DESIGNERS' MANUAL

FOR

CIRCUIT DESIGN BY ANALOG/DIGITAL TECHNIQUES

by

Charles H. Beck and Ming H. Kuo
Department of Electrical Engineering

July 1970

Prepared under Contract No. NAS 12-2059 by

Systems Laboratory
School of Engineering
Tulane University

for

Electronics Research Center
National Aeronautics and Space Administration
575 Technology Square
Cambridge, Massachusetts 02139

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES

LIST OF TABLES

Chapter I

INTRODUCTION

Computer techniques are essential to the efficient design of complex electronic circuits. This is especially true in the case of integrated circuits since it is often difficult and expensive to predict circuit response accurately by experimental "breadboard" testing. Certain hybrid techniques offer the fast solution times of the analog computer for dynamic analysis, automated problem setup on the analog controlled by digital subroutines, the capability of including actual physical circuit devices in the simulation thus reducing the digital memory and the number of nonlinear analog computing elements required, and the decision and arithmetic capabilities of the digital computer required for optimization and sensitivity calculations. The availability of digital computer subroutines and such methods as those based on the separation principle cut the programming time formerly required in the case of analog computation.

Hybrid techniques applied to system design require that the system simulation and the various computations required to effect the design be partitioned and that each operation be appropriately assigned to either the analog or the digital computer. As a result of the effort in computer-aided design at Tulane University, a hybrid computer system has been developed, and hybrid techniques have been applied to circuit design. These techniques serve as useful supplements to the basic digital NASAP*[1,2]

*Network Analysis System Applications Program developed by NASA/Electronics Research Center.

program for the purpose of designing models of electronic devices, performing transient analysis of linear and nonlinear circuits, and the direct design of dynamic systems based on design specifications.

In most computer studies of physical systems, assumed mathematical models of the systems are used so that off-line computations can be performed. The absence of accurate models practically precludes any systematic analytical treatment and raises questions about the validity of computerized designs. This report concerns results from employing high-speed, fully-automated techniques for obtaining models of systems based on time-domain measurements or specifications. These techniques do not require the usual assumptions such as low order, a priori knowledge of model form, fixed parameters, minimum phase, and linearity.

This manual is part of a series covering design areas which include aerospace circuits, instrumentation circuits, communication circuits, filters, etc.[3] These manuals have been developed to assist in the use of NASAP and various supplementary techniques.

1.1 References

1. McNamee, L. P. and H. Potash, "A User's Guide and Programmer's Manual for NASAP", Report No. 68-38, University of California, Los Angeles, August 1968.

2. Rooney, C. J. and E. W. Weber, "Application of NASAP to the Design of Communication Circuits," Final Technical Report, Contract NAS 12-650, Illinois Institute of Technology, Chicago, Ill., May 1969.

3. Happ, W.W., "Flowgraph Techniques for Closed Systems," IEEE Transactions on Aerospace and Electronic Systems, vol. AES-2, no. 3, pp. 252-264, May 1966.

Chapter II

DESIGN CRITERIA

This chapter covers material on the basic design criteria which relate to the hardware system, software, and design methods used during the course of this study.

2.1 Hardware Requirements

A hybrid computer system consists of a general-purpose digital computer and a general-purpose analog computer interconnected through a conversion and control linkage system plus various input/output devices. This type of computing system has distinct advantages and disadvantages compared with either pure analog or digital computers for certain classes of problems. It is remarkable that most of the desirable characteristics of both analog and digital computers are conserved in hybrid systems[1,2]. Minimal requirements for hybrid computer hardware are as follows:

1) An analog computer with buffered digital controlled parameter units and integrator mode and time scale control, high-speed overload detectors, and digital controlled patching of a portion of the analog program.
2) A digital computer with buffered input and output registers.
3) A linkage system with a multiplexed A-D converter; buffered D-A converters; and control, interrupt, trunk, and sense lines.

The hybrid computer system at Tulane University consists of four EAI TR-48, two EAI 16-31R, and one EAI 231R analog computer, a Univac AN/GSK-1

digital computer, and a flexible linkage system. The hybrid computer system diagram is shown in Figure 2-1.

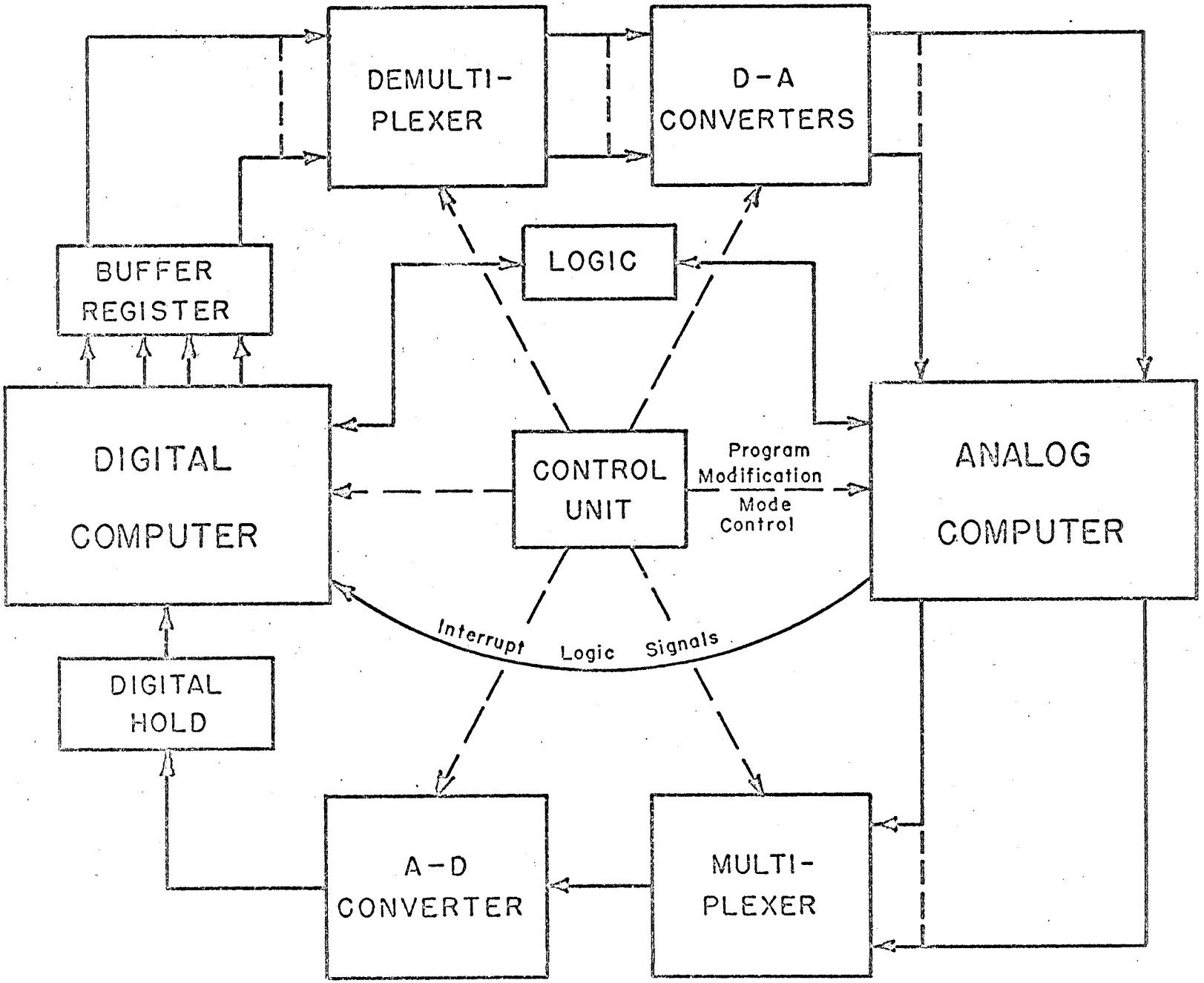
2.1.1 Analog Computer Section

Analog computation is involved basically with time domain information in continuous form. Since total solution times are commonly of the order of a few milliseconds, an oscilloscope is often used to display the continuous dynamic output response for photographic recording. Direct visual observation is possible with a storage oscilloscope, or iterative solutions can be executed to provide for viewing on a non-storage oscilloscope.

Representation of information in continuous form eliminates such problems as round-off error which is so troublesome in iterative computations using digital computers. The accuracy of analog computation is limited by the precision with which a quantity can be represented and measured on the computer. Analog computer accuracy is ordinarily limited to approximately 0.01% of full-scale by the tolerance of computer components.

This type of information can be transmitted and used without requiring costly devices such as the registers which provide access to the main memory unit of digital computers, hence time-sharing of hardware is unnecessary for many problems. Because of the continuous manner and economy of this type of computation, it is common practice to use separate computing elements to implement every similar mathematical function of different arguments as well as the different functions of a given argument. This parallel or simultaneous operation of all computing elements such as summers, integrators, and multipliers is the primary

Figure 2-1. TULANE HYBRID COMPUTER SYSTEM



reason for the high computing speed that is possible with analog computers. Practically instantaneous execution is limited mainly by the bandwidth of the computing elements rather than by the complexity of the problem.

Digital controlled parameter units (DPU) have been added to the computing units of the analog computers to provide for automatic adjustment of circuit and performance index parameters, scaling of the analog program, and weighting factors used in the optimization programs. Mode and time scale interfaces have also been developed to provide for automated sensing of the nonlinear operation of any amplifier computing unit. Direct circuit design including automated structure manipulation has necessitated the development of digital controlled patching of appropriate portions of the analog program.

2.1.2 Digital Computer Section

The digital computer provides the capability of performing arithmetic computations, logical decisions, data storage, and modification of a program on the basis of computations. These features permit the capability of using stored programs, nonlinear function generation, and time delay of a sampled waveform. An additional important characteristic is the inherent precision which is limited only by the number of bits used in the memory. Computational accuracy is further dependent on the particular numerical algorithm used.

The Univac digital computer, originally used for USAF missile guidance, was obtained as Government surplus property. The chief merit of this computer is its high reliability resulting from the requirements of the Titan I missile weapons system. It does have several buffered

input and output registers which provide for transfer of the necessary data and control information.

2.1.3 Linkage System

The conversion and control linkage system expands the storage capacity of the digital computer to effectively include the analog computer and associated peripheral analog devices and systems. In addition, this interface permits the digital computer to perform many of the functions of a human operator relative to the analog computer and associated equipment.

This unit provides for encoding and decoding of information which is transmitted between portions of the system, for logic operations, and for appropriate routing of control and information channels. The linkage system has the following three modes of operation: control, conversion, and logic. The control mode may take on any of three possible forms. In mode CMXXX, control signals are passed from the A-register on the digital computer to the analog integration mode control inputs specified by the three least significant digits of the linkage mode status word. In mode CTXXX, control signals are passed from the D-register to the analog integrator time scale control inputs specified by the three least significant digits of the linkage mode status word. In mode CPXXX, control signals are passed from the S-register to the DPU specified by the three least significant digits of the linkage mode status word.

In the conversion mode, both A-D and D-A operation are possible. In mode ADXXX, address signals are passed from the D-register on the digital computer to the multiplexer. Linkage mode status word ADPXX is used to control the operation of the A-D converter. In mode DAXXX,

address signals from the D-register cause the D-A converter specified by the three least significant digits of the linkage mode status word to be selected. The S-register supplies the data word to the D-A converter.

The logic mode provides the ability for the digital computer to send or receive logic signals. This provides an alternate means for controlling the operation of the analog computers. Logic signals such as synchronization and overload signals originating on the analog computers can be sensed by the digital computer to provide interrupts.

2.2 Software Requirements

Since a hybrid computer provides for simultaneous use of an analog computer and a digital computer, hybrid systems offer all the advantages of both analog and digital computers. If the digital computer has at least a 32K core memory, then it can also serve as a stand-along digital computer to provide for execution of a digital computer-aided circuit analysis program such as NASAP. However, since most hybrid computer systems currently have no more than 16K core memory, their appropriate use relative to existing computer-aided design programs is supplementary in nature.

For circuit design problems where no synthesis procedures are available, optimization techniques can often be used to advantage[3,4]. Since optimization requires that parameters be repetitively adjusted until the best design has been obtained, the number of iterations may be large. Dynamic circuit simulation performed on the analog computer portion of the hybrid system is by far the most efficient portion of the computation. Since the analog computer execution time is independent of circuit

complexity or the order of the system, one means of accomplishing a reduction in execution time is to emphasize the analog portion of the circuit design algorithm. This high speed characteristic of the analog portion permits the increased use of elementary optimization techniques, which require larger numbers of iterations, to achieve faster solutions than possible with more efficient all-digital optimization techniques which require fewer iterations. With simple optimization algorithms there is little need for other than machine language programming. Besides, this makes it possible to increase the execution efficiency of the digital computer portion of the program. The remaining portion of the digital computer program which primarily accomplishes control operation can also be appropriately written in machine language with little effort.

This unshphisticated programming requirement is especially desirable because individual hybrid computers differ considerably. When automatic patching of the analog portion and universal hybrid software become commonly available as is the case for digital computers, then the case for machine language programming may no longer hold. The programming of the dynamic system simulation on the analog computer is accomplished in a simple and straight forward manner as described in Chapter III.

In general, hybrid software is required to provide a convenient means for mechanizing sets of ordinary and/or partial differential equations. Software for hybrid computation is often required to provide for real-time and time-critical operation. Real-time operation is required in such cases as those where actual transistors and diodes serve as computing elements in the network simulation to represent corresponding network devices. The software must also assure synchronization of the

analog and digital computer operation. There must be provision for the programmer to control time-critical computations for suitable operations.

The major disadvantage of hybrid computation is the requirement of hardware-oriented real-time programming.

2.3 Hardware-Software Trade-Off Possibilities

Exploitation of computer-aided circuit design techniques generally starts with an assumed mathematical model of the circuit to be designed so that off-line computations can be performed. It is believed that the availability of accurate device models is crucial for any systematic analytical treatment, and the absence of such models practically precludes the justifiable use of the computer for circuit design. It is also desirable to obtain the simplest possible device models of sufficient accuracy since higher-order models limit the size circuit that can be treated using either analog or digital computers. Since execution time for digital computer programs increases for higher order models, the expense of computer-aided design may become economically prohibitive for certain types of calculation such as those which involve optimization techniques.

The programming of the circuit simulation on the analog computer is primarily accomplished by substituting analog computing elements for corresponding elements or parameters of the physical circuit being studied. However, when accurate models are required, actual physical electronic devices can also be included as computing elements in the circuit simulation as described in Chapter III.

2.4 Performance Indices

Computer-aided circuit design usually involves a step-by-step process

of repeated analysis[5]. It is often advantageous to simulate the circuit including the equivalent circuit representations of active devices[6-8]. The model is then analyzed for a given set of parameter values, and the results are compared with the specified design criteria. Based on the findings of this comparison and the designer's past experience, new parameter values that are expected to improve the design are chosen. An iterative process is continued until the prescribed tolerance figure is met. This technique requires involvement of the designer, and it lacks throughput speed.

It appears that relatively few computer-aided techniques are available for direct circuit design. If the circuit designer constructs an algorithm which accomplishes the required decision making process involved in step-by-step repeated analysis and parameter adjustment, then an automated direct design is possible. This permits designers to be freed from much that is routine so that their experienced engineering judgment can be directed to efficient effort such as evaluation of the final design.

In obtaining the best values for the model parameters, it is necessary to establish a comparison criterion or performance index. An obvious choice is an index based on the difference (error) between the actual or desired transient response and the transient response of the model to the specified input. The integral of the squared error (ISE) has been used extensively in this project as well as the integral of the absolute error (IAE). Other criteria can also be used with equal ease since an analytical solution is not required[9]. The chosen performance index can be computed on the analog computer and returned to the digital portion of the system, or it can be computed on the digital computer. The optimization algorithm operates upon this quantity.

2.5 Data Requirements

The internal processes of electronic devices are not of specific interest for many circuit design problems, but rather the influence on external performance. In these cases, device models only need to reproduce the desired terminal characteristics. Emphasis will then be placed on obtaining the simplest models which meet the required specifications subject to certain constraints such as physical realizability and the range of allowable parameter values.

Input data are in the form of continuous time-domain input-output measurements or specifications. The network topology or a mathematical model is programmed on the analog computer as described in Chapter III.

2.6 Design Flow Diagram

The first step in the network design or device modeling process is the selection of a possible network or device model. The form of this model will be influenced by the specifications of the problems, the available technology, the experience of the designer, and the allowable design techniques. For difficult design problems where no synthesis procedures are known, optimization techniques performed with the aid of computers have proven to be useful for obtaining the best set of parameters for a given model form. An iterative process of repeated analysis and parameter variation continues until the optimum set of parameters is identified for the initial model. This is illustrated in the flow diagram of Figure 2-2.

If it is verified after experimenting with the resulting design that the specifications have been met, then fabrication is an appropriate recommendation. Otherwise, a new model must be selected and the process

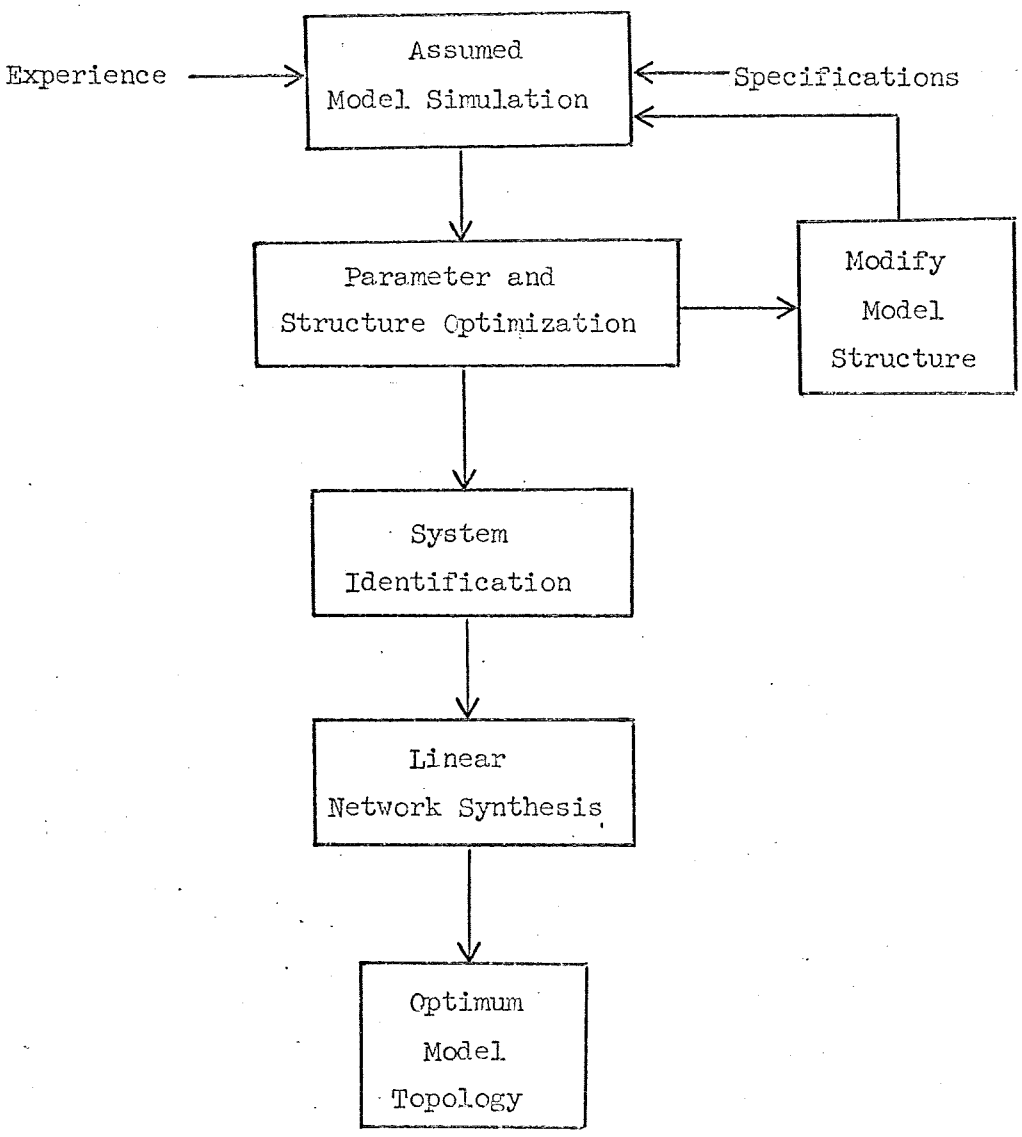


Figure 2-2. Flow diagram for model development using an analog/digital technique

repeated until the specifications have been met. This procedure as applied to network design is summarized in the flow diagram of Figure 2-3. Another technique which is useful in circuit design is the combined analog/digital-NASAP technique for transient analysis. This is illustrated in the flow diagram of Figure 2-4.

2.7 References

1. Bekey, G. A. and W. J. Karplus, Hybrid Computation. New York: John Wiley and Sons, Inc., 1968.

2. Korn, G. A. and T. M. Korn, Electronic Analog and Hybrid Computers. New York: McGraw-Hill Book Company, 1964.

3. Gilbert, E. G., "A Selected Bibliography on Parameter Optimization Methods Suitable for Hybrid Computation," Simulation, vol. 8, pp. 350-352, June 1967.

