AN ABSTRACT OF THE THESIS OF

Waldo George Magnuson, Jr. (Name) for the Ph.D. (Degree)

in Electrical Engineering (Major)

Date thesis is presented April 14, 1966

Title CIRCUIT CHARACTERIZATION FOR EFFICIENT COMPUTER ANALYSIS

Abstract approved [redacted] (Major professor)

Analysis, coupled with an experimental investigation, is an important step in the design process of electronic circuits. Fortunately, with the development of computer programs to perform general network analysis, the job of analyzing relatively complex networks is not as formidable as it once was. This investigation develops guidelines for efficiently modeling active devices and circuits for time domain analysis using general network analysis computer programs. The amount of computer time required for large-signal transient analyses is investigated both analytically and experimentally in terms of the network parameters.

Three factors are found to be important in determining the amount of computer time that will be required in performing a computer analysis of a network. The factors

are: (1) The number of nodes the network contains, (2) the network time constants and network natural frequencies, and (3) the representation of the circuit elements comprising the network.

The application of guidelines based on these factors is directed primarily to large-signal transistor circuits but their application to the field of integrated circuits is also indicated. Suggestions for future work on general network analysis computer programs is suggested.

CIRCUIT CHARACTERIZATION FOR EFFICIENT COMPUTER ANALYSIS

by

WALDO GEORGE MAGNUSON, JR.

A THESIS

submitted to

OREGON STATE UNIVERSITY

in partial fulfillment of
the requirements for the
degree of

DOCTOR OF PHILOSOPHY

June 1966

APPROVED:

[redacted]

Assistant Professor of Electrical Engineering
In Charge of Major

[redacted]

Head of Department of Electrical and
Electronics Engineering

[redacted]

Dean of Graduate School

Date thesis is presented April 14, 1966

Typed by Erma McClanathan

ACKNOWLEDGMENT

It is a pleasure to acknowledge the assistance of two institutions in the course of this study. The first is the Electrical Engineering Department at Oregon State University for permitting me to complete my thesis off the university campus. In particular, to Professors J. C. Looney and L. N. Stone at O.S.U. for their guidance and encouragement. The second is the Electronics Engineering Department at the Lawrence Radiation Laboratory, Livermore, California for providing time, the environment, and financial support. The help of Messrs. E. H. Hulse and A. J. Stripeika at L.R.L. is gratefully acknowledged. Special appreciation goes to my wife, Barbie, for her continued encouragement and support.

TABLE OF CONTENTS

TABLE OF CONTENTS (cont.)

LIST OF FIGURES

LIST OF TABLES

CIRCUIT CHARACTERIZATION FOR EFFICIENT COMPUTER ANALYSIS

I. INTRODUCTION

The Design Process

As engineering systems become increasingly complicated, the designer must rely on more sophisticated design techniques. The design process for a system or for a subset of a system may be characterized by a simple block diagram such as appears in Figure 1.

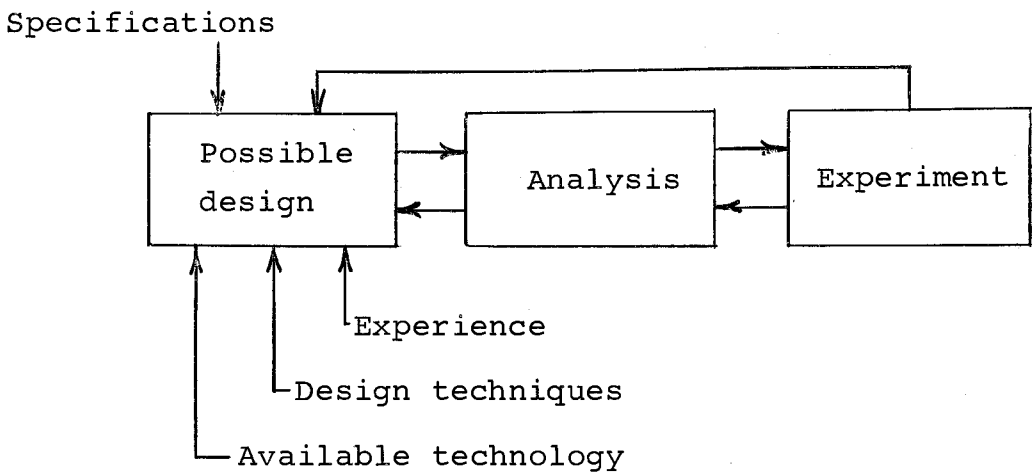


Figure 1. The design process.

Using available technology, design techniques, and relying on his experience the designer will formulate a possible design to meet the necessary specifications. This design will generally next be analyzed to find out if the proposed design has a reasonable chance of

operating. If so, then generally, for electronic circuit design, the next step is to construct a breadboard model and perform laboratory tests on the design. There is a feedback at the analysis stage and at the experimental stage until the final design is achieved.

