process unit operation to a completely integrated avionic system where all phases of operation, from communication and navigation to fire control and diagnostics, are under central control. Fig. 10 shows in block diagram form how such a simulation might be divided among the three basic types of equipment. Here the high speed arithmetic section makes its first appearance as a required element. At the same time, much care must be taken not to overload it with tasks more efficiently performed elsewhere. Therefore, radar simulation, function generation, and special purpose digital devices such as reaction jet control systems, are simulated in the parallel digital section, leaving the serial digital section free to simulate the central computer. Conversion control must be quite versatile to provide linkage to serial memory as well as to core memory, and put no restrictions on groupings or rates of transfer. In general, certain channels must be sampled at maximum rate while others are sampled at various reduced rates. A significant amount of one bit communication is also necessary, such as comparators, D/A switches, sense lines, and output control lines.

G. Dynamics/Extensive Algebra

At times the major reason for choosing a hybrid approach lies in the impracticality of handling large sets of algebraic equations in dynamic systems. A case in point is the study of transient radiation effects in electronic circuits. Devices containing more than about two transistors have too many algebraic loops for practical all-analog simulation, yet the all-digital simulation is expensive. Retaining integrations on the analog section, algebra and control in the digital section, leads to economic simulation of very complex situations. Fig. 11 illustrates the division of computing tasks. Conversion control again should be versatile enough to handle certain channels at higher sampling rates than others.

H. Dynamics/High Resolution

This application may well be the classical one for hybrid computation. Simulation of space vehicles places such a contrasting demand on simulation equipment for speed and resolution that hybrid approaches are mandatory for economic solution. The concept here is to allow the serial digital section to provide the environment, i. e., trajectory and dynamic pressure, as well as guidance, diagnostics, mission data, etc. The vehicle