4. Wilde, D. J. and C. S. Beightler, Foundations of Optimization, Englewood Cliffs, New Jersey: Prentice-Hall, 1967.

5. Herskowitz, G. J., Computer-aided Integrated Circuit Design. New York: McGraw-Hill Book Company, 1968.

6. Angelo, E. J., Jr., J. Logan, and K. W. Sussman, "The separation technique: a method for simulating transistors to aid integrated circuit design," IEEE Transactions on Computers, vol. C-17, no. 2, pp. 113-116, February 1968.

7. Gummel, H. K. and B. T. Murphy, "Circuit analysis by quasi-analog computation," IEEE Proceedings, vol. 55, pp. 1758-1760, October 1967.

8. Balaban P., and J. Logan, "Analog computer simulation of semiconductor circuits," Proc. SJCC, 1968.

9. Gilbert, E. G., "The application of hybrid computers to the iterative solution of optimal control problems," Computing Methods in Optimization Problems. Conference Proceedings, Los Angeles, California, (A.V. Balakrishnan and L. W. Neustadt, eds.) pp. 261-284, New York: Academic Press, 1964.

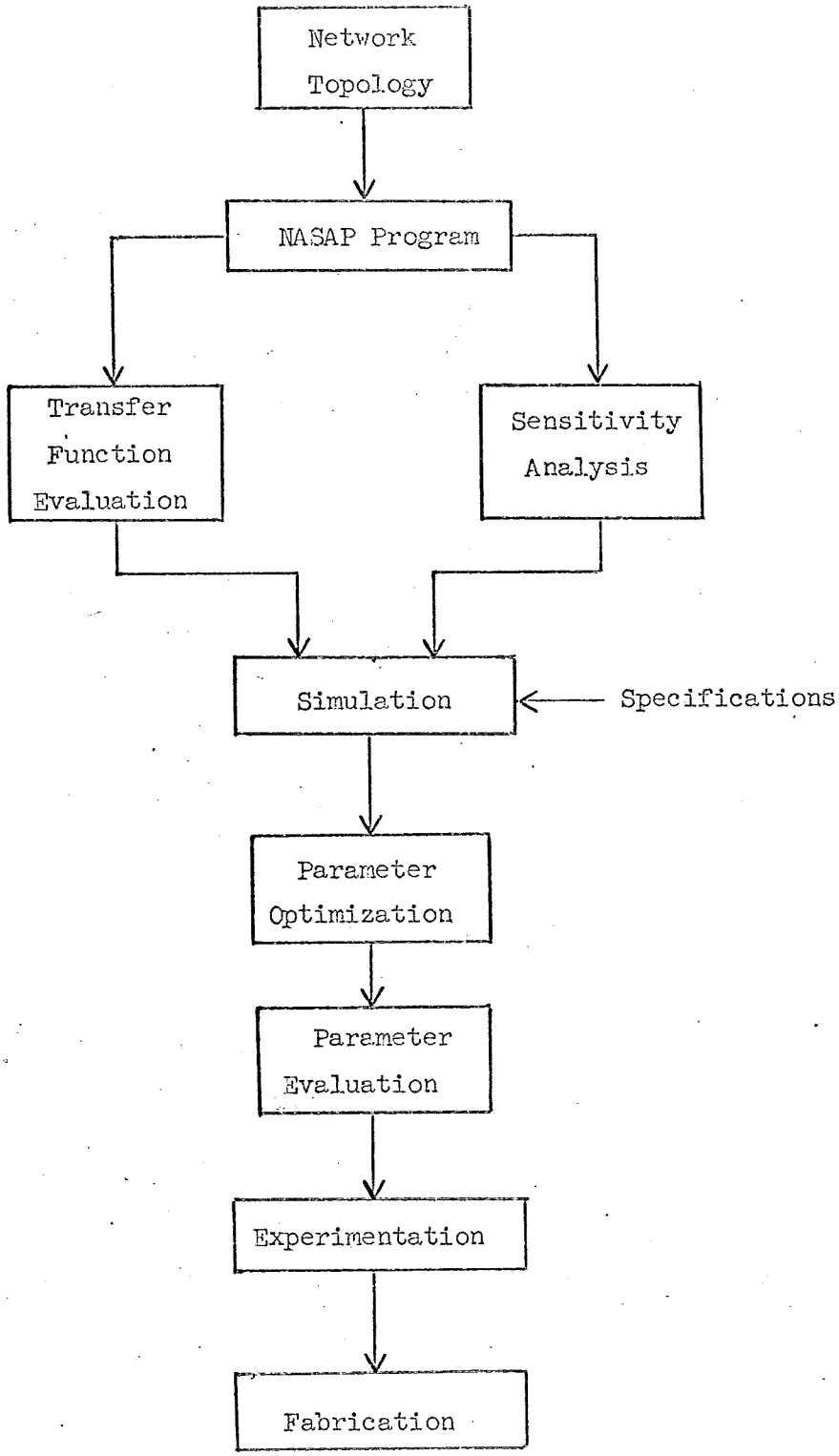


Figure 2-3. Flow diagram for network design using an analog/digital NASAP technique.

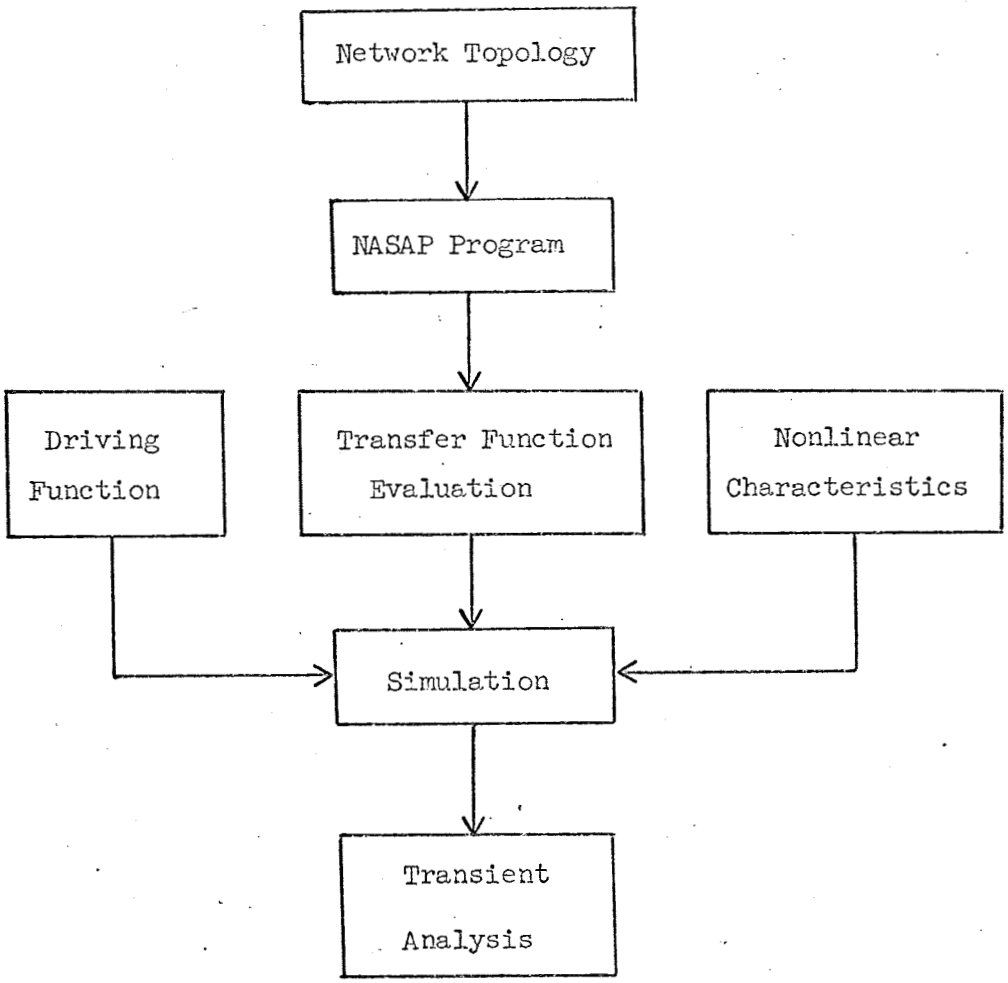


Figure 2-4. Flow diagram for transient analysis using an analog/digital NASAP technique.

Chapter III

RECOMMENDED PRACTICES

3.1 Guidelines for Recommended Analytical Methods

This chapter presents material concerning practices which have been found to be useful during the course of this work.

3.1.1 Indirect Analog Simulation of Linear Circuits

Mathematical models for electronic circuits are based on Kirchhoff's laws which describe the inter-relations between currents and voltages in the circuits. Passive linear circuit elements can be represented as shown in Table 3-1.

The circuit shown in Figure 3-1 can be modeled on the analog computer using the breadboard technique by simulating the following set of equations:

$$i_1 = i_2 + C_3 \frac{dv_3}{dt} \qquad (3\text{-}1)$$

$$V = R_1 i_1 + L_1 \frac{di_1}{dt} + v_3 \qquad (3\text{-}2)$$

$$v_3 = L_2 \frac{di_2}{dt} + R_2 i_2 \qquad (3\text{-}3)$$

The flow graph representation of Equations (3-1), (3-2), and (3-3) is given in Figure 3-2, and the computer diagram for the breadboard simulation of this circuit is given in Figure 3-3. It is seen that the breadboard technique retains circuit topology, and that the individual circuit elements are parameters of the simulation. This technique is satisfactory for the analysis of a circuit of known topology[1,2].

If input-output information is of interest for zero initial conditions, then conventional analog computer programming based on transfer relations

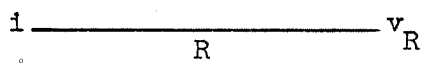
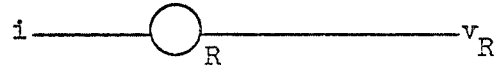
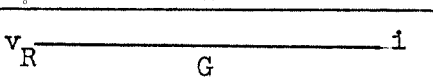
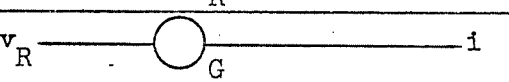
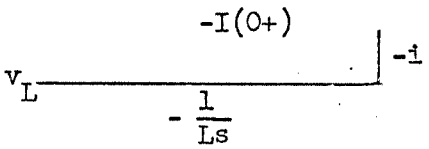
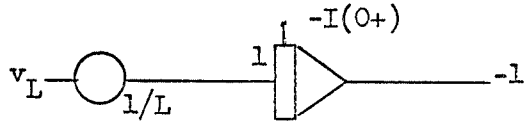
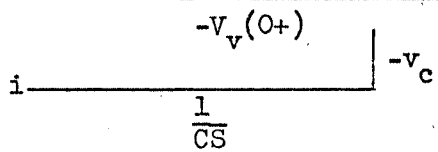
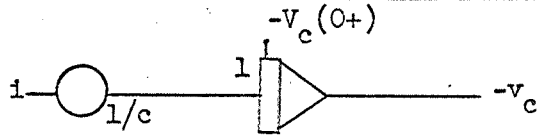
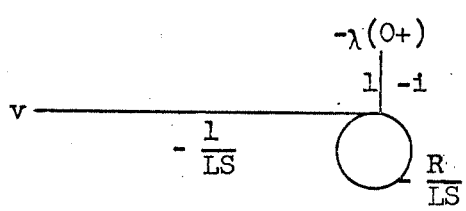
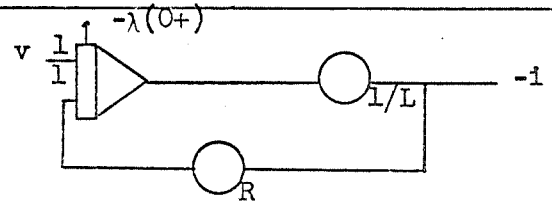
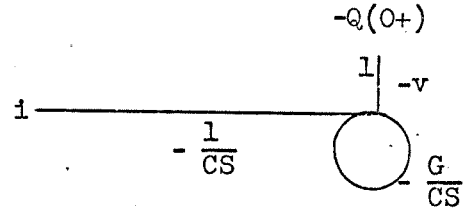
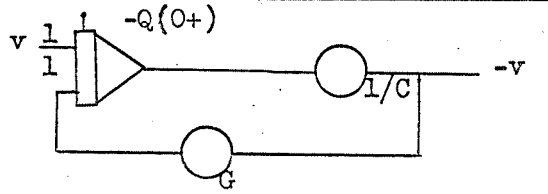
ELEMENT	RELATION	FLOW GRAPH	MODEL
Resistance	$v_R = Ri$		
	$i = Gv_R$		
Inductance	$v_L = L\frac{di}{dt}$		
Capacitance	$i = C\frac{dv_c}{dt}$		
Series RL	$v = L\frac{di}{dt} + Ri$		
Parallel RC	$i=C\frac{dv}{dt} + Gv$		

Table 3-1. Indirect analogs of passive linear circuit elements.

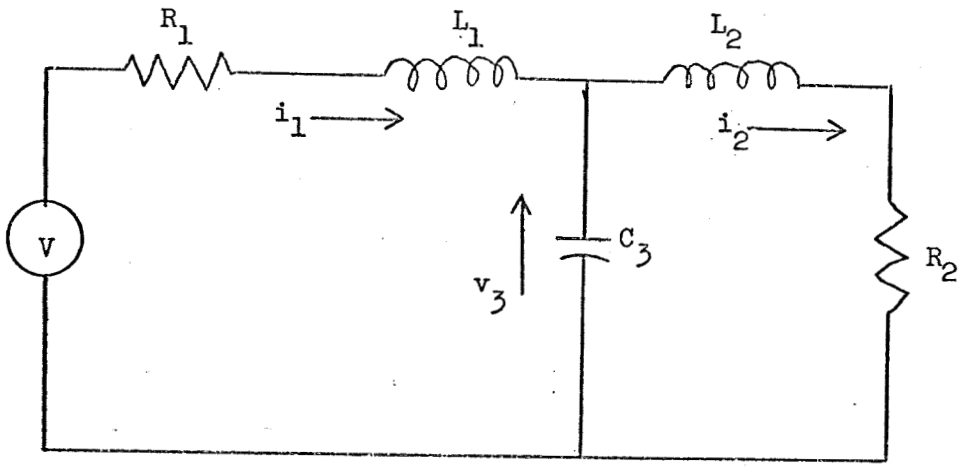


Figure 3-1. Linear third order R-L-C circuit

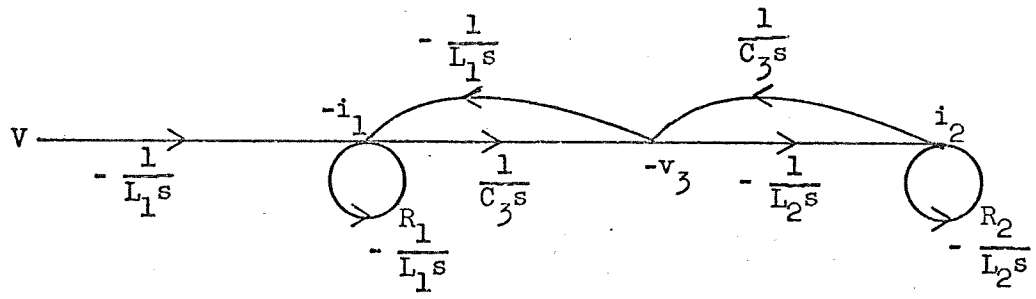


Figure 3-2. Flow graph representation of the circuit in Figure III-1.

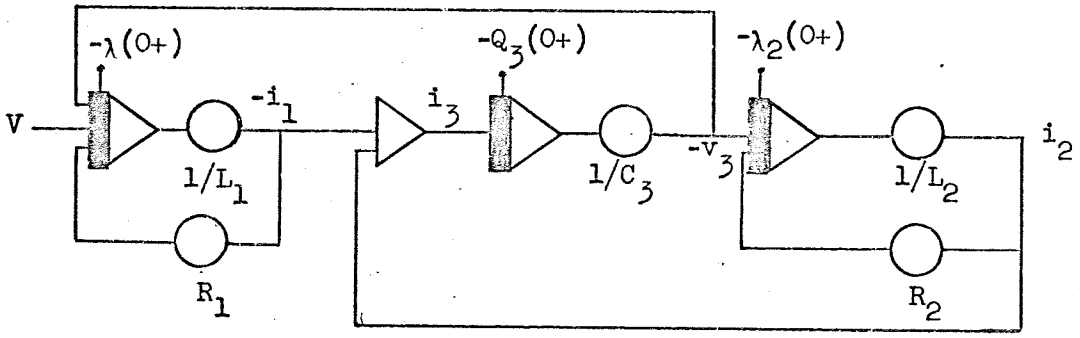


Figure 3-3. Computer diagram for the breadboard simulation of the circuit in Figure III-1.

is preferred. Since the NASAP program can be used to determine the transfer function[3], it is desirable to use this feature of NASAP to obtain the mathematical model. For the circuit of Figure 3-1, the transfer function for the voltage response across the capacitor with respect to an applied driving function is of the form

$$\frac{V_3(s)}{V(s)} = \frac{b_1 s + b_o}{s^3 + a_2 s^2 + a_1 s + a_o} \qquad (3\text{-}4)$$

The flow graph representation of Equations (3-4) is given in Figure 3-4, and the computer diagram obtained by conventional analog computer programming is given in Figure 3-5. An equivalent flow graph to that given in Figure 3-4 for the mathematical model of Equation (3-4) is given in Figure 3-6 and the corresponding analog computer diagram is given in Figure 3-7.

3.1.2 Modeling Nonlinear Semiconductor Devices

Hybrid computers are particularly well suited to the analysis of linear and nonlinear dynamic circuits and systems. When accurate models of active devices are required, considerable advantage can be realized by using actual physical circuit devices as computing elements in the analog computer portion of the hybrid system. For example, a given transistor or an appropriate substitute can be used as a nonlinear analog computing element which represents a dc model of itself in a breadboard or quasi-analog type simulation. Time scaling the ac portion of the Ebers-Moll or charge-control models by a factor k is accomplished by including feedback capacitors in the simulation that are k times the corresponding junction capacitances.

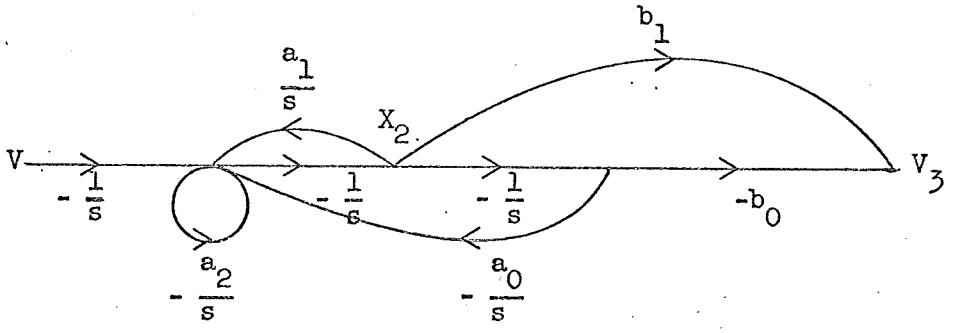


Figure 3-4. Flow graph representation of the mathematical model of Equation (III-4).

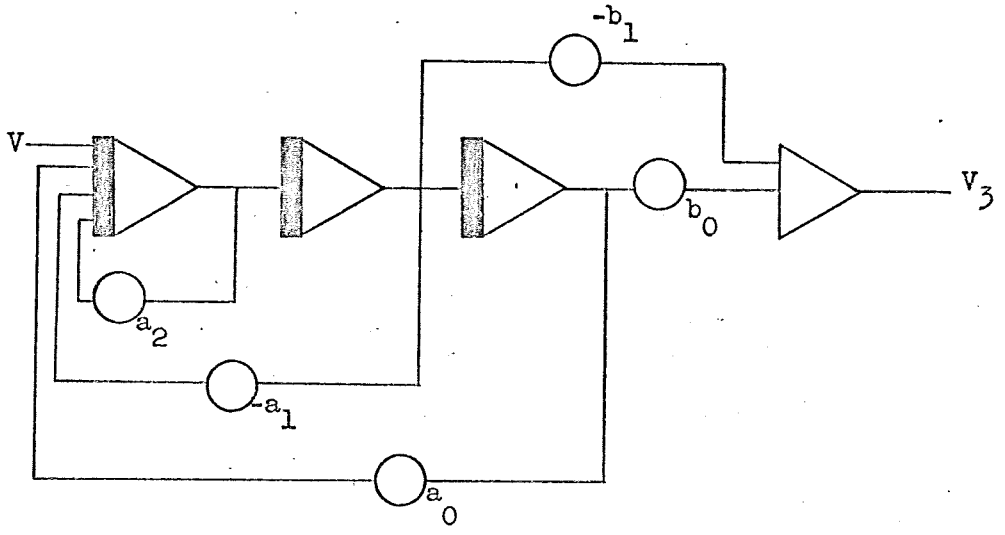


Figure 3-5. Conventional analog computer diagram for the mathematical model of Equation (III-4).