This paper will focus on the center block of Figure 1, that is, the analysis phase of engineering design. In particular, it will be concerned with digital computer aided time-domain analysis, and specifically with investigating the efficient modeling of active devices for transient analysis.

The Need for Computer Analysis

A general purpose network analysis computer program offers the electronic circuit designer a direct means of studying the behavior of circuits. A designer may have a variety of reasons for analyzing a network; to find the output response for a given input, to increase the understanding of the circuit operation, to optimize the circuit performance, or to study circuit reliability.

The potential value of a digital computer network analysis program is soon evident when even a relatively simple circuit is analyzed by conventional numerical or transform techniques. For example even a simple two-stage amplifier may contain over ten nodes and ten loops.

An analysis of such a circuit would require the evaluation of at least a tenth order determinant and the factorization of a tenth order polynomial. An exact analytical solution for the response quickly becomes nearly impossible or even irretractible.

Evidently then, to obtain an analytical solution, the design engineer is left to three alternatives: to perform a long tedious attack on the problem with conventional techniques, resort to the use of gross simplifications and approximations, or to use a computer network analysis program. Alternatively a breadboard design approach may be taken entirely, omitting the analysis phase, but when a circuit is developed entirely experimentally the resulting design may be far from optimum in terms of dc stability, switching speeds, reliability, or other characteristics.

Computer analysis permits the designer to analytically verify or discount laboratory observed response. It permits a one-parameter-at-a-time control over the response, and used properly, can lead to insight of the circuit operation.

Plan and Scope of Presentation

This study will investigate the characterization of active devices for large-signal transient analysis. In

particular, linear and non-linear transistor models will be investigated in regard to their effect on program solution times while retaining a desired degree of accuracy.

The plan of presentation will be to first discuss transistor modeling, presenting static and dynamic models. Although the emphasis is on transistor equivalent circuits the final results may easily be extended to include other devices; i.e., diodes, FETs, SCRs, etc., and other construction techniques; i.e., integrated-circuits.

Following the chapter on transistor modeling will be a chapter on general network analysis. To phrase the network analysis problem in general terms, the state-variable approach to network analysis is used. The system of network equations is derived, the elimination procedure carried out, and the solution procedure given.

The fourth chapter investigates modeling and computer running times. Findings from these studies are related to the analytical development of network equations and in the final chapter general guidelines are summarized and presented. Primarily these guidelines are directed towards improving the speed of the transient solutions.

Relationship to Previous Work

To relate the material in this report to previous work it is necessary to consider previous efforts in three areas: Transistor modeling, general network analysis, and computer network analysis programs. This division serves three objectives; first it is a convenient partition of earlier work, it provides orientation within each area, and it is basically the way the following material is presented.

Similarities and differences of the present work to previous efforts will not be mentioned in this section but will be commented upon in the appropriate chapters. The object here is to provide some measure of perspective to earlier work and consequently the work of only a few authors is presented. It is not the object to present a comprehensive bibliography of all previous work, just a sampling of a few of the more important contributions.

Transistor Modeling

Transistor modeling has had an illustrious history, dating to the late 1940's and early 1950's. It is natural that device modeling has been important from the first days of transistors since a study and use of models of the electronic processes is necessary for the understanding and development of electronic circuits.

Perhaps the classic article on large-signal junction transistor models is that of Ebers and Moll (12). Ebers and Moll characterized the static behavior of the transistor in terms of network elements and their approach has found wide acceptance, particularly for those people closer to the laboratory design phase.

In 1957 Beaufoy and Sparkes (4) introduced a model slightly different from, but entirely equivalent to, that proposed by Ebers and Moll. In their model, Beaufoy and Sparkes relate the collector current to the charge in the base region of the transistor. Their concept considers the transistor as a charge controlled device as opposed to a current controlled device. The two views differ only in their description of the external behavior and both lead to exactly equivalent results as measured at the external terminals.

In an effort to correlate the physics of transistor action to modeling theory, Linvill in 1958 took a different approach and presented what has become known as the lumped model (19). In the lumped model attention is focused on minority carrier flow in the semiconductor and the processes of recombination, storage, diffusive flow, and drift flow are modeled directly. While the lumped model describes the physical processes of the device quite closely it is not entirely convenient as

an analysis model, particularly with computer programs where often element types are restricted to resistors, capacitors, and inductors.

These three modeling techniques are summarized and compared in an article by Lindholm, Hamilton, and Naurd (15). While a considerable amount of effort has been done on large-signal modeling and although numerous articles have appeared the work has, for the most part, been related to one of the above three modeling techniques.