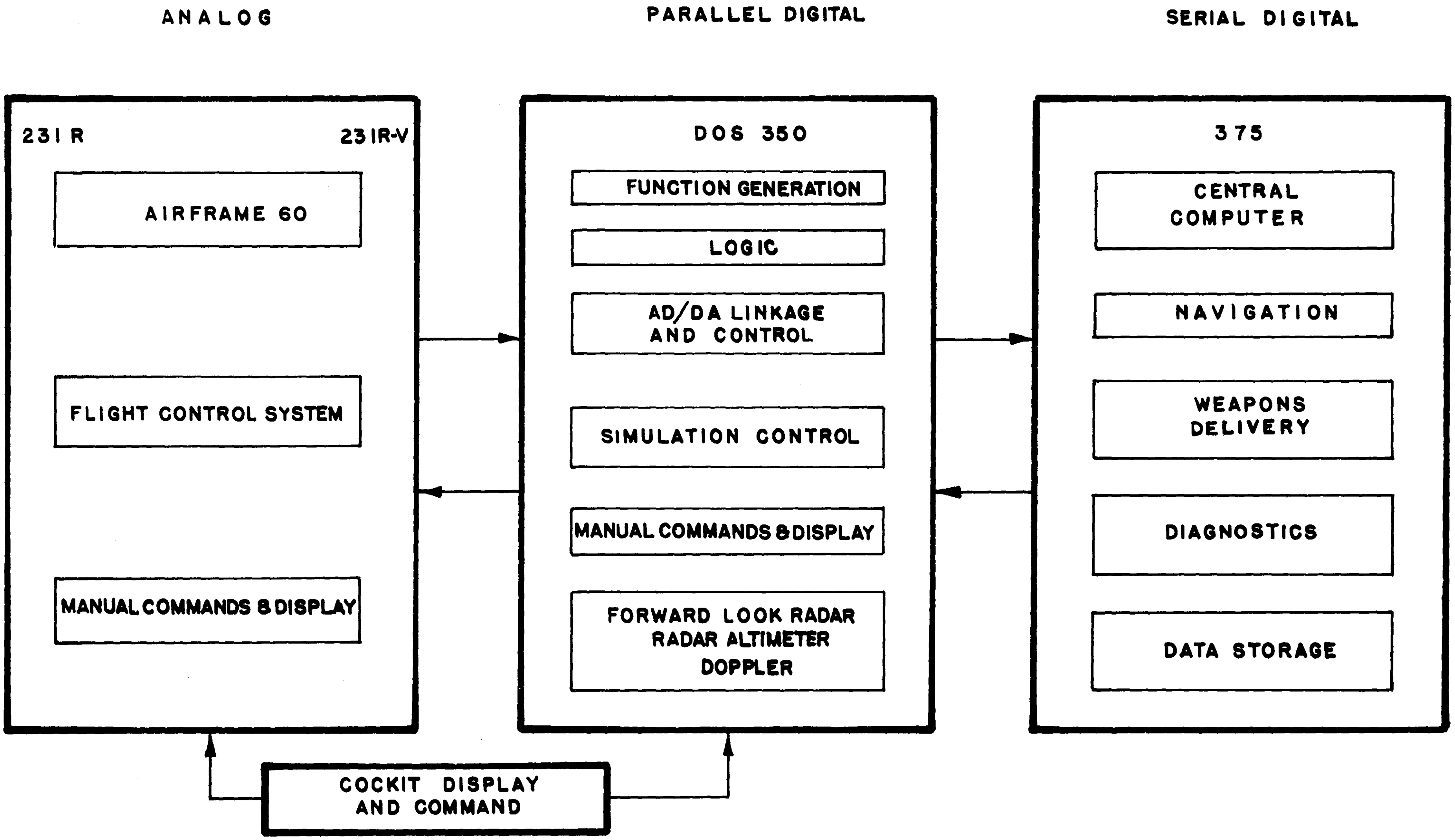


FIG.10 .BLOCK DIAGRAM OF A TYPICAL AVIONIC SYSTEM

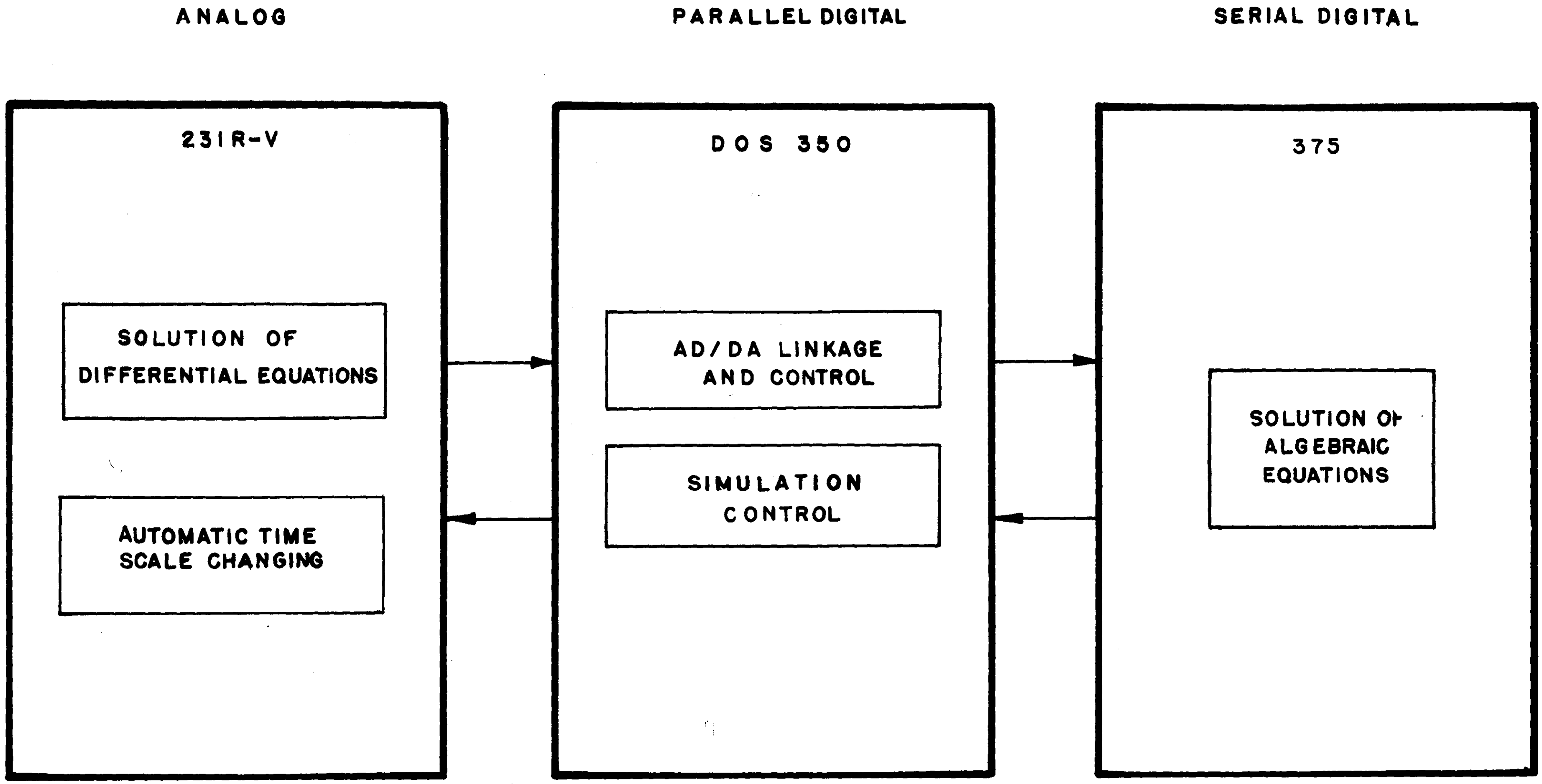


FIG. 11 BLOCK DIAGRAM FOR TRANSIENT RADIATION EFFECTS ON ELECTRONIC CIRCUTS

attitude, control system, displays, etc., are simulated on the analog section. This application in general overlaps the Digital Subsystems grouping in that digital controllers such as reaction jet control systems are generally present. The predominant need, however, is that of high resolution in the trajectory calculations for journeys to the moon and planets. Conversion control, as in previous sections, should provide no limitation as to sampling rates for individual channels, groupings of A/D and D/A transfers, and allow for a large amount of one bit communication such as comparators, D/A switches, sense lines, and output control lines. Chapter 3 gives ample illustration of this type of simulation.

III. Summary

As discussed above, the spectrum of hybrid applications is a broad one, and one that is developing quite rapidly. With software becoming available in each category, the development of these and new applications will show even faster growth. The user knows what applications are of prime interest for his company now, and must govern himself accordingly. However, he usually does not know what might be vital to his company tomorrow. The user's main criteria should be to remain versatile and flexible, and use, to the greatest extent, the software and experience of the manufacturer and fellow users, and be aggressive in training personnel and educating management as to the benefits derived from hybrid simulation.

REFERENCES

1. BARNETT, R. M., "NASA Ames Hybrid Computer Facilities and Their Application to Problems in Aeronautics", International Symposium on Analogue and Digital Techniques Applied to Aeronautics, Liege, Belgium, (September 1963).

2. BAUER, W. F., and WEST, G. P., "A System for General Purpose Analog-Digital Computation", JACM, Vol. 4, No. 1 (January 1957), p. 12.

3. BIRKEL, G., Jr., "Hybrid Computers for Process Control", Communication and Electronics, No. 52, (January 1961) pp. 726-734; discussion, p. 734.

4. BURNS, A. J., and KOPP, R. E., "A Communication Link Between an Analog and a Digital Computer (DATA-LINK), Grumman Aircraft Engineering Corp., Research Department, Research Report RE-142, October 1960); ASTIA No. AD 244 913.