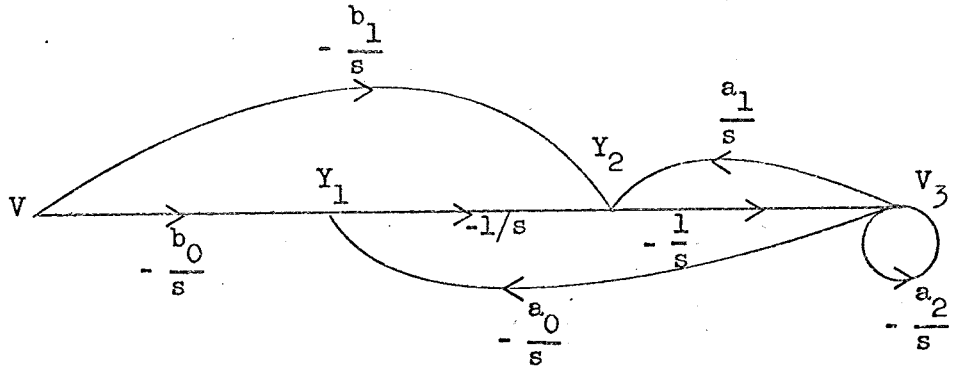


Figure 3-6. Equivalent flow graph representation of the mathematical model of Equation (III-4).

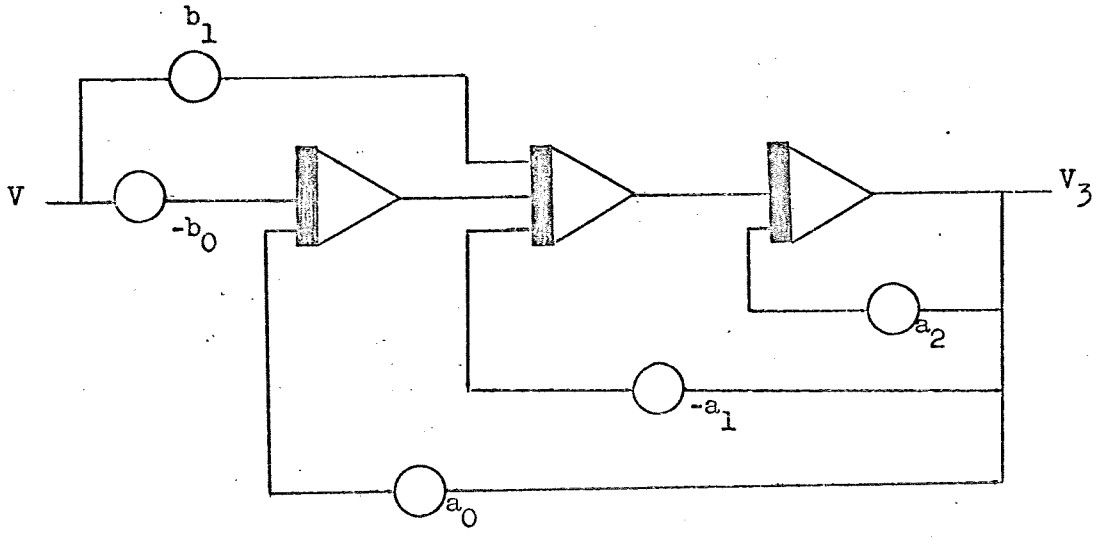


Figure 3-7. Equivalent analog computer diagram for the mathematical model of Equation (III-4).

This procedure named the separation technique by Angelo, Logan, and Sussman (1968)[4], is based on the work of Gummel and Murphy (1967)[5]. With this technique it is convenient to vary circuit and device parameters[6], perform sensitivity analysis, and obtain optimum circuit designs based on dynamic specifications. Total execution times of the order of milliseconds for a complete dynamic analysis are possible regardless of the circuit complexity since all analog computing elements operate simultaneously or in parallel. Programming time is reduced over that required for either analog or digital computer-aided circuit analysis.

The instantaneous base current in a transistor derived for the extended Ebers-Moll or the charge-control model is

$$i_b = -\frac{q_f}{\tau_{bf}} + \frac{q_r}{\tau_{br}} + \frac{d}{dt}(q_f + q_r) + C_{je}\frac{dv_{ej}}{dt} + C_{jc}\frac{dv_{cj}}{dt} \qquad (3\text{-}5)$$

where

q_j is the forward component of charge stored in the base.

q_r is the reverse component of charge stored in the base.

τ_{bf} is the effective base recombination lifetime for forward injection.

τ_{br} is the effective base recombination lifetime for reverse injection.

C_{je} is the emitter junction transition region capacitance.

C_{jc} is the collector junction transition region capacitance.

v_{ej} is the emitter junction voltage.

v_{cj} is the collector junction voltage.

A suitable transistor can be used as an analog computing element which generates the portion of the nonlinear low frequency model represented by the first two terms in Equation (3-5). Based on the gross assumption that all lifetimes are equal, the low-frequency component of

the base current is

$$i_b' = -\frac{1}{\tau}(q_f + q_r). \tag{3-6}$$

This assumption of equal lifetimes is satisfactory unless the collector junction is forward biased. If this is the case, then the conventional analog programming technique should be used. The instantaneous base current can also be expressed as

$$i_b = i_b' + \tau \frac{di_b'}{dt} - C_{je} \frac{dv_{ej}}{dt} - C_{jc} \frac{dv_{cj}}{dt} \tag{3-7}$$

If all expressions are time scaled according to the relationship $T = kt$, then there results

$$i_b = i_b' + k\tau \frac{di_b'}{dT} - kC_{je} \frac{dv_{ej}}{dT} - kC_{jc} \frac{dv_{cj}}{dT} \tag{3-8}$$

The breadboard representation for a time scaled simulation of a transistor is shown in Figure 3-8. Assuming that the voltage drop across the sensing resistor r is small compared with the voltage drops across the transition region capacitances, the current through these capacitances is

$$i_k = -kC_{je} \frac{dv_{ej}}{dT} - kC_{jc} \frac{dv_{cj}}{dT} \tag{3-9}$$

The current through capacitance C is

$$i_s = CAr \frac{di_b'}{dT} \tag{3-10}$$

Summing the currents in Equations (3-6), (3-9), and (3-10) gives

$$i_b = i_b' + CAr \frac{di_b'}{dT} - kC_{je} \frac{dv_{ej}}{dT} - kC_{jc} \frac{dv_{cj}}{dT} \tag{3-11}$$

Equation (3-11) is equivalent to Equation (3-8) if

$$k\tau = CAr \tag{3-12}$$

Hence, the transistor model shown in Figure 3-8 is time scaled by a factor $\frac{CAr}{\tau}$, and the effective time-scaled life time is $k\tau$.

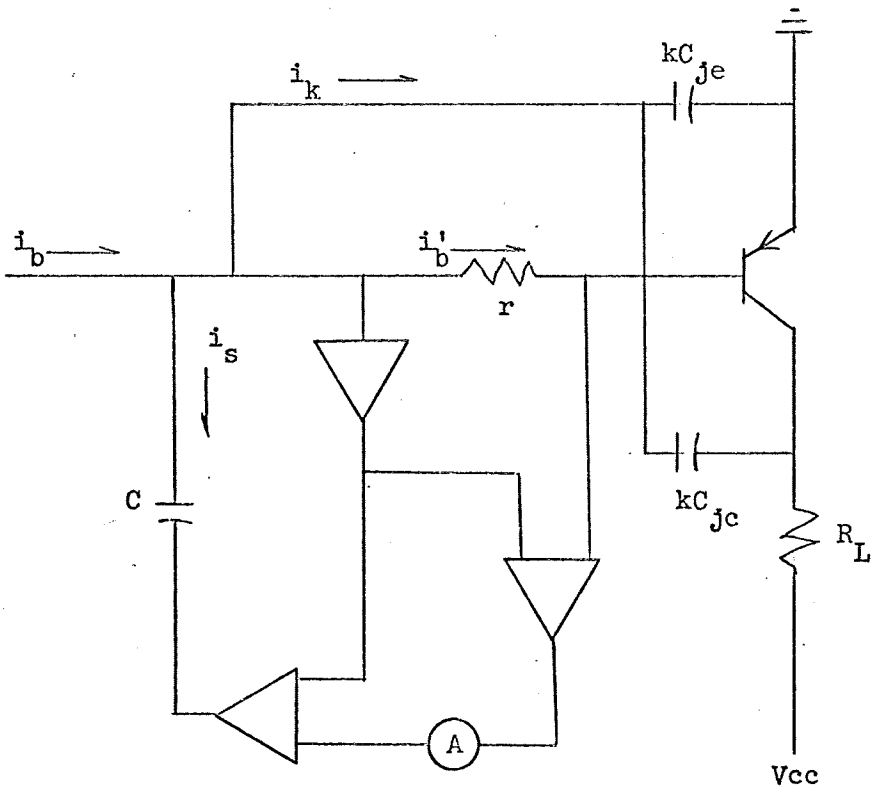


Figure 3-8. Breadboard separation model for transistor simulation.

Passive circuit components can also be represented by standard analog computing elements. Consider the RC load illustrated in Figure 3-9. This circuit can be simulated for dynamic computation as indicated in Figure 3-10.

The conventional analog computer method of modeling a transistor is based on the simulation of the extended Ebers-Moll model illustrated in Figure 3-11. In this case time-scaled diodes are used to simulate the junction nonlinearities. The emitter and collector currents are given by

$$i_e = q_f \left(\frac{1}{\tau_f} + \frac{1}{\tau_{bf}}\right) - \frac{q_r}{\tau_r} + \frac{dq_f}{dt} + C_{je} \frac{dv_{ej}}{dt} \qquad (3\text{-}13)$$

and

$$i_c = -\frac{q_f}{\tau_f} + q_r \left(\frac{1}{\tau_r} + \frac{1}{\tau_{br}}\right) + \frac{dq_r}{dt} + C_{jc} \frac{dv_{cj}}{dt} \qquad (3\text{-}14)$$

where

τ_f is the minority carrier excess charge stored in the device.

τ_r is the reverse injection charge control parameter.

The forward conduction current is

$$i_{fr} = \frac{q_f}{\alpha_f \tau_f} \qquad (3\text{-}15)$$

and the reverse current is

$$i_{rr} = \frac{q_r}{\alpha_r \tau_r} \qquad (3\text{-}16)$$

The effective base recombination lieftime for forward injection may be expressed as

$$\tau_{bf} = \tau_f \frac{\alpha_f}{1 - \alpha_f} \qquad (3\text{-}17)$$

where α_f is the forward current gain, and the effective base recombination lifetime for reverse injection may be expressed as

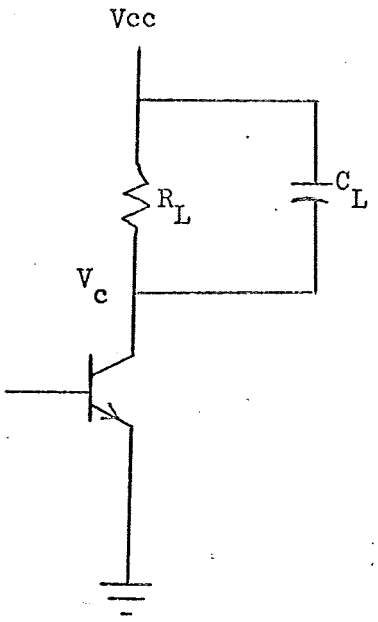


Figure 3-9. RC collector load output stage.

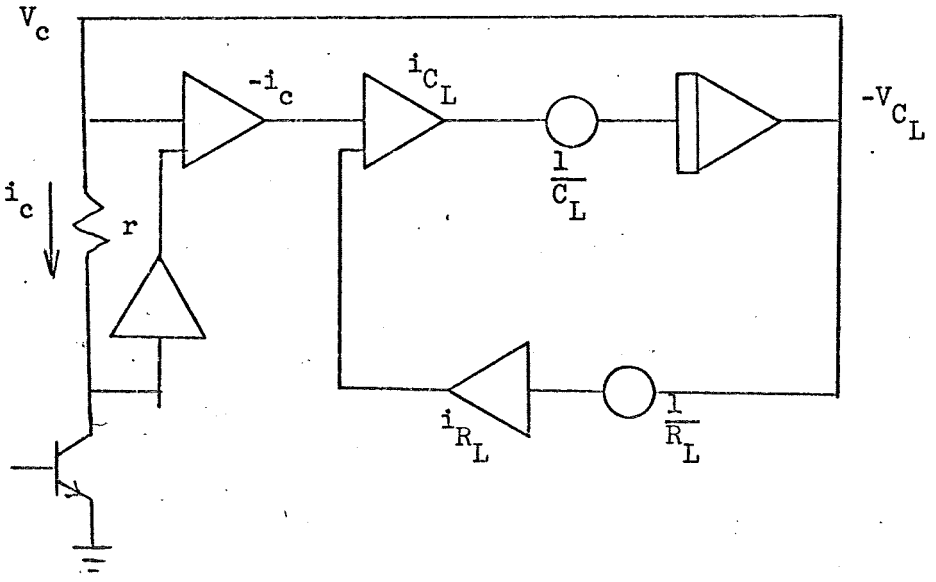


Figure 3-10. Analog computer simulation of RC load.

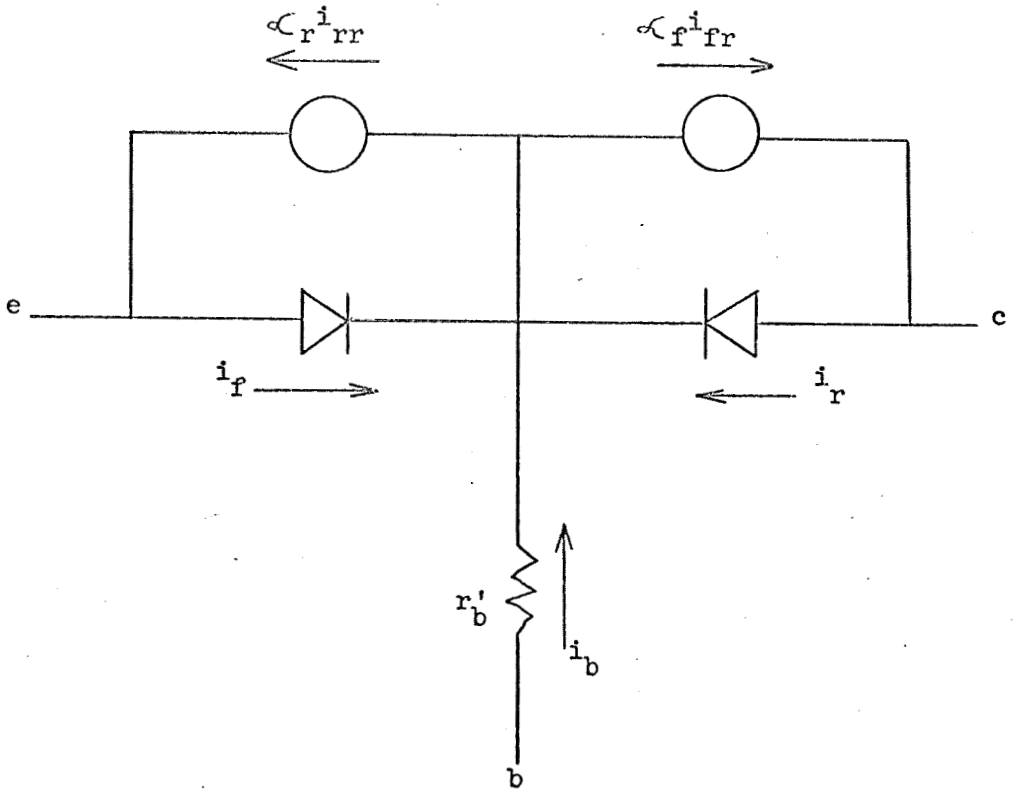


Figure 3-11. Extended Ebers-Moll Transistor model.

$$\tau_{br} = \tau_r \frac{\alpha_r}{1 - \alpha_f} \tag{3-18}$$

where α_r is the reverse current gain.

Substitution of Equations (3-15), (3-16), (3-17), and (3-18) into Equations (3-13) and (3-14) yields

$$i_e = i_{fr} + \alpha_f \tau_f \frac{di_{fr}}{dt} + C_{je} \frac{dv_{ej}}{dt} - \alpha_r i_{rr} \tag{3-19}$$

and

$$i_c = i_{rr} + \alpha_r \tau_r \frac{di_{rr}}{dt} + C_{jc} \frac{dv_{cj}}{dt} - \alpha_f i_{fr} \tag{3-20}$$

The diodes can be modeled by simulating the equation for instantaneous diode current. Charge control theory provides the relationship

$$i_d = \frac{q}{\tau_f} + \frac{dq}{dt} + C_j \frac{dv_d}{dt} \tag{3-21}$$

where

i_d is the instantaneous diode current

q is the minority carrier excess charge stored in the device

τ_f is the minority carrier lifetime

C_j is the junction transition-region capacitance

v_d is the voltage across the junction

The low frequency diode current is

$$i_R = \frac{q}{\tau_f} \tag{3-22}$$

Substitution of this relationship in Equation (3-21) yields

$$i_d = i_r + \tau_f \frac{di_R}{dt} + C_j \frac{dv_d}{dt} \tag{3-23}$$

If this expression is time scaled according to the relationship $T = kt$, then Equation (3-23) becomes

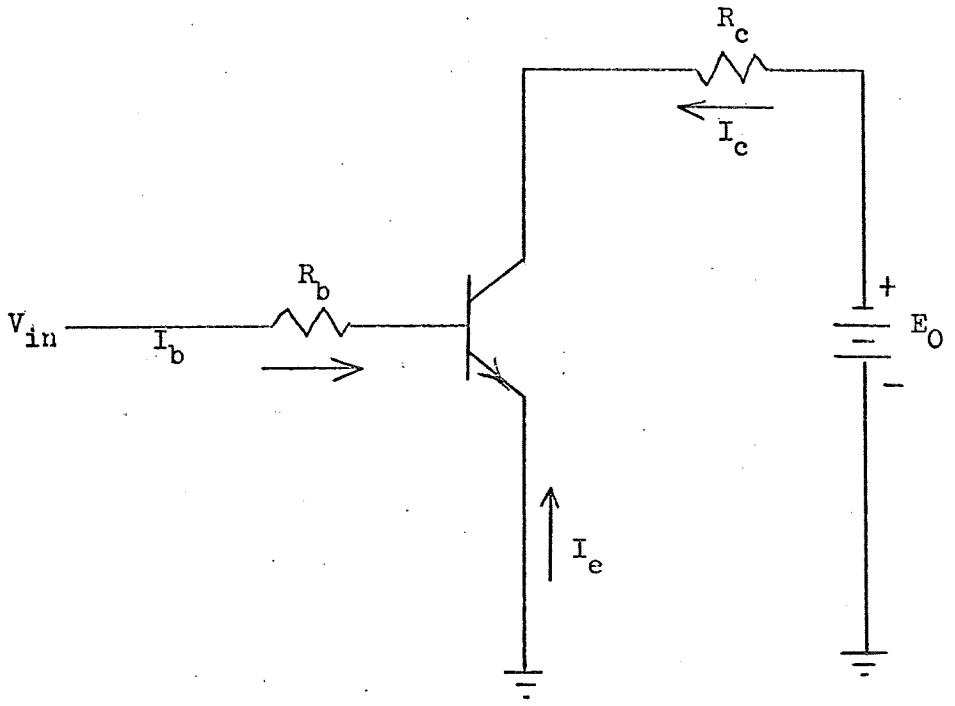


Figure 3-12. Circuit of a grounded emitter transistor amplifier.

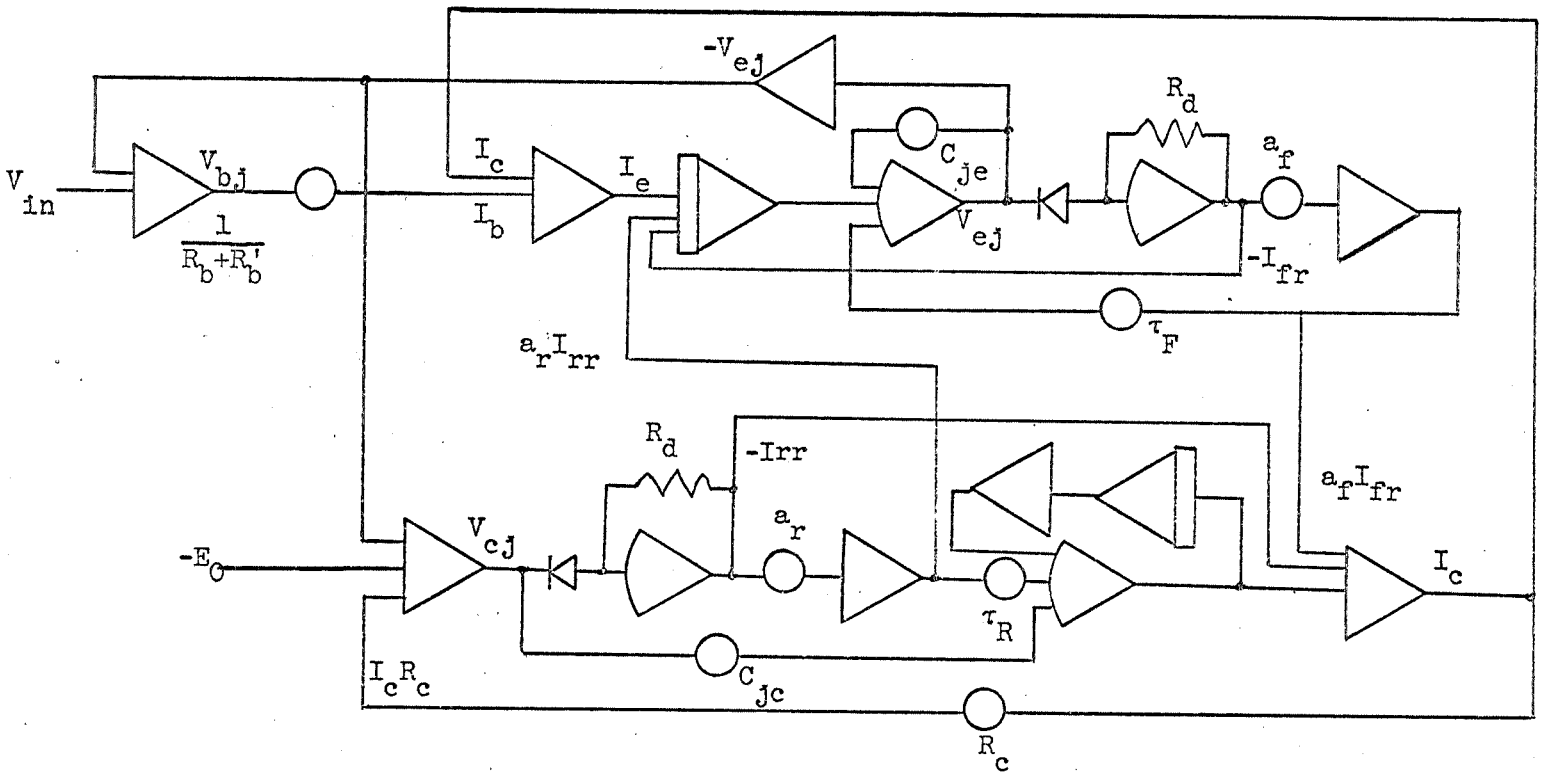


Figure 3-13. Conventional analog computer diagram for a grounded emitter transistor amplifier.

$$i_d = i_R + k\tau_f \frac{di_R}{dT} + kC_j \frac{dv_d}{dT} \qquad (3\text{-}24)$$

The Laplace transform of Equation (3-24) is

$$I_d(s) = I_R(s) + sk\,\tau_f I_r(s) + s\,k\,C_j\,V_d(s) \qquad (3\text{-}25)$$

or

$$V_d(s) = \frac{1}{kC_j}\,\frac{I_d(s) - I_r(s)}{s} - k\tau_f I_R(s) \qquad (3\text{-}26)$$

As an example of the use of the separation principle, the grounded emitter transistor amplifier shown schematically in Figure 3-12 will be modeled using conventional analog computer programming of Equations (3-19), (3-20), (3-25), and (3-26). The resulting analog computer diagram is shown in Figure 3-13.

It has been shown that the breadboard method of simulation retains actual circuit topology. This is made possible by using sensing resistors to detect junction currents. The low frequency component of the junction current is used as a measure of the charge stored in the junction. Effects such as base widening and various interdependencies are provided without any programming required.

In the conventional analog method of simulation, diodes are used to provide the junction nonlinearities in transistor models. Simulation based on the Ebers-Moll transistor model includes two-simulated interacting diodes which permit adjustment of individual device parameters such as nonlinear current gain and recombination lifetimes.

In a hybrid simulation, nonlinear functions such as nonlinear current gains can be provided by digital computer function generation. Base resistance and other parameter changes can be controlled by the digital computer program. This permits the designer to consider devices which have arbitrary characteristics. If a complex network has sections which

have similar topology, then the topological structure and the parameter values of the general simulation can both be modified under digital computer control as required. The conventional analog method is well suited to hybrid computer implementation of dynamic optimization techniques and sensitivity analysis since solution times of the order of milliseconds are typical.

3.1.3 Nonlinear Function Generation

Nonlinear functions can be generated on an analog computer using diode function generators. Their main disadvantage is the set up time. If the function requires a large number of segments, then the cost becomes excessive. These special purpose devices cannot be used for other purposes.

The hybrid computer can provide nonlinear function generation in a more efficient manner. The instantaneous value of the independent circuit quantity represented on the analog computer is digitized, the digital computer is programmed to perform a table-lookup, and the resulting value is transformed into an analog voltage level and transmitted to the analog computer.

Individual breakpoints are supplied by the circuit designer as input data to the digital computer. The digital computer is programmed to perform linear interpolation between these breakpoints. The buffered output is a zero order hold approximation of the desired output function. This staircase function can be smoothed by passing it through an analog computer representation of a low-pass filter with a transfer function of the form

$$\frac{E_o(s)}{E_i(s)} = \frac{K}{s + \omega_o} \qquad (3\text{-}27)$$

The smoothing is improved for a lower cut-off frequency, ω_o, but the resulting phase shift or time delay is increased. A predictor network can be included to compensate for this time delay.

3.1.4 Direct Design of Linear Dynamic Circuits

Exploitation of computer-aided design techniques generally starts with an assumed mathematical model of the system or circuit to be designed. It is believed that the availability of accurate device models is crucial to the justifiable use of the computer for circuit design. It is also desirable to obtain the simplest possible device models of sufficient accuracy since higher-order models limit the size network that can be treated using either analog or digital computers. Since execution time for digital computer programs increases for higher order models, the expense of computer-aided design may become economically prohibitive for certain types of calculations such as those which involve optimization techniques.

For device specifications in the time domain, analog computers have been used effectively to identify the parameters of system functions of assumed form which resulted in the best match between the device response and the model response to a specified input function[7-9]. More recently the digital computer has been used to implement automated parameter identification for electronic circuits[10]. Hybrid computer systems have been used very little for electronic circuit design[1,2].

For many purposes the internal processes of the device are not of specific interest, but rather the influence on external system performance[11]. In these cases, the model only needs to reproduce the desired terminal characteristics[12-17]. Emphasis will then be placed on obtaining the

simplest model which meets the required specifications subject to certain constraints such as physical realizability and the range of allowable parameter values[18]. Attention is concentrated on linear circuits described by pre-determined mathematical models for which the best values of the parameters are to be selected[19]. An extension to the problem is to optimize the structure of the mathematical model as well as the parameters of the given model[20]. This model structure is represented by an analogous interconnection of physical devices available as computing elements on the analog computer.

It appears that most effort in the area of computer-aided circuit design has been devoted to techniques for network analysis and that relatively few techniques are available for direct circuit design. If the circuit designer constructs an algorithm which accomplishes the required decision making process involved in step-by-step repeated analysis and parameter adjustment, then an automated direct design is possible. This permits designers to be freed from much that is routine so that their experienced engineering judgment can be directed to efficient effort such as evaluation of the final design.

With this direct design technique, a physically realizable model constructed of analog computer elements is immediately available. Whether a transform approach or a state variable approach is chosen, the model of the dynamic circuit programmed on the analog computer can take many forms including a breadboard arrangement, a cascade arrangement, a parallel arrangement, or a feedback arrangement. The analog computer model is complete when all gain elements have been specified. The integrators provide the dynamics associated with the transient response.

In obtaining the best values for the model parameters, it is necessary

to establish a comparison criterion or index of performance. An obvious choice is an index based on the difference (error) between the actual or desired transient response and the transient response of the model to the specified input. The integral of the squared error (ISE) has been used extensively in this project as well as the integral of the absolute error (IAE). Other criteria can also be used with equal ease since an analytical solution is not required[21]. The chosen index of performance can be computed on the analog computer and returned to the digital portion of the system, or it can be computed on the digital computer. The optimization algorithm operates upon this quantity.

There are numerous optimization techniques which have been well covered in recent literature[22,23]. The techniques employed in this project have been simple ones based on sequential univariate searches. The relaxation method requires that only one parameter be changed at a time and a minimum of the index of performance is found by varying this single parameter with all other parameters held fixed. Each parameter is varied in turn and then the whole process is iterated until no additional improvement is possible.

This simple technique is not known for its mathematical elegance, however, it does have the advantages that the logic required is relatively simple to program, no partial derivatives need to be taken, and it works on simple problems. A univariate search appropriately represents the automation of the human decision process. It has proven to be effective for showing clearly the desirable capabilities of direct design in numerous cases. Also, because of the simplicity of this optimization technique and the low cost operation of the hybrid system, the execution can easily be verified by stepping through the program in a manner that

permits the designer to mentally check the logical decisions implemented by the optimization algorithm. It has the disadvantage, as do other methods, that it does not succeed in the location of the optimum conditions for certain problems.

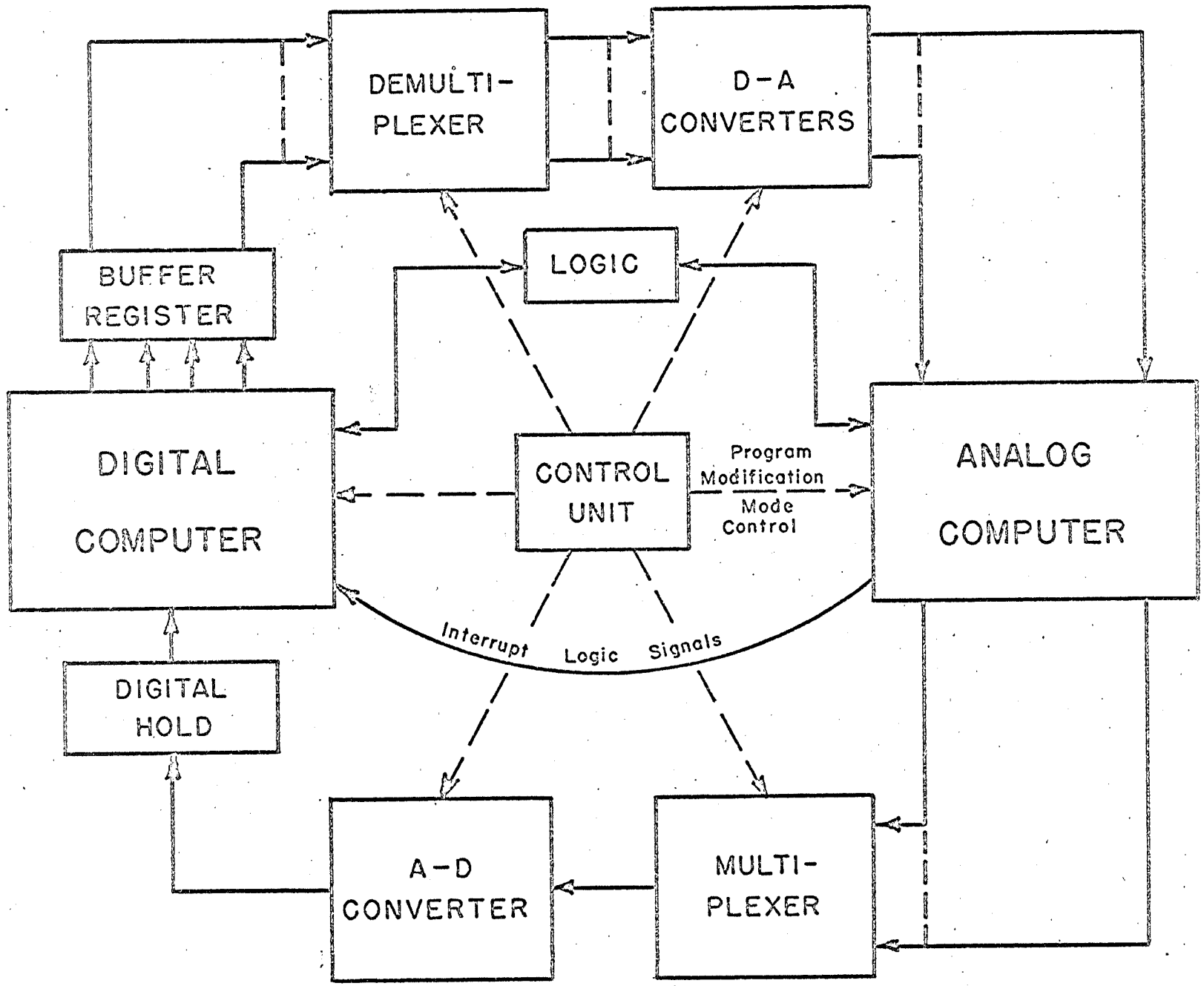
An alternate method employing univariate search has been used. This involves the determination of the local gradient of the index of performance with respect to the parameters at a test point and then proceedings with a univariate search along the local gradient to find a minimum. The process is then repeated.

3.2 Hardware Design, Maintenance, and Diagnostics

The hybrid computer system developed at Tulane University can be represented as shown in Figure 3-14. The digital computer is a Univac AN/GSK-1. This machine has an 8K magnetic drum memory and a 256-word 24-bit scratch pad core memory. The basic instruction time is 40 μsec. Program input is by either of two paper tape readers or by manual console entry. Data input is through the eight constant registers to the accumulator or through a buffered data register directly to core memory. Four output registers and a printer provide for the necessary display and data transfer.

The analog section is composed of four EAI TR-48, two EAI 16-31R, and one EAI 231R analog computer. These computers have been modified to provide for digital mode and time scale control, automated patching, digital parameter units, and high-speed overload detectors. The conversion and control linkage system includes several multiplexed analog-to-digital converters and several digital-to-analog converters.

FIGURE 3-14. TULANE HYBRID COMPUTER SYSTEM



Modern commercial hybrid computer systems which are currently available represent a wide selection. There are more than ten manufacturers of analog computers for hybrid computer use, and the number of digital computer manufacturers is even larger. Since each manufacturer has several different models available, there are a large number of hybrid configurations available, at widely differing costs.

With most analog computer programming accomplished by patchboard connections, it is desirable to have a convenient patchboard layout. If hybrid computation is to be efficient, then the monitoring of all analog, logic, and linkage elements by the digital computer is necessary. Suitable hardware and software must be provided or developed for diagnostic and program check-out purposes. Perhaps the most desirable feature of all is the requirement of reliability. It will also be essential that provision be made to expand analog, logic, and linkage elements, by field installation with minimal effort. For high-speed iterative operation to be feasible, it will be essential that reset and hold times be minimal and that amplifier bandwidths be in excess of 100KHz.

In the case of the digital portion of the hybrid system, the cycle time is the most significant requirement. This is true since the speed of the digital computer is the most serious limitation of this portion of the system. It will be important to have a cycle time of 1μsec. or less. A 16-K memory of 24-bits or more is desirable. A complete software documentation including standard FORTRAN IV, hybrid linkage control subroutines, and system diagnostic subroutines will be important.

If funds are available, it will be appropriate to select an experienced manufacturer who can provide complete systems responsibility and maintenance service. It was decided at Tulane University to develop

in-house capability in these areas.

3.3 System Management

In the educational environment, an interactive computer-aided design program is desirable. However, a direct design program is usually more efficient cost-wise in the industrial environment. This is partially due to the efficiency of operating a closed-shop digital facility. The desirable features of a large, complex digital facility would be especially convenient for hybrid computation, hence a modified open-shop basis of operation is recommended. The associated programming and operating assistance should prove beneficial along with the participation of the problem originator who is familiar with the basic design concepts. This philosophy may preclude the availability of a large-scale digital computer for hybrid computation. The need for real-time access to the digital computer will also eliminate prime shift use of many systems. These considerations suggest the need for a committed medium-scale digital computer with off-line or non-real-time coupling to the large-scale machine. The background of the usual digital computer center staff also tends to limit the usefulness of a large-scale machine for hybrid computation unless the machine is committed nearly full time to hybrid problems.

It can be reasoned that as the speed and cost of digital computers improve then the digital computer can eliminate the need for hybrid computation. It results, however, that as digital devices become available which are faster and less expensive then the hybrid interface also becomes faster and less expensive. This coupled with the high speed parallel integration capability of the analog computer gives ample justification for anticipation of a useful future for hybrid computation.

3.4 Data Display

Analog computation is involved basically with time domain information in continuous form. Since total solution times are commonly of the order of a few milliseconds, an oscilloscope is often used to display the continuous dynamic output response for photographic recording. Direct visual observation is possible with a storage oscilloscope, or iterative solutions can be executed to provide for viewing on a non-storage oscilloscope.

After iterative direct design computation, the values of computed optimum parameter values are listed on the output printer. Scaled values of the optimum performance indices and information concerning optimum network topology are also listed.

3.5 References

1. Bohling, D. and J. Chernak, "A hybrid computer technique for optimization," Simulation, vol. 5, no. 4, pp. 227-231, October 1965.

2. Chernak, J. "Hybrid computer techniques and applications," System Analysis by Digital Computer. (F.F. Kuo, and J. F. Kaiser, eds.) pp. 315-348, New York: John Wiley and Sons, Inc., 1966.

3. Happ, W. W., "Flowgraph Techniques for Closed Systems," IEEE Transactions on Aerospace and Electronic Systems, vol. AES-2, no. 3, pp. 252-264, May 1966.

4. Angelo, E. J., Jr., J. Logan, and K. W. Sussman, "The separation technique: a method for simulating transistors to aid integrated circuit design," IEEE Transactions on Computers, vol. C-17, no. 2, pp. 113-116, February 1968.

5. Gummel, H. K. and B. T. Murphy, "Circuit analysis by quasi-analog computation," IEEE Proceedings, vol. 55, pp. 1758-1760, October 1967.

6. Balaban, P., and J. Logan, "Analog computer simulation of semiconductor circuits," Proc. SJCC, 1968.

7. Clymer, A. Ben, "Direct system synthesis by means of computers," AIEE Transactions (Communications and Electronics), pp. 798-806, January 1959.

8. Harvey, R. A., et al., "Analog computer methods for parameter optimization," Simulation, vol. 6, pp. 181-191, March 1966.

9. Thomas, L. C., "An application of the analog computer to electronic circuit simulation," IEEE Transactions on Electronic Computers, vol. EC-16, no. 4, pp. 515-516, August 1967.

10. Herskowitz, G. J., Computer-aided Integrated Circuit Design. New York: McGraw-Hill Book Company, 1968.

11. Zadeh, L. A., "On the Identification Problem," IRE Transactions On Circuit Theory, vol. CT-3, pp. 277-281, December 1956.

12. Bekey, G. A., "Optimization of multiparameter systems by hybrid computer techniques, Part I and II," Simulation, vol. 2, no. 3 and 4, February, March, 1964.

13. Bekey, G. A. and R. B. McGhee, "Gradient methods for optimization of dynamic system parameters by hybrid computation," Proceedings Conference on Computing Methods in Optimization Problems, (Eds. A. Balakrishnan and L. Neustadt,) Academic Press; 1964.

14. Bekey, G. A. and H. F. Meissinger, "An Analysis of Continuous Parameter Identification Methods," Simulation, vol. 6, pp. 94-102, Feb. 1966.

15. Bekey, G. A. et al, "Parameter optimization by random search using hybrid computer techniques," AFIPS Proc. FJCC, vol. 29, pp. 191-200, 1966.

16. Meissinger, H. F., "Parameter optimization by an automatic open loop computing method," 4th Inters. Conf. on Analog Computation AICA, Brighton, England, September 1964.

17. Melanson, F., and P. Lanner, "Parameter optimization by hybrid techniques," Instruments and Control Systems, vol. 37, no. 1, pp. 125-128, Jan. 1964.

18. Beck, C. H., and R. L. Drake, "A Hybrid-NASAP Module for Direct Design of Linear Dynamic Circuits," IEEE Transactions on Education, vol. E-12, no. 3, pp. 31-35, September 1969.

19. Beck, C. H., R. L. Drake, and M. H. Kuo, "A NASAP module for direct design of linear dynamic circuits using a hybrid computer optimization technique," Conference Record, Twelfth Midwest Symposium on Circuit Theory, pp. VIII2.1-VIII2.9, Austin, Texas, April 1969.

20. Beck, C. H., R. L. Drake, and M. H. Kuo, "System Modeling and Configuration Optimization Using Hybrid Computer Techniques," Cornell Conference on Computerized Electronics, Ithaca, N. Y. Aug. 26-28, 1969.

21. Gilbert, E. G., "The application of hybrid computers to the iterative solution of optimal control problems," Computing Methods in Optimization Problems. Conference Proceedings, Los Angeles, California, (A.V. Balakrishnan and L. W. Neustadt, eds.) pp. 261-284, New York: Academic Press, 1964.

22. A. V. Balakrishnan and L. W. Neustadt, eds., Computing Methods in Optimization Problems. New York: Academic Press, 1964.

23. Gilbert, E. G., "A selected bibliography on parameter optimization methods suitable for hybrid computation," Simulation, vol. 8, pp. 350-352, June 1967.

Chapter IV

REPRESENTATIVE DESIGNS

This chapter is intended for review and reference. The various hybrid methods which have been found to be appropriate for network design are presented. The significant capabilities and limitations of these methods are discussed from the point of view of the user.

4.1 Representative Methods

The most appropriate methods to be considered in connection with hybrid computation will make use of one or more of the following strong points of either analog or digital computers.