5. DEBROUX, A., and GREEN, C., and D'HOOP, H., "APACHE - A Breakthrough in Analog Computing," IRETEC, Vol. EC-11, No. 5, (October 1962).

6. GAINES, W. M. and FISCHER, P. P., "Terminology for Functional Characteristics of Analog-to-Digital Converters," Control Engineering, Vol. 8, No. 2, (February 1961), pp. 97, 8.

7. HALBERT, P. W., "Hybrid Simulation of an Aircraft Adaptive Control System," Proc. FJCC, Vol. 24, (1963).

8. HARTSFIELD, E., "Timing Considerations in a Combined Simulation System Employing a Serial Digital Computer," Proceedings of the Combined Analog Digital Computer Systems Symposium, Philadelphia, (December 16-17, 1960).

9. KORN, G. A., "The Impact of the Hybrid Analog-Digital Techniques on the Analog Computer Art," Proc. IRE, Vol. 50, No. 5, (May 1962).

10. LANDAUER, J. P., "Simulation of Space Vehicle with Reaction Jet Control System," (1962), EAI Bulletin No. ALHC 62515.

11. MITCHELL, B. A., "A Hybrid Analog-Digital Parameter Optimizer for ASTRAC-II", Proc. SJCC, Vol. 25, (April 1964).

12. MIURA, T., IWATA, J., "Effects of Digital Execution Time in a Hybrid Computer," Proc. FJCC, Vol. 24, (November 1963).

13. SHAPIRO, S., and LAPIDUS, L., "A Combined Analog-Digital Computer for Simulation of Chemical Processes, "Proceedings of the Combined Analog Digital Computer System Symposium, Philadelphia, (December 16-17, 1960).

14. SUSSKIND, A. K., "Notes on Analog-Digital Conversion Techniques," M. I. T., Technology Press, (1957).

15. WEST, C. P., "Computer Control Experience Gained from Operation of a Large Combined Analog-Digital Computation System, Proc. of Computers in Control Systems Conference, Atlantic City, (October 1957), p. 95.

16. WITSENHAUSEN, H., "Hybrid Simulation of a Tubular Reactor," (1962), EAI Bulletin, No. ALHC 6252-25.

17. WORTZMAN, D., "Use of a Digital/Analog Arithmetic Unit within a Digital Computer," Proc. of EJCC, (December 1960), pp. 269.

18. ZETKOW, G., and FLEISIG, R. (Grumman Aircraft Corp.), "Dynamic Analysis of OAO Spacecraft Motion by Analog-Digital Simulation," IRE Convention, Space Electronic Session, (March 1962).

SOFTWARE IN HYBRID COMPUTATION

I. Introduction

Today when a new digital computer is announced and marketed, the manufacturer of the equipment is required to advertise and deliver, as part of his product, automatic programming language systems and various programming aids if he expects to be competitive. The original need for these program packets with written procedures describing their use developed when it was found at an early stage that the serial nature of these machines present inordinate difficulties for the applications engineer. Coding, program testing and interpreting large volumes of digital results proved most tedious, time-consuming and expensive. Some very clever techniques have since been employed to reduce these undesirable activities. By using the computer to help in the preparation of programs (automatic programming), by taking advantage of previously written codes (library or canned routines), through automatic monitoring of a program during their execution (dumping, tracing, dynamic debugging) and by having the computer handle the needs of many users during a single time period (monitors, executive routines, an efficiency has been achieved through the creation of a programmed interface between the engineer and the machine. This interface, an adjunct to the hardware, is variously referred to as "software". This software development, therefore, by making digital computer programming easy for the non-computer-expert, is responsible for the almost universal acceptance of digital computer in the scientific field.

The development of this sophisticated software has made it possible to increase computer utilization, to gain wider use of computers with minimum training of personnel, and to reduce duplication of programming effort for programs of general utility. On the other hand, total dependance upon automatic programming, has the disadvantage of isolating the problem analyst, and even the programmer himself, from the computer. The analyst is restricted from communicating with his computer model while computation takes place. Similarly, the programmer is often limited in taking full advantage of the computer's special features.

In the analog computer field the reverse situation exists where no comparable "software" interface has grown up and the user communicates directly with the computer. There is little in the way of a requirement for the preservation of standard programs by analog computer programmers. As a result, the main feature of simulation by analog computation is that the problem analyst is directly involved in building the computer model and he maintains rapport with his model from origination to completion of the problem.

II. Hybrid Software

The role that software must play in hybrid simulation of large complex systems is evident. "Hybrid software" must ease the programmer's burden, as it does for the digital programmer, and at the same time it must bring the analyst closer to his model rather than isolating him from it. Hybrid software must include not only coding for the sequential computer but also inter-connection diagrams and prewired patch panels for the parallel machine. The following types of software are needed to support growth of hybrid computation to meet the simulation needs of today.

A. Compilers and Assemblers

Conventional compilers and assemblers, providing automatic conversion from a symbolic to a machine language program, have a useful place in hybrid computation. In general, programming systems for hybrid computation may differ from conventional systems in four ways:

(1) running time of the object program is minimized at the expense of compiling time.
(2) actual running time for each program statement is precalculated or estimated to aid the programmer with timing aspects of the problem.
(3) mostly machine-dependent programmer's language is utilized, since the programmer must be able to use special machine features to program for control of all interface operations.
(4) the programmer's language is extended to all sections of the hybrid system. Such software, applied to an efficiently designed hardware configuration, greatly facilitates the programmer's task in programming not only the digital, but also the analog and interface sections of a hybrid system. For example, setting analog function generators and potentiometers to predetermined values, initializing integrators may be performed by some pseudo-instructions of the assembly language.