1. High speed parallel operation of all analog computational elements including multiplication, integration, and nonlinear function generation.
2. Facility for including actual physical devices from the network under study in the computer simulation.
3. Ability to "trade off" digital solution time and accuracy.
4. Facility for performing logical operations and providing unlimited time delay or memory.
5. Ability to automatically modify the digital program on the basis of calculations.

The high speed, parallel operation of the analog computer will be desirable primarily for analyzing or simulating the operation of networks with dynamic and/or nonlinear characteristics. The analog computer also makes it possible to include actual physical devices such as diodes and

transistors in the simulation. This removes some of the restrictions on the size network that can be handled. When accuracies of more than 0.1% are required, it is necessary to trade increased off-line digital solution time for accuracy. Iterative computation and logical operations as required in optimization calculations are possible through the linkage of digital and analog computers. Automatic modification of the digital program on the basis of calculations can be supplemented by digital control of the analog program to effect automatic structure modification.

4.1.1 Direct Design Example

As a simple specific example of the direct design method described in Section 3.1, a single-stage attenuated first-order low-pass filter will be designed. The transfer function of the filter is

$$\frac{T_m(s)}{x(s)} = \frac{K}{s + a} \qquad (4\text{-}1)$$

It is desired to find the values of K and a_1 that cause the model output $T_m(t)$ to give "best" agreement with the actual output or desired output, T(t), in response to a step function input x(t). In terms of state variables the network equations consist of a single state equation

$$\dot{q}_1(t) = - a\, q_1(t) + K\, x(t) \qquad (4\text{-}2)$$

and a single output equation given by

$$T_m(t) = q_1(t) \qquad (4\text{-}3)$$

A diagram giving the analog model of the low-pass filter is shown in Figure 4-1. There it is seen that the digital computer under program control inserts parameters K and a into the analog model through the digital-to-analog converters. The index of performance is generated in

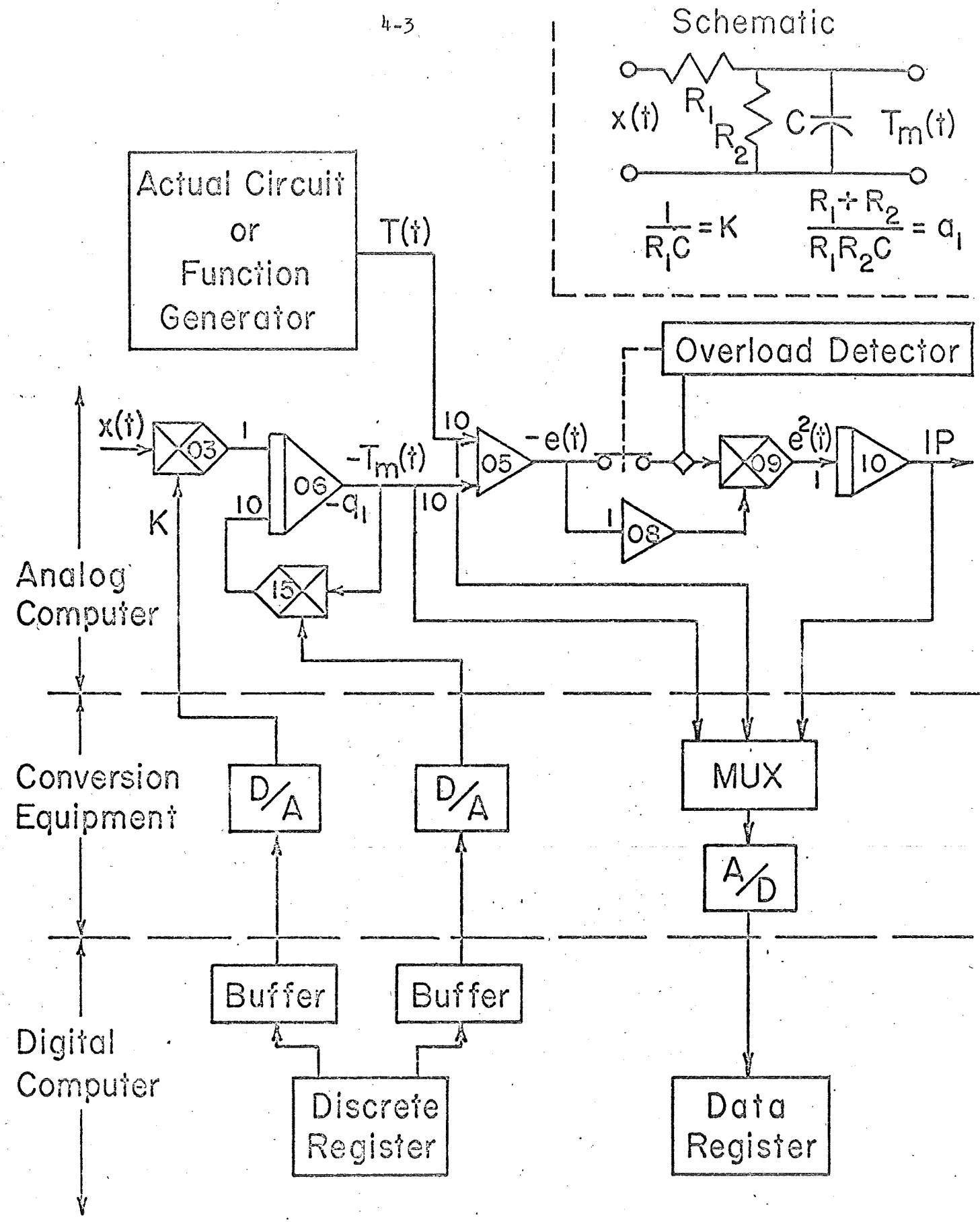


Figure 4-1. Hybrid Computer Block Diagram and Circuit Schematic

the analog computer and returned to the digital computer through the analog-to-digital converters.

Since individual hybrid systems vary considerably and since a given program which is not compiler oriented is completely machine dependent, only the general flow chart of the digital computer program is shown in Figure 4-2. When automatic patching and universal hybrid software become commonly available as is the case for digital computers, then completely documented programs can be shared. General flow charts are usually sufficient, however, since the dynamic circuit simulation is accomplished in a simple and straightforward manner on the analog computer. The digital computer program includes the logic for changing the parameters supplied to the analog model, instructions to cause the parameters to be converted to analog quantities, instructions for reading the performance index, and stopping criteria to determine when the "best" model has been achieved. The ratio of K to a_1 gives the dc gain, and a_1 represents the cut-off frequency. Knowing these quantities, passive circuit element values can be computed, or the realization obtained with the analog computer operational amplifiers can be used directly.

A typical step response is illustrated by the criterion function in Figure 4-3. This function was obtained by setting the gain constant of a first-order step response function generator to a value of 2.512 and the cut-off frequency to a value of 1.646. The initial model response is also illustrated in Figure 4-3. Model parameters are perturbed according to the hybrid algorithm and a check is made for improvement in the index of performance. This process is repeated until the tolerance criterion is met. It is seen in Figure 4-3 that the specified parameter tolerance criterion was met for a performance index of 254.

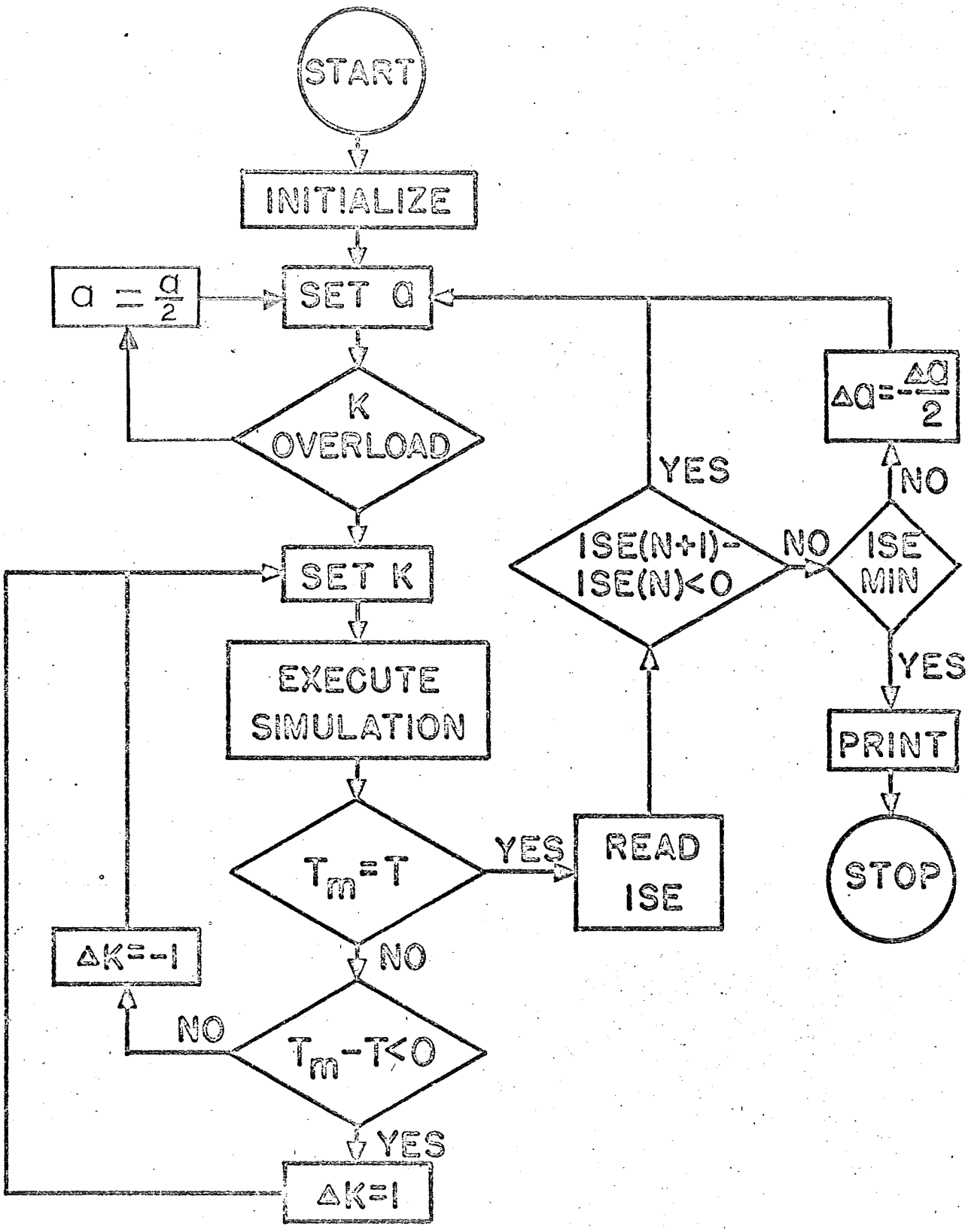


Fig. 4-2. Flow Chart for Illustrative Design Example

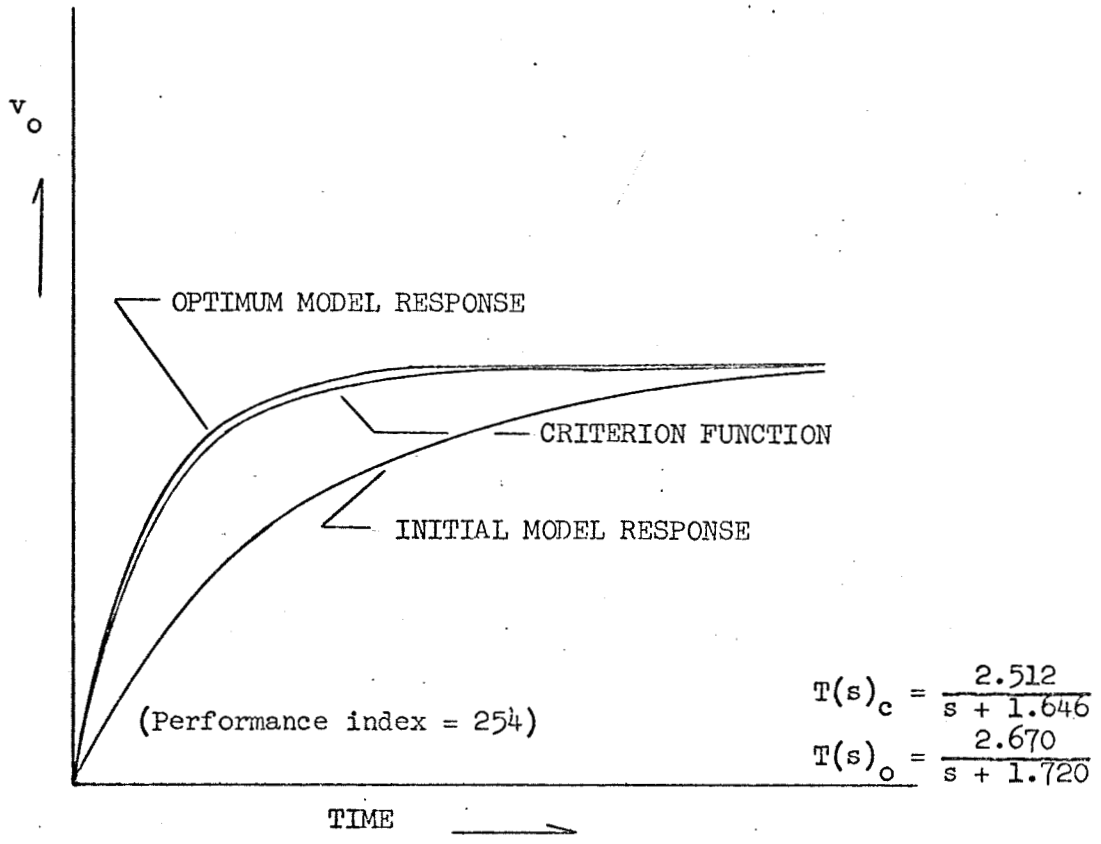


Figure 4-3. Optimum dynamic response for a first-order low-pass filter.

The criterion function and the successive model response functions are monitored on a dual-trace storage CRO. The printed output is listed in Table 4-1. There it is seen in addition to the gain constant of 2.512 and the cut-off frequency of 1.646 that the gain constant of the near-optimum model is 2.670 and the cut-off frequency is 1.720 for the near-optimum model obtained by this direct designed technique.

An iteration time of the order of milliseconds is possible using a relatively inexpensive hybrid system. A single iteration on a relatively expensive digital computer requires approximately one second using a fourth order Runge-Kutta integration routine.

4.1.2 Network Structure Optimization

Another technique has been developed for eliminating the usual assumption of model form with the result that the optimum form, as well as the optimum set of parameters is found by "growing" models. The use of the hybrid computer is indicated by the block diagram of Figure 4-4. An assumed model of low order is programmed on the analog portion of the hybrid computer. The transient response of the model to a given input is obtained for an initial set of parameter values, and the performance index (IP) is computed. The parameter values are varied under control of the digital portion and the response is again obtained without operator intervention. The performance index from this second run is compared with that from the first run. A search algorithm is used to check for improvement in the performance index. Parameters are again varied with the aid of high speed digital parameter units (DPU), and the process is repeated until the optimum set of parameters is identified for the low-order model and the optimum performance index is stored. The form of

K_c	0 0 0 0 2. 3 1 2
a_c	0 0 0 0 1. 6 4 6
K_0	0 0 0 0 2 6 7 0
a_0	0 0 0 0 1 7 2 0
IP(v)	0 0 0 0 0 2 5 2
IP(v+1)	0 0 0 0 0 2 5 4

Table 4-1. Tabulated output from direct design of a first order-low-pass filter.

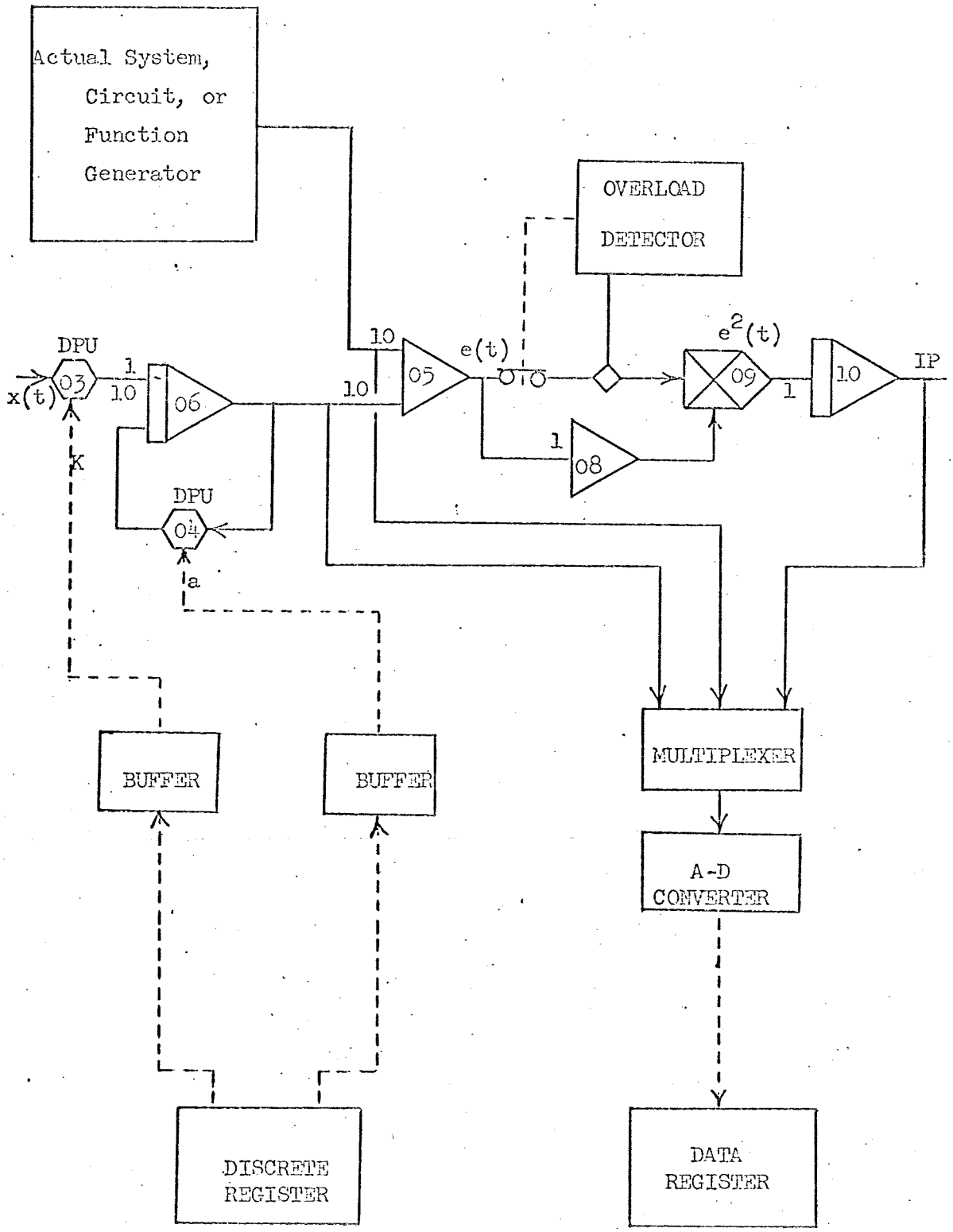
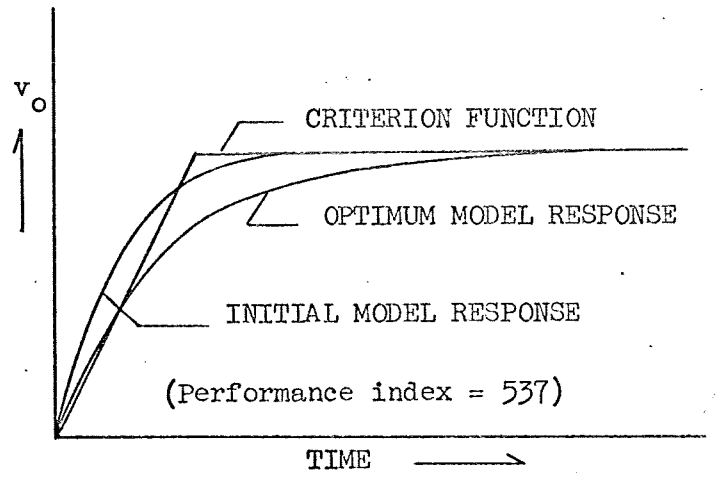


Figure 4-4. Hybrid Computer Block Diagram for System Modeling

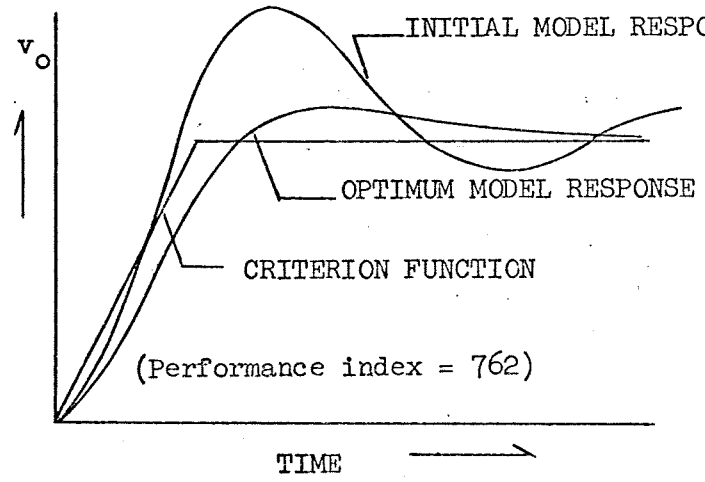
the model is changed to the next higher order model under control of the structure optimization algorithm through the use of automated patching of the analog portion of the hybrid computer. Then, an optimum set of parameters is obtained for the second model form, and the optimum performance index is compared with that for the former model form. This process is continued under control of the search algorithm until direct optimization of model form, as well as the parameters of this model, is obtained. Overload detection in and automated scaling of the analog portion are required in this process.