B. Debugging Aids and Diagnostics

Conventional software aids for debugging a digital program include memory dump, updater and tracer routines, while diagnostics usually check out the proper operation of the digital processor hardware. Hybrid computer software, again, extends its capability to include debugging aids and diagnostics for the entire system. The following additional software of this type must be provided for efficient utilization of hybrid hardware:

(1) Analog section diagnostics

This is a symbolic program which accepts a problem-oriented language description of the analog flow diagram and produces automatically

(a) setup documents (pot sheets) for the analog run
(b) digital static check calculations for the analog program
(c) possibly a dynamic check calculation for the analog program

(2) Logic and Data Transfer Diagnostics

This digital program is used to ascertain that the linkage, interface and logic hardware are operating properly. It checks out

(a) the bidirectional data flow
(b) the bidirectional control signal flow
(c) the logic and secondary storage components, etc.

C. Utility Library

The conventional utility routines, prepared for a general-purpose hybrid computer include:

(1) mathematical routines
log, ln, exponential, sine, cosine, tangent, arctangent, square root
(2) computational routines
single precision floating point, double precision floating point, double precision fixed point

(3) conversion routines
(double and single precision) BCD/fixed point, BCD/ floating point, fixed point/floating point

(4) input/output routines
typewriter, paper tape, magnetic tape, automatic formatting on output.

(5) algorithms
a selection of numerical algorithms (integration methods, etc.)

(6) models
subsystems, environments, etc.

The last two categories require some additional explanation.

In a hybrid system one may envision frequent need for various numerical operations such as definite integrals, solution of differential equations, etc. Subroutines mechanizing various integration methods such as Runge-Kutta, Adams, Milne, etc. have been constructed. The different methods exhibit varying speeds and accuracies, and the user should choose to use those that fit best his problem. Furthermore, the hybrid simulation library should expand with routines for specific transfer functions of useful subsystem models that have become standard and are used in larger models; e.g., a typical servo controller. Another type of example is a function generator program for any number of aerodynamic functions. A standard program for a complete aerodynamic vehicle simulation is of value in a great number of aerospace computing laboratories.

III. HYDAC 2400 Software

EAI has had little need in the past to provide this type of support because of the parallel, engineer oriented aspects of the general-purpose analog computer. With the introduction of the HYDAC 2400 having, as a basic component, a straight digital computer EAI has chosen to improve its competitive position with respect to other manufacturers by:

(1) offering a standard hybrid computer
(2) providing comprehensive engineering backup and support for such a system
(3) initiating development of hybrid-oriented automatic programming languages and aids.

A. Basic Digital Software

1. Assembler - This is a metaprogram* which enables the user to write in symbolic machine language. A resultant translation provides the user with an edited listing of his original program and a condensed version of the code on punched paper tape. Several parts of a program may be "assembled" at different times later to be linked together and loaded automatically.

a. DAP - (Digital Assembly Program) - a metaprogram based on the SHARE standard assembler.

b. Macro-DAP - (for 8K-32K machines) - enables programmers to define aggregates of codes under a single heading and thereby reduce programming effort.

2. Interpreter - This is a monitor which executes pseudo-instructions to perform a given calculation. A complete new instruction repertoire is created for the engineering programmer which enables him to write in a DAP - like language but relieves him of the machine dependent aspects of DAP (input-output timing, scaling, storage allocation, etc.)

a. DIP - (Digital Interpretive Program) - a special translator-monitor which allows direct execution of symbolic program steps. Floating point calculation and input-output functions are automatic.

3. Executive - a resident groups of subroutines which helps the programmer to test and correct his program dynamically. It will perform such functions as loading new programs, printing partial results, automatically inserting or deleting instructions in the code, at the operator's direction. It may also contain standard input-output codes, interrupt bookkeeping routines, accounting routines and in fact all of those necessary for operation.

a. DEP - (Digital Executive Program)
  (1) Relocating and Linking Loader
  (2) Input-Output Control
  (3) Editing and Monitoring functions
  (4) Debugging Aids (Dumps, Snaps, Tracer)

---

* Metaprogram is a program used for the production of other programs.

4. Basic Subroutine Library - a set of subroutines which perform useful and standard calculations or manipulations which can be easily incorporated in the users own program. This library tends to grow as more and more diversified applications are programmed. A written procedure describing the requirements, inputs, outputs, range of computation, limitations of each such subroutine accompanies a paper tape version (either in symbolic DAP, FORTRAN or machine code).

a. Computational
ADDX, SUBS, MPYX, DIVX - Double precision fixed point
FADD, FSUB, FMPY, FDIV - Single and Double precision floating point
POLY - Odd polynominal evaluation (used by many math subroutines)

b. Mathematical

(1) Fixed point - single and double precision
SQDX, SQRX - Square Root
EXPX, EXDX - Exponential
LG2X, DPL2 - Logarithm (base 2)
LGEX, DPLE - Logarithm (base e)
LGAX, DPLT - Logarithm (base 10)
SINX, SNDX - Sine X (input in radians)
COSX, CODX - Cosine X (input in radians)
TANX, TNDX - Tangent X (input in radians)
ATNX, ATDX - Arctangent (output in radians)

(2) Floating Point - single and double precision

FSQR - Square Root
FEXP - Exponential
FLG2 - Logarithm (base 2)
FLGE - Logarithm (base e)
FLGA - Logarithm (base 10)
FSIN - Sine (input in radians)
FCOS - Cosine (input in radians)
FTAN - Tangent (input in radians)
FATN - Arctangent (output in radians)

c. Conversion - single and double precision

XBXD - Binary-to-decimal (fixed point)
XDXB - Decimal to binary (fixed point)
FBFD - Binary to decimal (fixed to floating point)
FDFB - Decimal to binary (floating to fixed point)

FBXD	-	Binary to decimal (floating point)
FDXB	-	Decimal to binary (floating point)

d. Input-Output

PTIN, PTOU	-	Paper tape I/0
TYIN, TYOU	-	Typewriter I/0

Routines for on-line printer, magnetic tape, light pen/scope, disk are also available.

e. Utility and Debugging

DUMP - output to typewriter, paper tape, printer magnetic tape in octal, decimal, alphanumeric or mnemonic instruction format.

DEBUG - Program testing communications package for on-line typewriter debugging - incorporates ability to modify or output data, search or clear memory, input correction tapes or output self loading dump tapes, set breakpoints, etc.

TRACER - Enables an automatic step-by-step monitored execution of a program. Intermediate register results are printed on the typewriter or on-line printer.

UPDATE - Enables the updating and correction of symbolic program tapes and/or data tapes.

UPSUB - Enables updating of the tape containing the library of subroutines - inserts, deletes and duplicates.

LOADER - Used to load absolute and relocatable DAP and/or FORTRAN object (binary output) taped programs. Automatically scans the library tape for standard subroutines and loads and links them to main program. It is the 375 equivalent of the SHARE 704/709-90 BSS Loader.

MISC - Quick-Look-and-Change, Insert (formatting editor), Relocatable Dump/Punch (octal output), Memory Clear, Bioctal Dump for self loading tapes, Tape-to-Tape conversion (for tapes with a code different from 375 input BCD code). One would have to see the writeups for the above in order to gain insight into their use.

5. Compiler - a metaprogram which translates directly into machine code the statement of a problem. An artificial language created specifically for expressing easily certain types of computations and/or logical manipulations (i. e. algebraic or engineering problems), together with a language translator is implemented on a computer. Many problem-oriented languages (POL's) exist. One such language, FORTRAN (FORmula TRANslator) is accepted in the U.S. as standard and an implementation exists for almost every machine. (N.B. Unfortunately no minimum standard definition has been accepted for the FORTRAN language. All implementations of it vary in scope for different computers, hence it is not machine-independent. Unless the programmer takes extreme care in using compatible subsets of the language(s) he cannot achieve equivalent results for parallel computations on dissimilar machines.)

a. FORTRAN - An engineering/algebraic compiler, easy to use and learn, flexible automatic formatting of input-output, easy to debug. Object programs compatible with DAP condensed output tape format.

(1) In-Line Machine Language Coding with FORTRAN.

The 375 FORTRAN Compiler provides special provision for real-time and hardware-oriented programming. A capability to write DAP-like entries "in-line" with the algebraic statements gives the programmer a control over computation not afforded by any competitive FORTRAN system. This ability is a necessity if any compiler metaprogram is to be used to generate the codes for a hybrid computation.

(2) Expanded features of 375 FORTRAN.

Although compatible with a generally accepted FORTRAN II, this 375 compiler has been extended to include the most desirable features of a new FORTRAN IV version as defined by IBM for its large scale computers.

B. Expansion of 375 Software for Hybrid Computation

1. Mathematical Library

EAI has provided many diversified subroutines for types of numerical integration. (Runge-Kutta, Adams, Milne, Euler, Predictor-

Corrector, etc) which exhibit varying speeds and accuracies. The user may choose to use those which fit best his problem or he may incorporate several at once to be optionally selected during the course of a run. This selection may be under programmed or manual control. Routines for high speed function generation and non-linear transfer functions are also available.

2. Automatic Programming Aids

HYTRAN (HYbrid TRANslator) will be used to refer to a family of metaprograms being implemented on the HYDAC 2400 computer to provide digital assistance in the programming of analog consoles. The three HYTRAN programs being written are:

(1) An Interpretive Static Check Generator
(2) As an extention of (1), a Documenting program
(3) An On-Line Diagnostic Generator

The first program reads and interprets the physical problem and the analog program description which is punched on paper tape in an analog-oriented (HYTRAN) language, computers parameters, initial conditions, and both the physical and the voltage Static Check. It examines the two Static Check values for consistency, pin-pointing any discrepancy or overload at the component level, and finally outputs an alphabetic list of static check values.

The second generates complete documentation of the analog program, featuring Pot-Sheets, Amplifier-Sheets, Alphabetic lists of all Parameters and Variables, Cross-Reference, and Pot-Setting and a Static Check tape in ADIOS* format.

The third and final program checks the measured Static Check Voltages for consistency with the analog circuit diagram and specifications of the analog components, providing a rapid means of locating mispatching or component failures.

Many additional HYTRAN metaprograms are planned including other variations on the APACHE, DAS and MIDAS efforts.

---

* Analog Digital Input-Output System

3. HYDAC - ALGOL - a self-compiling compiler to be used for the definition of a command and control language for HYDAC 2400. This special type of translator has flexibility enough to enable the user to generate his own software with relative ease. Future expansions in the hybrid hardware can be immediately reflected in an updated version of this real-time programming tool. (N. B. the implementation is to be based on the NELIAC system as developed for real-time command and control at the Naval Electronics Laboratory, San Diego, California)

C. HYDAC 2400 Software - (Operating on 375/DOS/231-R-V)

1. Dual-Processing Executive Program

For hybrid systems with at least 8K 375, a provision is made through hardware modification for independent computation of two problems in tandem. The 375 memory and I-O equipment may be shared and a hybrid computation can continue at the same time a purely digital computation (i. e. a FORTRAN compilation) is being processed. No interference between the two co-residents will occur because the storage lockout/protect mode is controlled externally through the logic of the DOS. The Executive Program resides in 375 memory and services the interrupts and performs other important housekeeping functions.

This feature is very important where the checkout and execution of a hybrid computation dominates the time available for using the 375 for program preparation. By time-sharing on HYDAC 2400 a greater efficiency is realized over any competitive system.