In the course of this work it has been found that the standard canonical forms illustrated in Figure 4-4 should be modified for more efficient hybrid optimization in "growing" models. The "growth" model structures shown in Figures 4-5 through 4-7 have been devised in this work to provide better initial parameter estimates than obtained with the canonical forms. Also, the factored form of the transfer function is always available with the growth model structures.

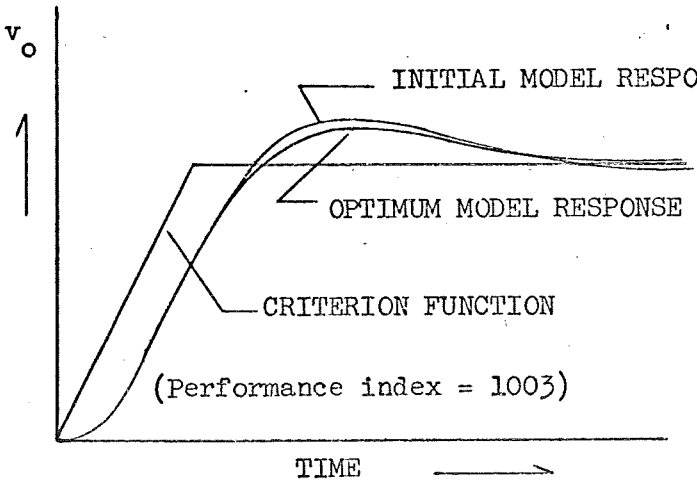
The use of the growth model in the realization of model structure modification is illustrated in Figure 4-8. There it is shown that models or order 1, 2, and 3 can be realized under digital control by the switching logic shown. Higher order models can be obtained by successively combining the first and second order growth models. The associated switching logic for the higher order models can be obtained algorithmically. The results obtained by growing models at high speed are shown in Figures 4-5 through 4-7. In Figure 4-5 it is seen that a first-order model is "optimal" for the hard-limited response shown. A second-order response is correctly identified with a second-order model as shown in Figure 4-6. In Figure 4-7 it is seen that a third-order model is "optimal" for the delayed



$$H(s) = \frac{1.449}{s + 1.347}$$



$$H(s) = \frac{5.017}{s^2 + 2.213s + 4.828}$$



$$H(s) = \frac{27.35}{(s^2 + 2.407s + 4.978)(s + 4.981)}$$

Figure 4-5. Examples of optimum direct design of various order systems for a hard-limited criterion function.

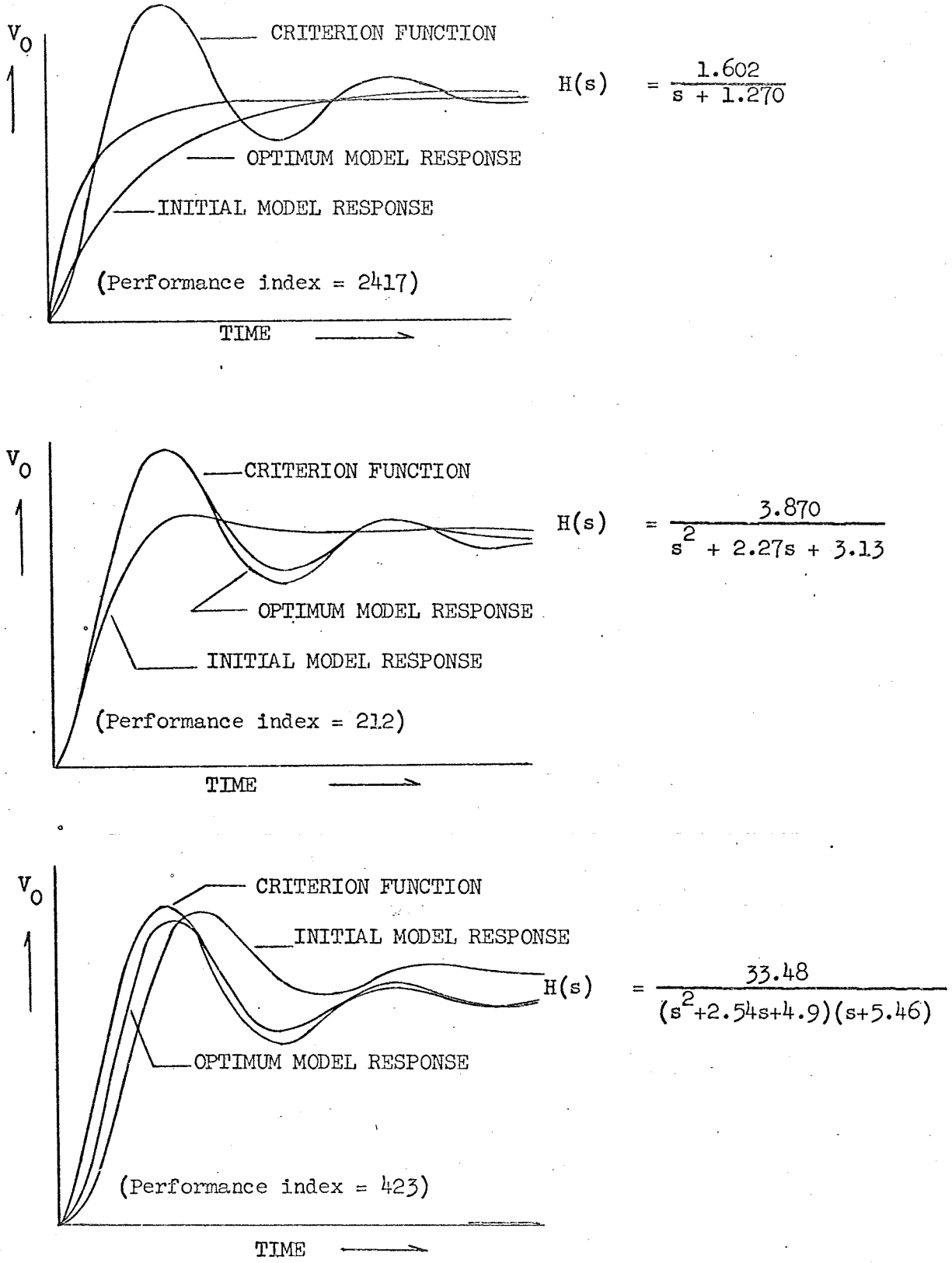


Figure 4-6. Examples of optimum direct design of various order systems for a second-order criterion function.

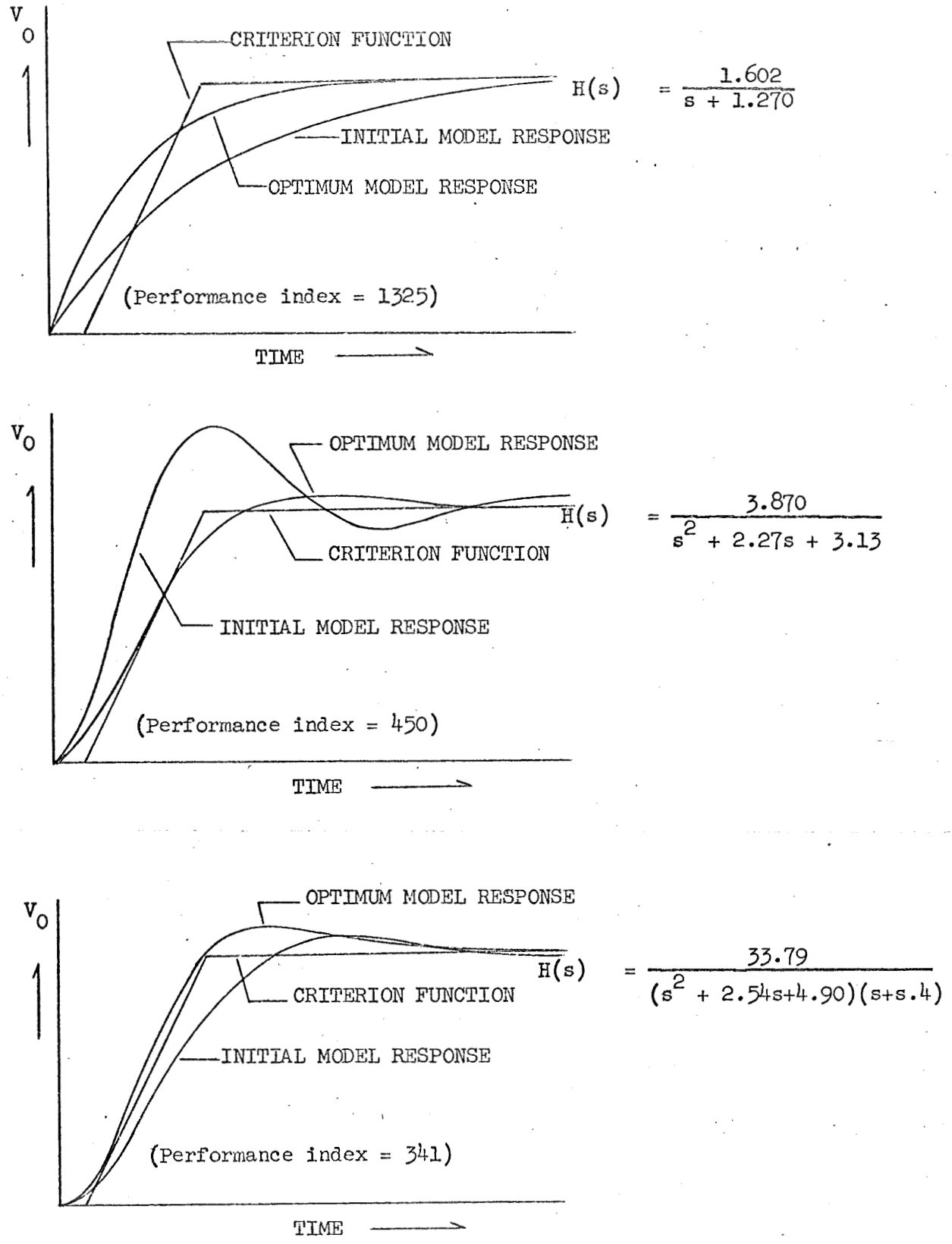
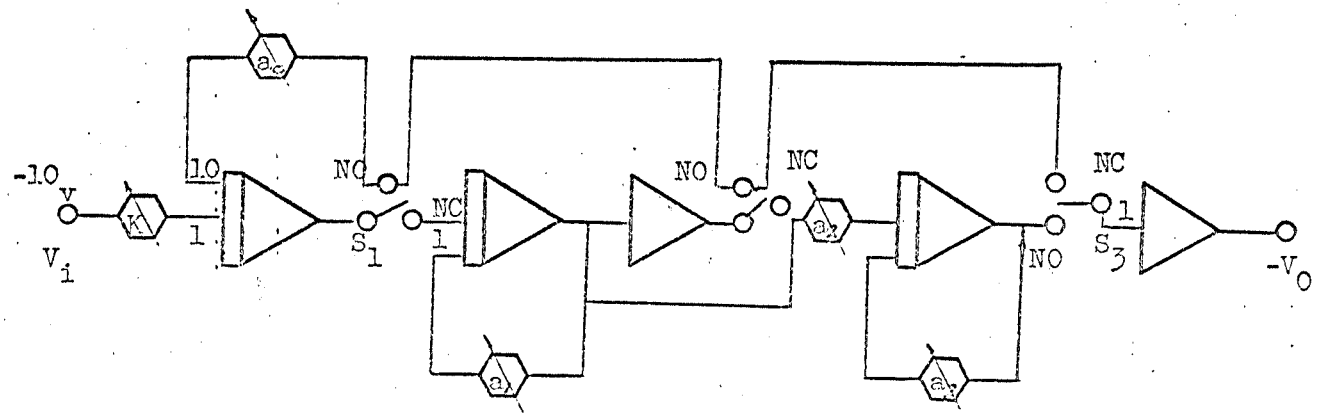


Figure 4-7. Examples of optimum direct design of various order systems for a delayed hard-limited criterion function.



Model \ Switch	S_1	S_2	S_3
First-order	E	$\overline{E}$	$\overline{E}$
Second-order	$\overline{E}$	E	$\overline{E}$
Third-order	$\overline{E}$	E	E

E: ENERGIZED

DIGITAL PARAMETER UNIT

Figure 4-8. Growth Model for Realization of Model Structure Modication

hard limited response shown.

The structure modification method developed in this work is believed to be a contribution to optimal system synthesis in that it accomplishes high-speed automated identification of dynamic systems, without a priori knowledge of the model form, as illustrated by the results reported here. In addition, the method makes immediately available to the designer a physically realizable direct model constructed of analog computer elements.

4.1.3 Nonlinear Network Design

Experience has shown that there is often need for methods that avoid such assumptions as linear behavior and a priori knowledge of model form. Nonlinear circuits and devices can also be designed using hybrid techniques and time-domain input-output measurements or specifications. Mathematical methods for linesr systems have been unified by the superposition principle. In contrast, there is no sufficiently powerful principle that will unify nonlinear system theory. Consequently, there is no general approach to the modeling of nonlinear systems. Methods are, therefore, forced to be specified for application to a restricted class of systems. This work treats the class of zero-memory large signal nonlinearities. It must be further specified that these nonlinearities can be modeled as continuous, single-valued functions where the nonlinear portion can be cascaded with a linear dynamic subsystem. Theoretical and experimental methods have been developed to study[1-5] the characteristics of nonlinear systems driven by random input signals. This emphasis on the use of random signals as the probe for nonlinear system characteristics is particularly significant if theory is to be applied to the modeling of real physical systems.

A method has been developed for high-speed, fully-automated modeling of a class of nonlinear systems driven by ergodic band-limited white random inputs. A situation frequently encountered is that not only are some or all of the parameters of a model unknown, but the structure of the model may also be unknown. A technique has also been developed for eliminating the usual assumption of model structure for the linear dynamic subsystem with the result that the optimum structure, as well as the optimum set of parameters is identified[6]. For this class of problems, it has been demonstrated that cost and speed advantages favor hybrid computation over all-digital computation[7].

In many actual physical systems, the nonlinearities occur primarily in the large signal behavior whereas a linear idealization leads to a very reasonable approximation of small signal system operation. The class of system under consideration contains a zero-memory nonlinear subsystem[8] as shown in Figure 4-9.

For a single input-single output nonlinear element, the output m is approximately equal to the output of a piecewise linear representation for a given input r[9]. The output of a zero-memory, single valued nonlinear element can be expressed as

$$m = F\ (r,\ m_o,\ \alpha_1,\ \ldots\ \alpha_n) \qquad (4\text{-}4)$$

where each parameter α_i is the slope of a segment of the piecewise linear function.

For a bounded input, quantization of the input dynamic range yields

$$\Delta r = \frac{R_{max} - R_{min}}{n} \qquad (4\text{-}5)$$

where R_{max} and R_{min} are upper and lower bounds on the input, and n is the number of quantization segments. To specify the parameters of the

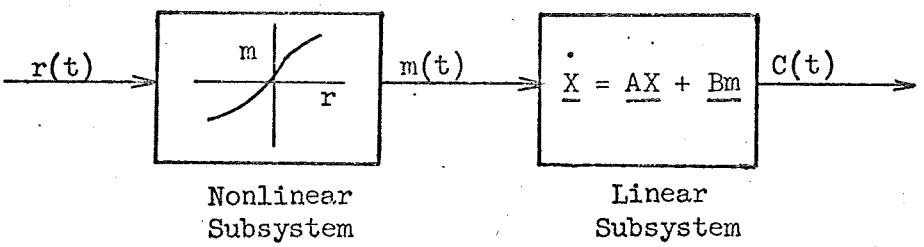


Figure 4-9. Block Diagram for Nonlinear System Model

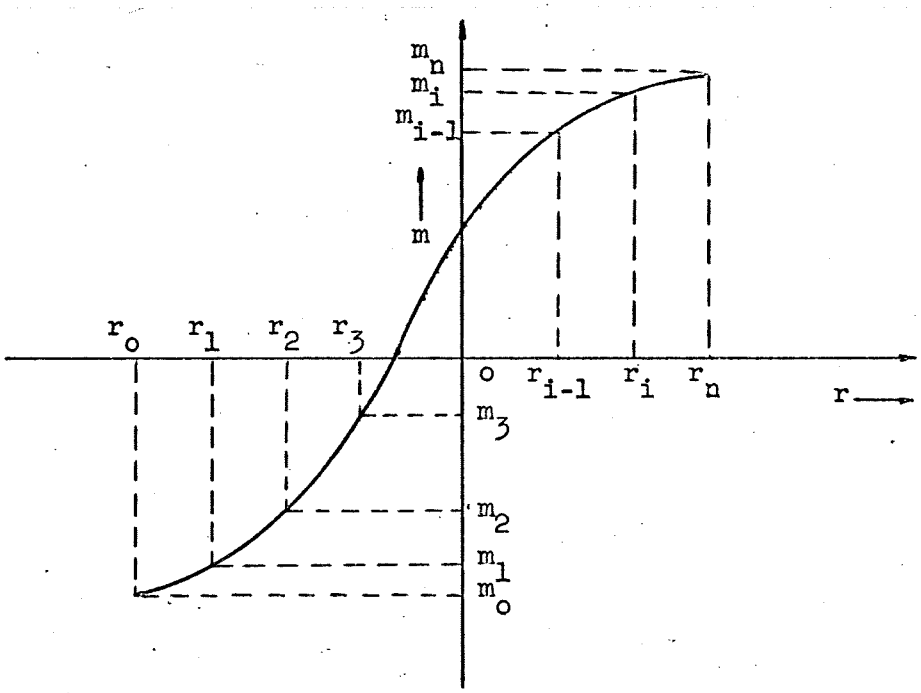


Figure 4-10. Single Valued Nonlinear System Model Characteristics

nonlinear element, let

$$\alpha_i \triangleq \frac{dm}{dr}\Big|_r \qquad \begin{array}{l} i\Delta r < r \leq (i+1)\Delta r \\ 1 \leq i \leq n \end{array} \tag{4-6}$$

For simplicity, let

$$m_o = m\,(R_{min}) \tag{4-7}$$

and

$$m_i = m\,(i\Delta r) \triangleq m\,(r_i) \tag{4-8}$$

Then

$$m = \sum_{i=1}^{n} \left(\left[r-r_i\right]\alpha_i + m_{i-1}\right)\left[u\,(r-r_i) - u(r-r_{i+1})\right] \tag{4-9}$$

Since m_{i-1} can be expressed as

$$m_{i-1} = \Delta r \sum_{k=1}^{i-1} \alpha_k + m_o \tag{4-10}$$

as indicated in Figure 4-10, Equation (4-8) becomes

$$m = \sum_{i=1}^{n} \left[(r-r_i)\,\alpha_i + \Delta r \sum_{k=1}^{i-1} \alpha_k + m_o\right]\left[u(r-r_i)-u(r-r_{i+1})\right] \tag{4-11}$$

Therefore, the output of the nonlinear element is

$$m = G\,(r,\ n,\ m_o,\ \alpha_1,\ \ldots,\ \alpha_n) \tag{4-12}$$

This nonlinear portion is cascaded with a typical linear subsystem described by the state differential equation

$$\dot{\underline{X}} = \underline{A}\ \underline{X} + \underline{B}\ \underline{M} \tag{4-13}$$

If the linear portion is a network of known form it will be desirable to use a transform approach for representation of the linear dynamics. This can be accomplished easily by obtaining the transfer function using the computer-aided network analysis program NASAP* [10].

*Network Analysis for System Applications Program developed by NASA/Electronics Research Center.