2. Modifications to DAP and FORTRAN for control of DOS/375 interface and further control of 231-R-V are anticipated. These additions allow symbolic programming to include:

a. Ability to perform ADIOS functions
b. Real-time iterative 231-R-V mode control through DOS
c. Serial Data Transfer Unit* and basic system control
d. 315 conversion system* control

3. HYDAC Monitor System (for dynamic program debugging). A set of hardware and software procedures are developed for automatic program checkout. These include such debugging packages as:

---

* Sections of the DOS 350 Computer

- a. Tracer for 375
- b. DOS memory dump and selective printout
- c. AERO scan and 231-R-V readout and selective print
- d. Automatic pot setting and checking
- e. GPAC Check Program execution. (This monitor will be incorporated with the DEP (Digital Executive Program) for standardization of on-line debugging techniques.)

4. Miscellaneous Hybrid Utility Programs - These are being developed as their need arises. Some are already available.

- a. Tape-to-Tape Conversion (ADIOS/DOS/375)
- b. Digital Plotting Subroutines. Strip Chart, X-Y, CRT
- c. DOS/375 direct memory linkage software
- d. DOS as satellite digital computer for 375 for use as:
  - (1) Digital Resolver
  - (2) Auxiliary input-output device
  - (3) Additional memory and logic capability for parallel computation
  - (4) An ultra high-speed function generator
- e. Use of DOS as real-time interrupt and control computer for 375 during complex hybrid simulations

IV Conclusions

EAI, through its own growing experience in hybrid computation, provides the HYDAC users with the necessary programming tools. Software never has solved a problem by itself. It can be used to get the solution sooner, however. Unique facilities are provided through the metaprograms for the user to produce his own software -- often reflecting his applications and parochial needs. There is no end to this process. As knowledge in computation grows, more difficult and even impossible, problem solutions are sought and solved. Software is improved, expanded and newly created in the customer's own laboratory. The preponderance of useful programming aids has always originated with users of the machines. This manufacturer hopes never to reverse this trend. By providing a comprehensive software package in the beginning, the hope is to get the users "on board" more quickly and to turn out useful work in short order. EAI points with ever increasing pride to the solutions which can be and have been implemented on the hybrid computer.

APPENDIX A

I. Digital Computer Equations

A. Basic Translational Equations in Local Horizontal System

The translational equations are solved in a local horizontal coordinate system with axes along the north, east, and radial directions. Fig. 22, Chapter 3 shows this coordinate system.

$$\ddot{r} = \frac{F_r}{m} - g_r + (\dot{\xi}\cos\Omega)V_E + \dot{\Omega}V_N \tag{1}$$

$$\dot{V}_N = \frac{F_N}{m} + g_N - (\dot{\xi}\sin\Omega)V_E - \dot{\Omega}\dot{r} \tag{2}$$

$$\dot{V}_E = \frac{F_E}{m} + (\dot{\xi}\sin\Omega)V_N - (\dot{\xi}\cos\Omega)\dot{r} \tag{3}$$

B. Aerodynamic Force Calculation and Transformation

The aerodynamic forces are calculated with respect to the relative velocity vector (lift, drag, and y axis acceleration) and then transformed into the local horizontal coordinate system.

$$L = qSC_L \tag{4}$$

$$D = qSC_D \tag{5}$$

$$Y = qSC_Y \tag{6}$$

$$\frac{F_r}{m} = \left(\frac{L}{m}\cos\mu - \frac{Y}{m}\sin\mu\right)\cos\gamma - \frac{D}{m}\sin\gamma \tag{7}$$

$$\frac{F_N}{m} = -\left(\frac{L}{m}\sin\mu + \frac{Y}{m}\cos\mu\right)\cos\sigma - \left[\left(\frac{L}{m}\cos\mu - \frac{Y}{m}\sin\mu\right)\sin\gamma + \frac{D}{m}\cos\gamma\right]\sin\sigma \qquad (8)$$

$$\frac{F_E}{m} = \left(\frac{L}{m}\sin\mu + \frac{Y}{m}\cos\mu\right)\sin\sigma - \left[\left(\frac{L}{m}\cos\mu - \frac{Y}{m}\sin\mu\right)\sin\gamma + \frac{D}{m}\cos\gamma\right]\cos\sigma \qquad (9)$$

C. Angular Rate and Position Calculation

The derivatives of the space fixed coordinates Ω and ξ are solved as functions of the inertial velocities V_N and V_E. The derivative of the longitude, is calculated by subtracting the earth's rotational rate from $\dot{\xi}$. Since the earth rotates in the logitudinal direction Ω is the latitude. Altitude is calculated with respect to an oblate earth.

$$\dot{\Omega} = \frac{V_N}{r} \qquad (10)$$

$$\dot{\xi} = \frac{V_E}{r\cos\Omega} \qquad (11)$$

$$\dot{\lambda} = \dot{\xi} - \omega_o \qquad (12)$$

$$h = r - r_E + \frac{r_E e^2}{2(1-e^2)}\sin^2\Omega \qquad (13)$$

D. Velocity, Velocity Heading and Flight Path Angle Calculations

The total relative (to the earth) velocity is calculated from the velocity components in the N, E and radial directions. The horizontal relative velocity has two components, east and north. The easterly relative velocity is the difference between V_E and the tangential rate of the earths rotation. The relative northerly velocity is the same as the northerly inertial velocity. The flight path angle γ, is the angle between the horizontal and the total velocity vectors. The velocity heading angle σ, is measured from east to north.

$$V = \sqrt{V_{ER}^2 + V_{NR}^2 + \dot{r}^2} \qquad (14)$$

$$V_H = \sqrt{V_{ER}^2 + V_{NR}^2} \qquad (15)$$

$$V_{ER} = V_E - r\omega_0 \cos\Omega \qquad (16)$$

$$\sin\gamma = \frac{\dot{r}}{V} \qquad (17)$$

$$\cos\gamma = \frac{V_H}{V} \qquad (18)$$

$$\gamma = \tan^{-1}\frac{r}{V_H} \qquad (19)$$

$$\sin\sigma = \frac{V_N}{V_H} \qquad (20)$$

$$\cos\sigma = \frac{V_{ER}}{V_H} \qquad (21)$$

$$\sigma = \tan^{-1}\frac{V_N}{V_{ER}} \qquad (22)$$

E. Density and Dynamic Pressure

The density is tabulated in square root form to compress the range of variation and was taken from U.S. Standard Atomophere, 1962. The dynamic pressure is a function of density and velocity.

$$\sqrt{\rho} = \sqrt{\rho}(h) \tag{23}$$

$$q = \left(\frac{1}{\sqrt{2}}\sqrt{\rho}\, V\right)^2 \tag{24}$$

F. Gravity Calculation

The gravity components are calculated for an oblate spheroid shaped earth. Since the earth is assumed to be rotationally symetrical, $g_E = 0$

$$g_R = -\frac{GM}{r^2}\left[1 + \frac{3Jr_E^2}{r^2}\left(\frac{1}{3} - \sin^2\Omega\right)\right] \tag{25}$$

$$g_N = -\frac{2GMJr_E^2}{r^4}\sin\Omega\cos\Omega \tag{26}$$

G. Heat Transfer and Temperature Equations

These equations are generated in the digital program and transferred to the analog for calculating parameters in the flight control equations.

$$\dot{q}_{cs} = \frac{2.70893}{\sqrt{R}}\sqrt{q}\left(\frac{V}{10^4}\right)^2 \tag{27}$$

$$\dot{q}_{RS} = 144.9\, R\, \rho^{1.57}\left(\frac{V}{10^4}\right)^{17} \tag{28}$$

$$\dot{q}_c = \dot{q}_{cs}\cos^{1.5}\alpha \tag{29}$$

$$\dot{q}_R = \dot{q}_{RS}\cos^6\alpha \tag{30}$$

$$T_s = \left[\frac{1}{\epsilon\sigma_B}(\dot{q}_{RS} + \dot{q}_{cs})\right]^{\frac{1}{4}} \tag{31}$$

$$T = \left[\frac{1}{\epsilon\sigma_B}(\dot{q}_R + \dot{q}_c)\right]^{\frac{1}{4}} \tag{32}$$

$$\dot{T}_s = \frac{T_s}{4}\left(3\frac{\dot{V}}{V} + \frac{\beta_e \dot{r}}{2}\right) \tag{33}$$

$$q_{TS} = \int_0^t (\dot{q}_{RS} + \dot{q}_{cs})\,dt \tag{34}$$

$$\beta_e = \frac{1}{\rho}\frac{d\rho}{dh} \tag{35}$$

H. Target Heading Angle, Range-to-go and Cross Range-to-go

The range and heading angles are calculated using spherical trigonometry and the resulting angles are multiplied by the earth radius yielding the range.

$$\tan\sigma_T = \frac{\cos\Omega\tan\Omega_T - \sin\Omega\cos(\lambda_T - \lambda)}{\sin(\lambda_T - \lambda)} \tag{36}$$

$$\tan\theta_{RTG} = \frac{\sin(\lambda_T - \lambda)}{\cos\sigma_T\left[\tan\Omega_T\sin\Omega + \cos\Omega\cos(\lambda_T - \lambda)\right.} \tag{37}$$

$$RTG = (\theta_{RTG})\,r_E$$

$$\tan CR = -\tan(\sigma_T - \sigma)\sin\theta_{RTG} \tag{38}$$

$$CRTG = (CR)\,r_E$$

I. Heading Error

The angle between the great circle heading to the target, σ_T, and the velocity heading, σ.

$$\text{Heading Error} = \sigma_T - \sigma \qquad (39)$$

J. Guidance Equations

Error equations for use in the TRFCS

$$DR_E = RTG - RTG_d \qquad (40)$$

$$RTG_d = f_1(V)$$

$$CR_E = CRTG - CRTG_d \qquad (41)$$

$$CRTG_d = f_2(V)$$

II. Analog Computer Equations

A. Angular Acceleration

The following equations are calculated in the body system:

$$\dot{P} = -\left(\frac{I_Z - I_Y}{I_X}\right) QR + \frac{L + M_X}{I_X} \qquad (42)$$

$$\dot{Q} = \left(\frac{I_Z - I_X}{I_Y}\right) RP + \frac{M + M_Y}{I_Y} \qquad (43)$$

$$\dot{R} = -\left(\frac{I_Y - I_X}{I_Z}\right) PQ + \frac{N + M_Z}{I_Z} \qquad (44)$$

B. Aerodynamic Torques

These parameters are generated by the moment of aerodynamic forces.

$$L = qSb\left[C_{\ell_\beta}(\alpha)\beta + C_{\ell_{\delta_r}}\delta_r + C_{\ell_{\delta_a}}\delta_a\right] \tag{45}$$

$$M = qSc\left[C_m(\alpha, \delta_E)\right] \tag{46}$$

$$N = qSb\left[C_{n_\beta}(\alpha)\beta + C_{n_{\delta_r}}(\alpha)\delta_r + C_{n_{\delta_a}}(\alpha)\delta_a\right] \tag{47}$$

C. Euler Angles

Generated for local horizontal to body axis transformation.

$$\theta = \int (Q\cos\phi - R\sin\phi)\,dt \tag{48}$$

$$\phi = \int \left[P + \tan\theta(Q\sin\phi + R\cos\phi)\right] dt \tag{49}$$

D. Attitude Rate Equations

Generated in this form for use in the heat sensor equations.