The dynamic output of the total single input-single output cascaded system is

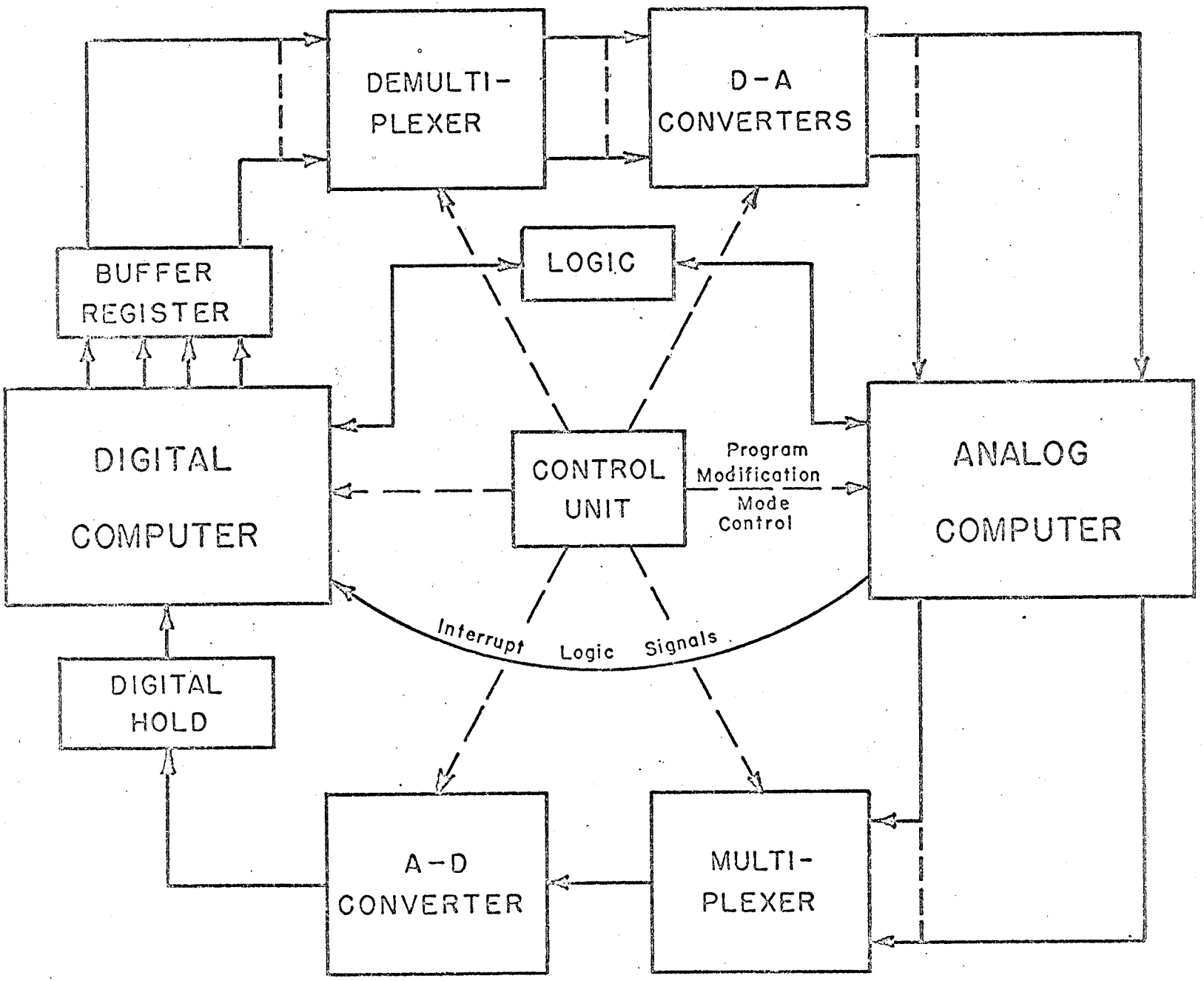
$$C_m(t) = f\left[r(t), \Delta r, m_o, \underline{\alpha}_i, \underline{a}_{jk}\right] \qquad 1 \leq i \leq n \qquad (4\text{-}14)$$

where the $\underline{a}_{jk}$ are related to parameters of the linear subsystem.

The modeling method makes use of hybrid computation and parameter optimization techniques[11]. The use of the hybrid computer is indicated by the block diagram of Figure 4-11. An assumed nonlinear model structure is programmed on the analog portion of the hybrid computer. The automated patching of the linear portion is controlled by the structure optimization algorithm portion of the digital program. The parameters of both the linear and nonlinear portions of the model are set under control of the digital program. High speed digital parameter units (DPU) have been constructed to provide for this automated feature.

The transient response of the model for a given input function is obtained for an initial set of parameter values, and the performance index (IP) is computed. Then, the response is again obtained for a variation of parameter values without operator intervention. The performance index from this second run is compared with that from the first run. An optimization algorithm is used to check for improvement in the performance index. Based on this decision, a search algorithm is used to select a new set of parameters. This process is repeated until the optimum set of parameters is identified for the assumed model structure. The form of the model structure of the linear portion is changed to the next higher order. Then, the optimum set of parameters is obtained for the second model structure, and the optimum performance index is compared with that for the former model structure. This process is continued under control of the optimization algorithm until the optimum model structure,

FIGURE 4-11. TULANE HYBRID COMPUTER SYSTEM



as well as the optimum set of parameters for this model, is identified. Parameters of the nonlinear portion are included by simply extending the parameter vector.

In obtaining the "best" values for the parameters of the model, it is necessary to establish a comparison criterion or performance index. An obvious choice is an index based on the difference (error) between the actual transient response of the physical system being modeled and the transient response of the computer model operated in parallel with and driven by the same input as the physical system[12]. The choice of performance index will be influenced by the type and the characteristics of the driving function.

In some cases, the economic or safety considerations associated with experiments which disturb the system or drive it outside of the normal operating range may preclude the use of classical analysis techniques. Only normal operating data with its inherent noise components is available, in these cases. To assure reliability when system parameters are variable, on-line continuous model updating may be required. If the system is high-order or nonlinear, the solution time for all-digital computation could be excessive. If it can be assumed that parameters are fixed, then it is possible to record driving and response functions for off-line modeling[13]. In other situations, the normal system operation can be shut down and off-line excitation is possible. It is then possible to drive the system with a random noise signal or a step function input.

If the system can be assumed stationary and ergodic for any of the situations in which a random driving function is used, then ensemble averages can be replaced by time averages. In such cases the mean value may be estimated from the finite time average

$$\overline{X(t)}_T = \frac{1}{T}\int_o^T x_j(t)\,dt \qquad (4\text{-}15)$$

where $x_j(t)$ is a particular sample function of the process, and T large compared to the lower bound of the frequency spectrum $X(\omega)$. With the presence of low frequency signal components, excessively long integration times are required to reach the mean value. A mean square error criterion is commonly used as a measure of the performance index for the study of random processes. This function can be expressed as

$$\overline{IP}_T = \frac{1}{T}\int_o^T e^2(t)\,dt \quad \text{for T long} \qquad (4\text{-}16)$$

Since the error is the difference between the actual transient response of the physical system and the transient response of the computer model, Equation (4-16) becomes

$$\overline{IP}_T = \frac{1}{T}\int_o^T \left[C(t) - C_m(t)\right]^2 dt \qquad (4\text{-}17)$$

Substituting for $C_m(t)$ given in Equation (4-14) there results

$$\overline{IP}_T = \frac{1}{T}\int_o^T \left\{C(t) - f\left[r(t),\ m_o,\ \underline{\alpha}_i,\ \underline{a}_{jk}\right]\right\}^2 dt \qquad (4\text{-}18)$$

Hence, the optimum or minimum value of IP is a function of the parameters of the nonlinear portion of the model as well as those of the linear subsystem.

The experimental determination of the characteristics of a given nonlinear system, based upon the described method, is demonstrated as shown in Figure 4-13. The given nonlinear system is characterized by the expression

$$\dot{C}(t) + 3.14\,C(t) = 2m(t) \qquad (4\text{-}19)$$

where m(t) is the characteristic of a nonlinear element expressed by

$$\begin{aligned} m(t) &= 0 && \text{for } 0 \geq r(t) \\ &= 2r(t) && \text{for } 0 < r(t) \leq 1 \\ &= 2 && \text{for } r(t) > 1 \end{aligned} \qquad (4\text{-}20)$$

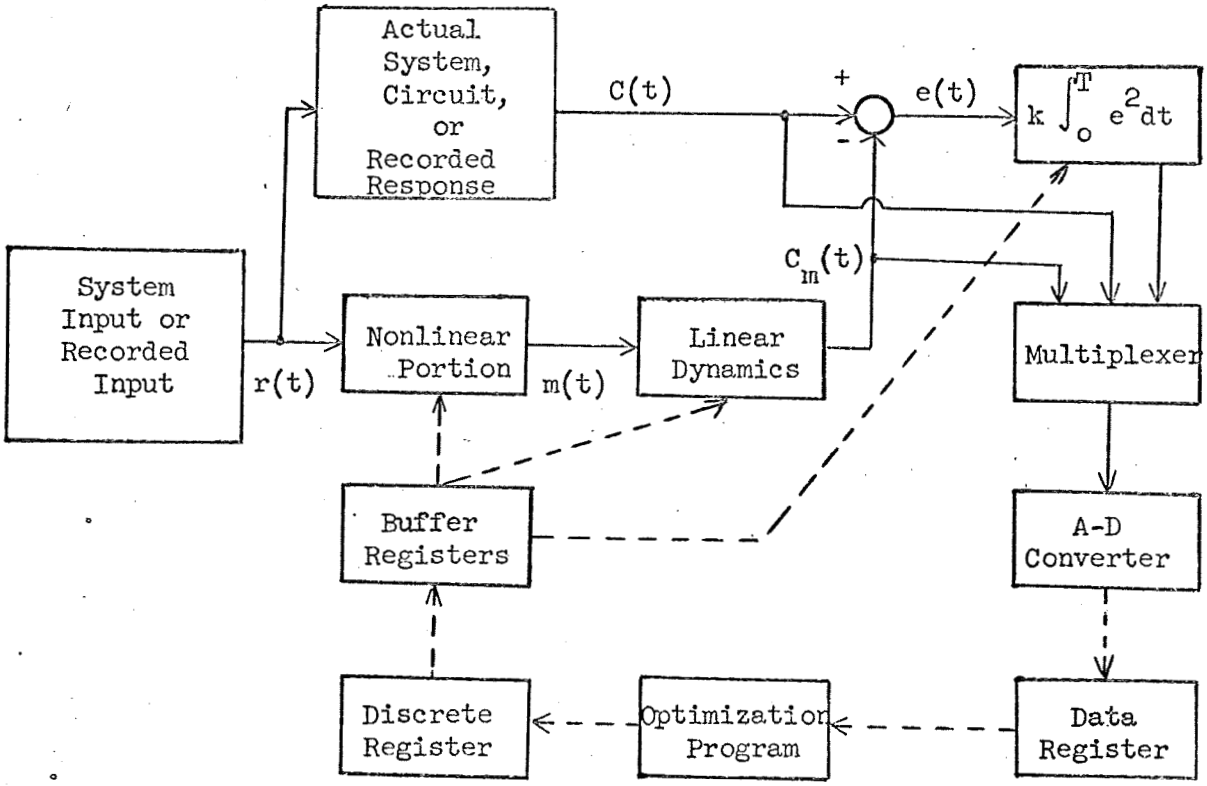


Figure 4-13. Hybrid Computer Block Diagram for Direct System Modeling

The given system and the modeled system, which is constructed of a cascade combination of a nonlinear element and a linear dynamic subsystem, are both simulated on the analog computer with random input. Since an entire ensemble of driving functions may arise under actual operating conditions, a random driving function such as an ergodic, band-limited, white noise generator with Gaussian amplitude distribution can appropriately be used for excitation. For practical reasons, the mean square error performance index given in Equation (4-16) has been modified to give a weighted integral square error (WISE) function of the form

$$IP = k \int_0^T e^2(t)\, dt \qquad (4\text{-}21)$$

Observation time, T, for the measurement of the performance index is determined by a preliminary parameter perturbation procedure. T is acceptable if the variance of the measurement of the performance index, defined by WISE, is small compared with the difference in absolute values of the IP measurements. Figure 4-14 shows the relation between T and the performance index measurements with different parameter perturbations. Obviously, from Figure 4-14, if an arbitrary point in the parameter space is far away from the desired optimum point, the error signal will contain a relatively large mean value; also, the absolute value of the variance of IP measurements will be large. Similarly, small absolute variance of IP measurements will occur for small parameter perturbations.

In this hybrid optimization algorithm, a large parameter increment step $\bar{p}$ is desired initially to prevent the measurements of IP from overlapping such that the difference of IP measurements with different parameter sets will be distingushable. For the point close to the optimum point in parameter space, a small step increment of $\bar{p}$ is acceptable

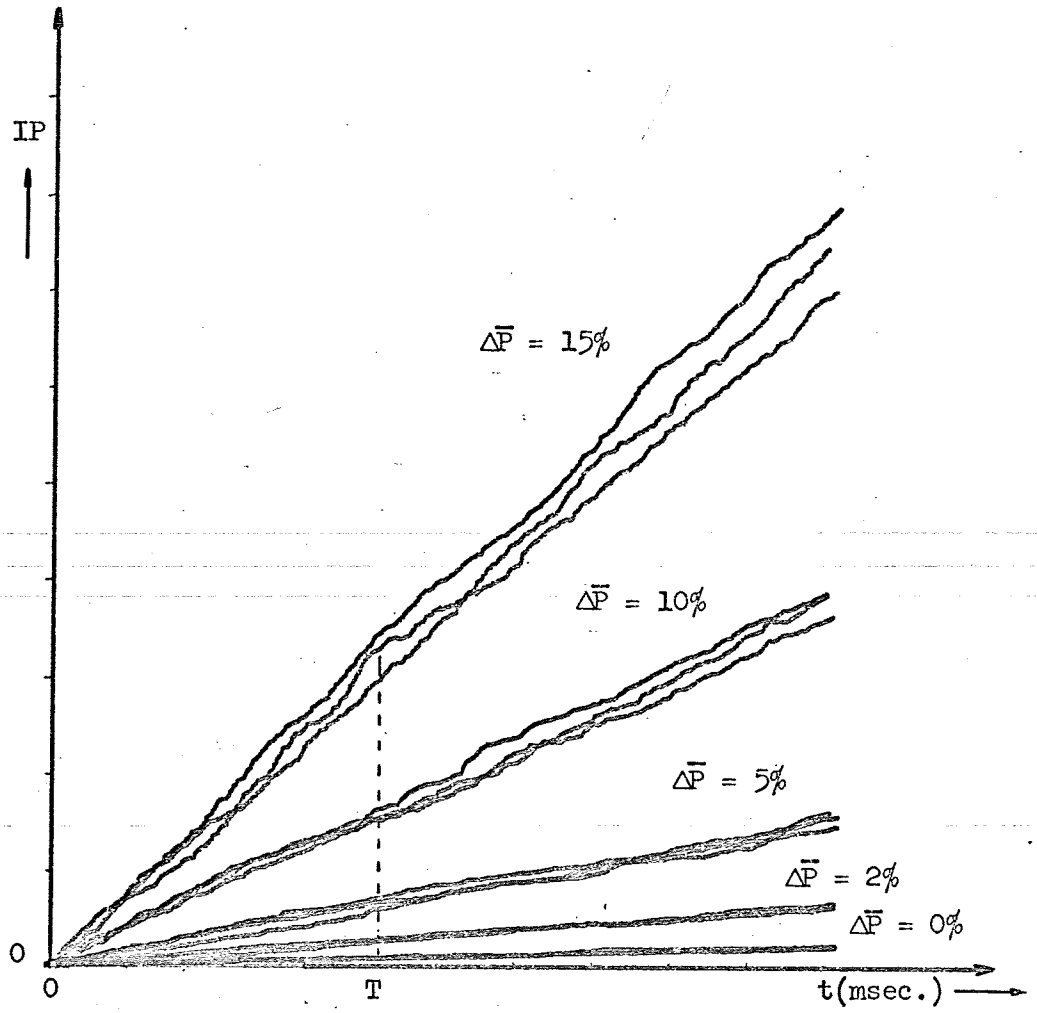


Figure 4-14. Performance Index (WISE) Measurements for Various Parameter Perturbations

because of the small absolute variance of the IP measurements.

For a given system as specified in Equation (4-20), the optimum structure was found to be first order and the parameters of the linear subsystem were found to be 1.89 for K_o and 3.11 for a. The nonlinear characteristic of the optimum model was identified as:

$$\begin{aligned} m(r) &= 0 && \text{for } 0 \geq r \\ &= 1.86\, r && \text{for } 0 < r \leq 1.1 \\ &= 2.05 && r > 1.1 \end{aligned} \qquad (4\text{-}22)$$

This off-line example assumes partitioning of the large signal nonlinearity and the linear dynamics. The optimum parameter set for the nonlinear portion can be identified independently of the optimum model structure and parameter set for the linear subsystem. A small step input is used to probe for the linear subsystem characteristics. Then, a random noise source is used to probe independently for the characteristics of the nonlinear portion.

4.1.4 Network Design Using NASAP Sensitivity Evaluation

This section describes a NASAP[10] oriented hybrid computer optimization technique for network design. The network design is based upon time domain measurements or specifications. The optimum network is a combination of the initially designed network and a network perturbation evaluated by the use of network sensitivities with respect to the adjustable network parameters.

This work has resulted from application of hybrid optimization techniques[11,14] to network design. The most appropriate methods considered make use of one or more of the strong points of either analog or digital computers. In this work the high speed parallel operation of all analog

computational elements including multiplication and integration is exploited fully, as well as automatic control of the digital program on the basis of calculations. High speed iterative computation and logical operations as required in optimization calculations are also possible through the linkage of digital and analog computers.

Because of the simultaneous nature of analog operations, solutions of higher order problems can be executed with no appreciable increase in execution time. Since execution time for digital computer programs increases for higher order systems, the expense of computer-aided design may become economically prohibitive for certain types of calculations such as those which involve optimization techniques.

If an assumed network topology and time-domain input-output test measurements or design specifications are given, then the network can be modeled on the analog computer using the "breadboard" technique. The breadboard technique retains circuit topology, hence the individual network elements are parameters of the model. This technique is tedious to implement for large networks, and it becomes impractical because of hardware limitations.

If input-output relationships are of interest, then conventional analog computer programming based on transfer functions is preferred because of the hardware advantage. However, computation of the transfer function is also tedious for large networks. Since the digital computer program, NASAP[15] yields the transfer function of a given network as a standard output, it will be desirable to use this off-line capability of NASAP to perform this computation. The use of transfer functions restricts the technique to linear networks.

If a transfer function is used to represent the mathematical model of a network, then the designer will be confronted with the difficulty of

adjusting network parameters by perturbing transfer function coefficients. The technique reported here which uses NASAP sensitivity evaluations offers significant advantages in this regard for a restricted class of design problems.

For a given network with n adjustable parameters, the terminal behavior can be expressed as $F(s,\bar{p})$ where $\bar{p}$ is defined as an n-dimensional parameter vector. If the parameter vector changes by an incremental amount, $\overline{\Delta p}$, then $F(s,\bar{p})$ changes by an incremental amount given approximately by the first term of a Taylor series as

$$\Delta F(s,\bar{p}_o) = F(s,\bar{p}) - F(s,\bar{p}_o) \qquad (4\text{-}23)$$

$$= \bar{F}'(s,\bar{p}) \cdot \overline{\Delta p} \qquad (4\text{-}24)$$

In Equation (4-24), $\bar{F}'(s,\bar{p}_o)$ is defined as an n-vector of first order partial derivaties of $F(s,\bar{p}_o)$, the i^{th} element being $\frac{\partial F(s,\bar{p}_o)}{\partial p_i}$ and $\overline{\Delta p}$ is defined as an n-vector of $(\overline{p-p_o})$, where $\bar{p}_o$ is an initial set of parameters.

Each element of $\bar{F}'(s,\bar{p})$, $\frac{\partial F(s,\bar{p})}{\partial p_i}$, can also be defined as the reciprocal of the small signal sensitivity $B^F_{p_i}$ of $F(s,\bar{p})$ with respect to parameter p_i. Thus

$$\frac{\partial F(s,\bar{p}_o)}{\partial p_i} = 1/B^F_{p_i} = Q^F_{p_i} \qquad (4\text{-}25)$$

Substitution of Equation (4-25) into Equation (4-24), leads to

$$\Delta F(s,\bar{p}_o) = \overline{1/B^F_p} \cdot \overline{\Delta p} = \overline{Q^F_p} \cdot \overline{\Delta p} \qquad (4\text{-}26)$$

For a given input function $R(s)$, the system response is given by

$$Y(s,\bar{p}) = R(s) \cdot F(s,\bar{p}) \qquad (4\text{-}27)$$

The change in the response, $Y(s,\bar{p})$ caused by the increment $\overline{\Delta p}$ is given by

$$\Delta Y(s,\bar{p}) = R(s) \cdot \overline{Q^F_p} \cdot \overline{\Delta p} \qquad (4\text{-}28)$$

If an initially designed network is reasonably close to the optimum, then the response of this near optimum network can be expressed in terms of the response of the initially designed network as

$$Y(s,\overline{p_o+\Delta p}) = Y(s,\overline{p}_o) + \Delta Y(s,\overline{p}_o) \qquad (4\text{-}29)$$

Substituting Equations (4-26), (4-27), and (4-28) into Equation (4-29) yields

$$Y(s,\overline{p_o+\Delta p}) = R(s)\left[F(s,\overline{p}_o) + \overline{Q_p^F} \cdot \overline{\Delta p}\right] \qquad (4\text{-}30)$$

Equation (4-30) leads to the transfer function of the desired design expressed as

$$F(s,\overline{p_o+\Delta p}) = \frac{Y(s,\overline{p_o+\Delta p})}{R(s)} \qquad (4\text{-}31)$$

$$= F(s,\overline{p}_o) + \overline{Q_p^F} \cdot \overline{\Delta p} \qquad (4\text{-}32)$$

If the desired time domain response for a given input is $y_d(t)$, then a design strategy based upon the improvement of the performance index by adjusting the increments $\overline{\Delta p}$ is performed until an optimum design is identified. The performance index can be arbitrarily defined as

$$IP(\overline{p}) = \int_o^T \left| y_d(t) - y(t,\overline{p_o+\Delta p}) \right| dt \qquad (4\text{-}33)$$

where $y(t,\overline{p_o+\Delta p})$ is the inverse Laplace transform of Equation (4-30). Figure 4-15 shows the functional block diagram of this hybrid computer implementation.

The design algorithm described previously was based upon the effect of network parameter perturbation in terms of parameter sensitivity evaluations. In this work the NASAP program has been used to calculate the transfer function of a given network and the sensitivity evaluations with respect to each adjustable parameter.

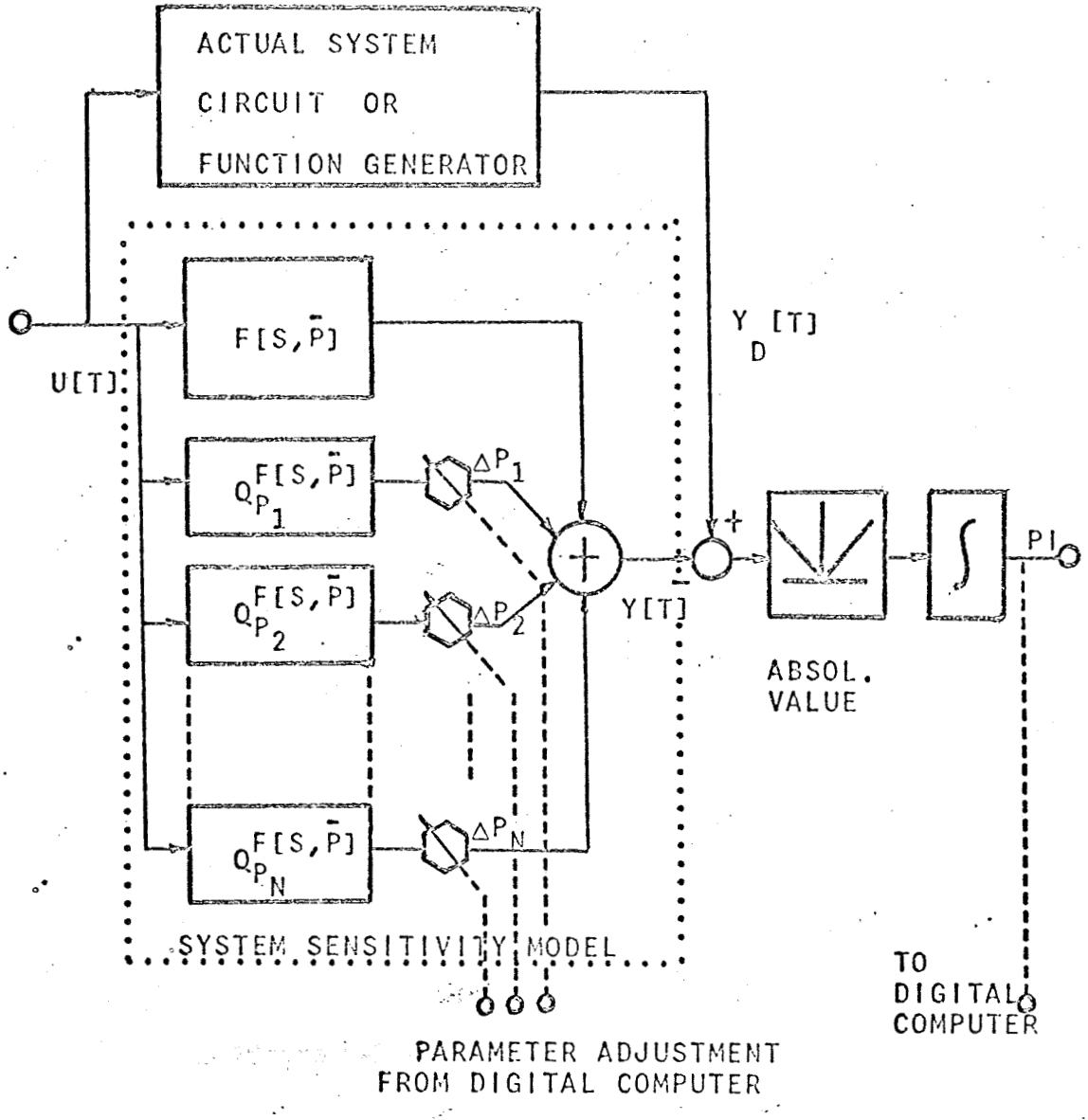


Figure 4-15. Functional Block Diagram for Hybrid Computer Optimization Including Sensitivity Functions

The NASAP program provides a basis for establishing three different types of sensitivity calculations depending upon the particular definition of sensitivity desired, namely:

$$\text{Type G} \qquad G_p^F \triangleq P/F$$

$$\text{Type B} \qquad B_p^F \triangleq dP/dF \tag{4-34}$$

$$\text{Type S} \qquad S_p^F \triangleq \frac{dF/F}{dP/P}$$

F is the transfer function of a given network and P is an arbitrary network parameter.

In this NASAP-hybrid oriented optimization technique, sensitivity defined by type B is desired. In the standard NASAP outputs[15], sensitivity of type B is given by

$$B_p^F = dP/dF = 1/\frac{dF}{dP} \tag{4-35}$$

$$= -\frac{H(0,1) + H(1,1)}{H(1,0) + H(1,1)} \tag{4-36}$$

where the H's are topology equations of the given network produced by a sensitivity output request in the NASAP program.

The significance of NASAP in this design method can be obtained from the block diagram of Figure 4-16, which describes the algorithm of this hybrid optimization technique. The transfer function of the initially designed network is obtained with NASAP and is programmed on the analog portion of the hybrid computer. The partial derivatives of $F(s,\bar{p})$ as indicated in Equation (4-24) are also obtained from the NASAP program by requesting sensitivity evaluation with respect to each of the adjustable parameters. For an n parameter network design problem, (n+1) NASAP runs will be required. The perturbation function given by Equation (4-26) is also programmed on the analog computer using digital controlled parameter

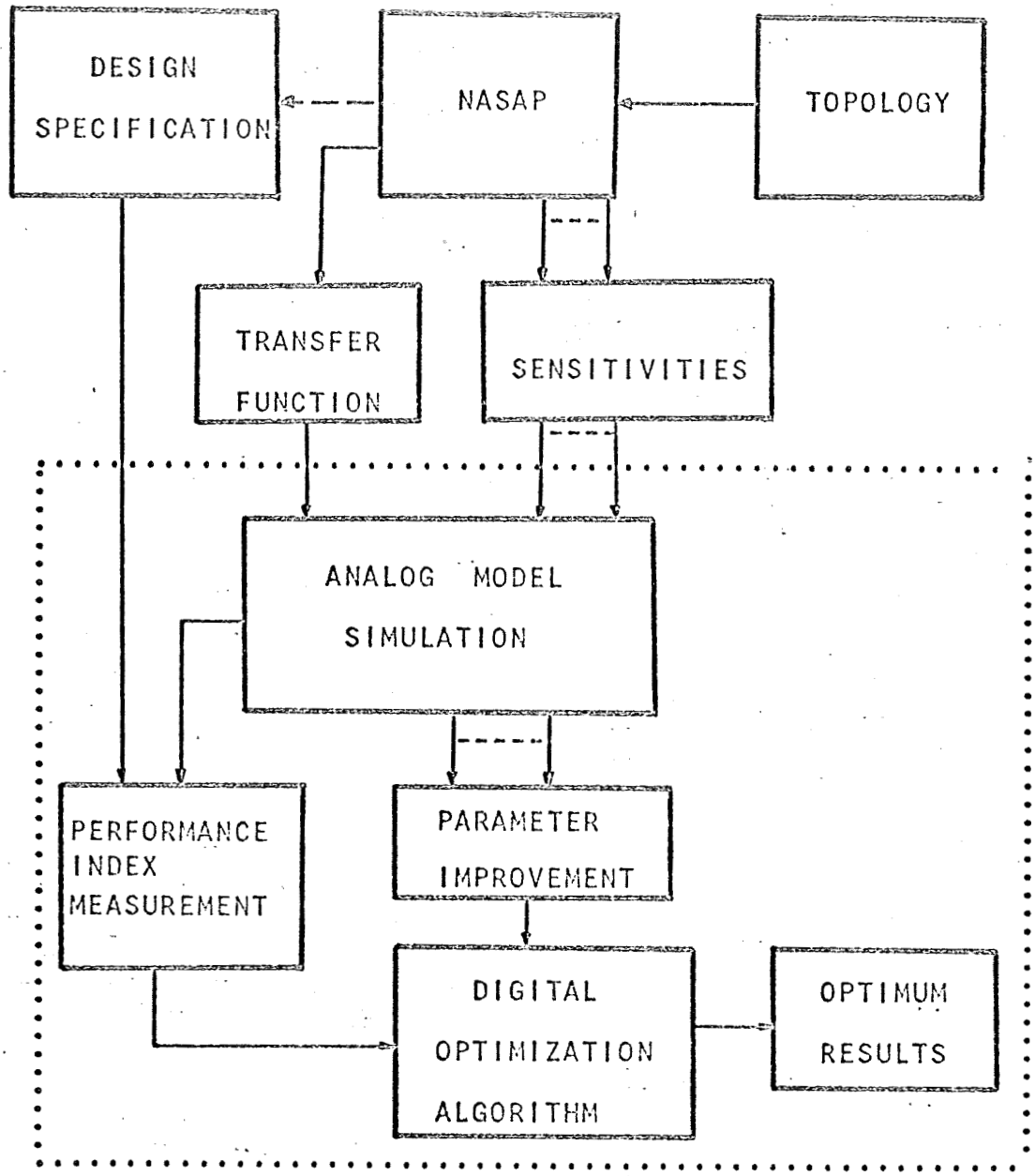


Figure 4-16. Block Diagram for NASAP-Hybrid Network Design

units for adjusting parameter increments $\overline{\Delta p}$. Details of the optimization techniques using the Tulane Hybrid Computer System have been reported elsewhere[11,14,16].

The network shown in Figure 4-17 illustrates the network design technique developed in this work. For this RC coupled FET amplifier, it is desired to obtain an optimum set of parameter values for bias resistor R_S and load resistor R_L to meet a given time domain input-output specification. For the purpose of laboratory demonstration, the given specification is generated by a time domain response of a transfer function. This transfer function is also evaluated using the NASAP program for the network shown in Figure 4-17 with R_S = 10KΩ and R_L = 10KΩ. The NASAP program is used to evaluate the transfer functions and sensitivities. The NASAP coded equivalent circuit is shown in Figure 4-18. This example has been demonstrated on the Tulane Hybrid Computer System. The computation result obtained was R_S = 9.27KΩ and R_L = 11.3KΩ.

It has been observed from the use of this technique that there are two inherent factors limiting the accuracy of the computational results. One is the approximation given in Equation (4-29). A more accurate design result can be obtained when the initial parameters $\overline{p_o}$ are chosen closer to those of the optimum network. The optimization algorithm used in this work also limits accuracy because a computational tolerance is provided in the optimization scheme to save computer time. Design accuracy can be improved in this case by setting a smaller computational tolerance.

This work provides another example of the speed and cost advantages of a hybrid computer system over pure digital computers for network design using parameter optimization techniques. The NASAP program has contributed significantly (in this hybrid method) by allowing evaluation of transfer functions and sensitivity functions for a given network or system

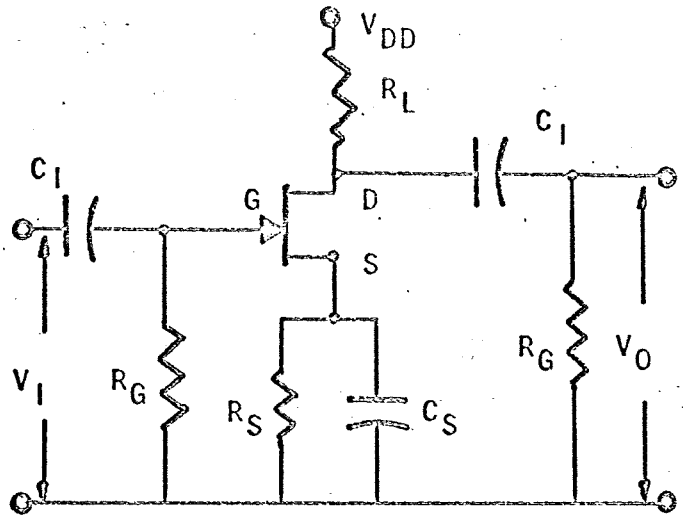


Figure 4-17. Schematic Diagram of an RC Coupled FET Amplifier

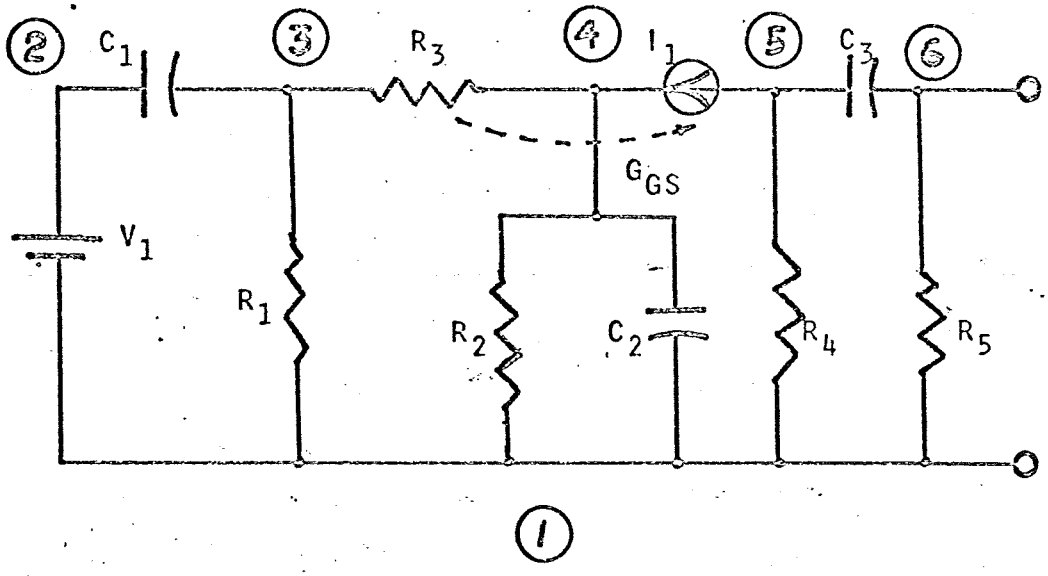


Figure 4-18. NASAP Coded Equivalent Circuit for and RC Coupled FET Amplifier

topology. At present the procuedure presented in this paper is executed off-line. In a hybrid facility with NASAP capability, this repetitive procedure could be automated for on-line use.

A major area for future investigation on this topic is the design of networks involving nonlinear elements. Hybrid computation offers significant features in modeling nonlinear characteristics[17] by hardware oriented programming.

4.2 Interpretation of Approach

A thorough review of pertinent technical literature and discussions with others who are experienced in the field of hybrid computation revealed that the most significant circuit design application for hybrid computation was in the area of dynamic optimization. This mode of computation involves iterative solutions composed of dynamic circuit simulation on the analog computer and digital optimization and search calculations. This manual presents the applications of hybrid computation and optimization techniques to the design of circuits/systems. The chief merits of these hybrid-optimization techniques are speed of execution and low hardware cost compared to those associated with the use of large-scale digital computers. Computer-aided circuit/system design is an example of a computer application which requires the determination of accurate models of physical devices and systems. Since it is assumed that time domain specifications are required, the transient specifications of existing devices and systems can be generated by parallel operation with the model. Only real-time operation is possible in this case unless the device is modified[18-20].

It has been observed that considerable benefit is gained by placing emphasis on the application of certain classes of automated techniques to

typical problems as well as on the specific techniques themselves. By using simple techniques initially, sophisticated programming requirements do not dominate the presentation. Understanding why and when certain classes of techniques should be used may ultimately prove to be of far more value than the mastering of any one sophisticated technique. This will be especially true if the suitability of certain techniques changes because of future hardware developments.

The most significant accomplishments of this work relate to the techniques for automated on-line nonlinear device modeling, model structure optimization, and the use of sensitivity analysis. These techniques require specialized hardware developments. The "growth" model devised in this work provides better initial parameter estimates than obtained with the canonical forms. Also, the factored form of the transfer function is always available with the growth model structures. In addition, these methods make a physically realizable direct model constructed of analog computer elements immediately available to the designer.

The experience of the authors to date indicates at least an order of magnitude ratio in speed and in capital expenditure, both favoring hybrid computation over all-digital computation, for this class of problems at present. As the order of the dynamic system increases, the favorable speed ratio will be further increased.

4.3 Limitations of Techniques

The major disadvantages of hybrid computation are the requirement of hardware-oriented, real-time programming and the limitation on system size imposed by the amount of analog computing hardware available in a given installation. At present, the use of sensitivity analysis presented in Section III is executed off-line. In a hybrid facility with NASAP

capability, this repetitive procedure could be automated for on-line use.

4.4 References

1. Booton, R. C., "Nonlinear control systems with random inputs," IRE Trans. on Circuit Theory, vol. CT-1, pp. 9-18, March 1954.

2. Chuang, K. and L. F. Kazda, "A study of nonlinear systems with random inputs," Trans. AIEE, Applications and Industry, vol. 78, part II, pp. 100-105, 1959.

3. Ku, Y. H., "On nonlinear networks with random inputs," IRE Trans. on Circuit Theory, vol. CT-7, pp. 479-490, December 1960.

4. Pervozvanski, A. A., Random Processes in Nonlinear Control Systems, Academic Press, New York, 1965.

5. Price, R., "A useful theorem for nonlinear devices having Gaussian inputs," IRE Trans. on Information Theory, vol. IT-4, pp. 69-72, June, 1958.

6. Beck, C. H., R. L. Drake, and M. H. Kuo, "System modeling and structure optimization using hybrid computer techniques," Proceedings, Cornell Conference on Computerized Electronics, Ithaca, N. Y., August 26-28, 1969.

7. Beck, C. H., R. L. Drake, and M. H. Kuo, "A NASAP module for direct design of linear dynamic circuits using a hybrid computer optimization technique," Conference Record, Twelfth Midwest Symposium on Circuit Theory, pp. VIII 2.1-VIII 2.9, Austin, Texas, April 1969.

8. Narendra, K. S. and P. G. Gallman, "An iterative method for the identification of nonlinear systems using a Hammerstein model," IEEE Trans. on Automatic Control, vol. AC-11, pp. 546-550 1966.

9. Lion, P. M. "Rapid identification of linear and nonlinear systems," JACC, pp. 605-615, August 1966.

10. Happ, W. W., "Network analysis for systems applications program: present capabilities of a maintained program" IEEE Int. Conv. Record, pt. 5, pp. 64-88, March 1967.

11. Beck, C. H. and R. L. Drake, "A hybrid-NASAP module for direct design of linear dynamic circuits," IEEE Trans. on Education, vol. E-12, pp. 251-255, December 1969.

12. Hsia, T. C. and V. Vimolranich, "An on-line technique for system identification," IEEE Trans. on Automatic Control, vol. AC-14 no. 1, pp. 92-96, February 1969.

13. Kirvartis, K. and K. S. Fu, "Identification of nonlinear systems by stochastic approximation," JACC, pp. 255-264, August, 1966.

14. C. H. Beck, R. L. Drake, and M. H. Kuo, "A NASAP Module for Direct Design of Linear Dynamic Circuits Using a Hybrid Computer Optimization Technique," Conference Record, Twelfth Midwest Symposium on Circuit Theory, pp. VIII 2.1-VIII 2.9, Austin, Texas, April 21-22, 1969.

15. L. P. McNamee and H. Potash, "A User's Guide and Programmer's Manual for NASAP," Report No. 63-68, University of California, Los Angeles, 1968.

16. C. H. Beck, R. L. Drake, and M. H. Kuo, "A High Speed, Automated Solution for Physical Network or System Parameters Based on Sensitivity Analysis and Hybrid Computer Optimization Techniques," Conference Record, Thirteenth Midwest Symposium on Circuit Theory, Minneapolis, Minn., May 7-8, 1970.

17. C. H. Beck, R. L. Drake, and M. H. Kuo, "Direct Modeling of Nonlinear Systems Using Hybrid Computer Optimization Techniques," Conference Proceedings, Seventh Annual Allerton Conference on Circuit and System Theory, C-4, 6, Urbana, Illinois, October 8-10, 1969.

18. Angelo, E. J., Jr., J. Logan, and K. W. Sussman, "The separation technique: a method for simulating transistors to aid integrated circuit design," IEEE Transactions on Computers, vol. C-17, no. 2, pp. 113-116, February 1968.

19. Balaban, P., and J. Logan, "Analog Computer Simulation of Semi-Conductor Circuits," Proc. SJCC, 1968.

20. Raymond, Joseph H., Jr., Time Response Scaling of Junction Field Effect Transistors for Switching Applications, M.S. Thesis, Tulane University, New Orleans, Louisiana, 1970.