$$\begin{aligned}\dot{\alpha} = {} & \left(\frac{g_r}{V} - \frac{V_I^2}{Vr}\right)(\cos\theta\cos\phi\cos\alpha + \sin\theta\sin\alpha) - \frac{L}{mV} \\ & + Q - (P\cos\alpha + R\sin\alpha)\beta\end{aligned} \tag{50}$$

$$\begin{aligned}\dot{\beta} = {} & \left(\frac{g_r}{V} - \frac{V_I^2}{Vr}\right)(\cos\theta\sin\phi) - R\cos\alpha + P\sin\alpha + \frac{Y}{mV} \\ & + \frac{D}{mV}\beta - \left(g_r - \frac{V_I^2}{r}\right)(\cos\theta\cos\phi\sin\alpha - \sin\theta\cos\alpha)\beta\end{aligned} \tag{51}$$

$$\dot{\mu} = P\cos\alpha + Q\beta + R\sin\alpha \tag{52}$$

E. Heat Sensor Equations

These short period equations are used for control and are discussed in the text.

$$\dot{T} = \dot{T}_S(1 - .1875\alpha^2) - .375\alpha\dot{\alpha}T_S \tag{53}$$

$$\Delta T = .021 T_S \beta \tag{54}$$

$$\Delta\dot{T} = .021(T_S\dot{\beta} + \dot{T}_S\beta) \tag{55}$$

F. TRFCS Control Equations

The control parameters include the variable surfaces and the commanded bank angle.

$$\delta_E = f_1(\dot{T}) \tag{56}$$

$$\delta_a = K_1(\mu - \mu_c) + K_2 P + K_3\delta_R \tag{57}$$

$$\delta_R = f_2(\Delta\dot{T}) + K_4\delta_a \tag{58}$$

$$\mu_c = f_3(\dot{T}) \tag{59}$$

APPENDIX B

DEFINITION OF SYMBOLS

Variable	Definition
$\dot{V}_N$	acceleration north (local level coordinates)
$\dot{V}_E$	acceleration east (local level coordinates)
$\ddot{r}$	acceleration up (local level coordinates)
$\frac{F_N}{m}$	aerodynamic acceleration north
$\frac{F_E}{m}$	aerodynamic acceleration east
$\frac{F_r}{m}$	aerodynamic acceleration up
g_r	gravitational acceleration up
g_N	gravitational acceleration north
$\frac{L}{m}$	lift acceleration
$\frac{D}{m}$	drag acceleration
$\frac{Y}{m}$	side force acceleration
V_N	velocity north, inertial
V_E	velocity east, inertial
$\dot{r}$	radius rate
V_I	total inertial velocity

Variable	Definition
V	total relative velocity
V_H	relative horizontal velocity
V_{NR}	velocity north, relative (equals V_N)
V_{ER}	velocity east, relative
M	mach number
$\dot{\Omega}$	latitude rate
$\dot{\xi}$	longitude rate, inertial
$\dot{\lambda}$	longitude rate, relative
$\dot{\alpha}$	rate of change of angle of attack
$\dot{\beta}$	rate of change of angle of sideslip
$\dot{P}$	roll angular acceleration
$\dot{Q}$	pitch angular acceleration
$\dot{R}$	yaw angular acceleration
P	roll rate about the body x-axis
Q	pitch rate about the body y-axis
R	yaw rate about the body z-axis
$\left.\begin{matrix}\dot{\theta}\\ \dot{\phi}\end{matrix}\right\}$	Euler rates of body axis with respect to local horizontal
L	aerodynamic torque about the body x axis
M	aerodynamic torque about the body y axis

Variable	Definition
N	aerodynamic torque about the body z axis
M_X	reaction jet moment about the body x axis
M_Y	reaction jet moment about the body y axis
M_Z	reaction jet moment about the body z axis
Ω	latitude
λ	longitude
α	angle of attack
β	angle of sideslip
μ	bank angle (lift vector with vertical plane)
ϕ	body roll angle
θ	body pitch angle
ψ	body heading angle
$\dot{\mu}$	bank angular rate
δ_E	elevator deflection angle
δ_a	aileron deflection angle
δ_r	rudder deflection angle
γ	flight path angle ($+$ for $\dot{r}$ $+$)
σ	velocity heading angle (to the north from east)
σ_T	heading to target ($+$ to the north from east)
ρ	air density
r	radial distance from earth center to vehicle

Variable	Definition
h	altitude
DR_E	down range error
CR_E	cross range error
RTG	range-to-go
CRTG	cross-range-to-go
C_Y	side force coefficient
q	dynamic pressure
C_L	lift coefficient
C_D	drag coefficient
C_M	pitching moment coefficient
$C_{\ell\beta}$	rolling moment coefficient due to sideslip
$C_{n\beta}$	yawing moment coefficient due to sideslip
$C_{\ell\delta a}$	rolling moment coefficient due to aileron deflection
$C_{n\delta a}$	yawing moment coefficient due to aileron deflection
$C_{n\delta r}$	yawing moment coefficient due to rudder deflection
$C_{\ell\delta r}$	rolling moment coefficient due to rudder deflection
$\dot{q}_{rs}$	radiative heat transfer rate at stagnation point
$\dot{q}_{cs}$	convective heat transfer rate at stagnation point
$\dot{q}_r$	radiative heat transfer rate at sensor location
$\dot{q}_c$	convective heat transfer rate at sensor location

Variable	Definition
q_{Ts}	total integrated heat flux at stagnation point
$\dot{T}$	temperature rate at sensor location
$\dot{T}_S$	temperature rate at stagnation point
T	temperature at sensor
T_S	temperature at stagnation point
$\Delta\dot{T}$	differential temperature rate between wing sensors
GM	earth gravitational constant
J	earth oblateness factor
r_e	radius of the earth-equatorial
e	eccentricity of the earth
ω_o	rotation rate of the earth
ϵ	emissivity
σ_B	Stefan Boltzman constant
g_o	sea level gravity
m	mass of the vehicle
S	wing area
b	wing span
c	wing chord
I_X	moment of inertial about x axis

Variable	Definition
I_Y	moment of inertia about y axis
I_Z	moment of inertia about z axis
R	nose cap radius
β_e	exponential index of the atmosphere