Small-signal transistor modeling has followed two general lines: A four-terminal representation using primarily h or y parameters, and incremental models based on describing the physics of the device by RLC network elements. The second method generally uses the hybrid-Pi or Tee equivalent circuit representation. A good survey of the earlier work in small-signal modeling has been given by Pritchard (24). A more recent presentation of the modeling process is given in Volume 3 of the SEEC series of volumes (27, p. 81-119).

General Network Analysis

Like any general analytical approach, general network analysis places a burden on the analyst in terms of numerical or symbolic manipulation. However, the computer has influenced the area of analysis to make a

general approach to network analysis feasible and highly worthwhile.

A good account of the approach to general network analysis is given in Seshu and Balabanian's book Linear Network Analysis (28, p. 77-142) and in Seshu and Reed's Linear Graphs and Electrical Networks (29, p. 117-152). The starting point for a general network analysis is the three basic relationships; Kirchhoff's voltage law, Kirchhoff's current law, and the branch voltage-current relationships or Ohm's law for linear elements. The classical approach does not always guarantee a set of independent network equations, consequently a state space approach is used.

Bashkow (2) first formulated the network analysis problem in terms of state variables. Bryant (9) and Brown (7) extended this work and recently a good tutorial article by Kuh and Rohrer (18) on the subject has appeared. The state variable application to network analysis has been the impetus to develop several general purpose digital computer analysis programs.

Computer Analysis Programs

For several years now, there has been considerable activity in the area of network analysis using digital computers. Generally it is possible to subdivide the programs written to this end into two groups:

(1) Those programs where the attention has been focused upon a particular problem to be solved and (2) a second group where the effort has been directed towards creating a program more general in nature, i.e., capable of analyzing more than a single problem of given topology. The first group is of limited use and will not be considered further here.

Computer programs of a more general nature can be classified as time domain or frequency domain analysis programs; linear or nonlinear analysis programs; statistical programs; and a host of other subdivisions. In the specific area of large-signal time domain analysis programs, perhaps the IBM Corporation has been one of the more active in program development. Domenico (11) in 1957 described a program capable of performing large-signal analysis. Branin (5), in 1962, discussed a further development of the program, capable of performing DC and transient analysis. In 1964 a short summary of then available programs was given by Falk (13).

Two of the more recent programs tailored specifically to large-signal transient analysis of electronic circuits are the NET-1 (Network Analysis Program) program (20) and the PREDICT program (32). The PREDICT circuit analysis program is described and used later in this report.

Summary of Results

In chapters three and four it will be shown that three factors are important in determining the amount of computer time that will be required in performing an analysis of a network. The factors are: (1) The size of the network, (2) the network time constants and the network natural periods, and (3) the representation of the parameters making up the network. Chapter five reduces these findings to general guidelines in modeling active devices for large-signal transient analysis by general network analysis computer programs.

II. TRANSISTOR MODELING

Introduction

To obtain meaningful results from a network analysis, passive and active linear or non-linear devices must be represented to an acceptable degree of approximation. Furthermore the model chosen should be as simple as possible, should present to the design engineer a readily interpretable equivalent circuit, and yet be sufficiently accurate in describing the physical device to predict the circuit performance.

Basically, in this chapter, we are interested in modeling transistors for large-signal transient analysis and in particular, characterizing transistors for efficient analysis by general network analysis computer programs. Although there are practically an unlimited number of large-signal models that may be used, three modeling methods have received much attention in the literature. They are:

1. Ebers and Moll large-signal model (12),
2. Beaufoy and Sparkes charge-control model (4),
3. Linvill lumped model (19).

All three of these models predict equally well the large-signal behavior of transistors but differ in their representation of the physical processes. The Ebers and

Moll and the Linvill models can be extended to include non-linear behavior of the junctions more conveniently than the charge-control model. The Linvill lumped model represents more closely the physical processes of the three methods and, when the interest is in relating device behavior to device physics, it has distinct advantages.

Generally, however, transient circuit analysis computer programs are developed to predict the network response in terms of element values and externally measurable device parameters. The programs are circuit design and not device design oriented and consequently the object is not to describe the physical processes of device operation but to compute the circuit response. Consequently the Ebers and Moll and the charge-control models are more useful for most computer analysis programs.

The mathematical model common to all three models is a set of six basic equations for the base region, together with a description of the junction conditions. The six equations deal with drift and diffusion currents, continuity of charge, impurity densities, and the electric field in the base region (15). The equations describing each model are partial differential equations and each model effectively removes the space variable by

appropriate approximations.

The approach to the characterization of transistors in this chapter will be the development of a composite model from a combination of a small-signal hybrid-Pi equivalent circuit and the Ebers and Moll model. The development of the small-signal equivalent circuit, the static model, and the dynamic model in the following sections do not produce a model that is new or different from those of previous efforts but is original in the presentation. The progressive development from a small-signal model to the large-signal dynamic model is different from earlier modeling in that it is not restrictive to either small or large-signal regions but includes both small-signal and large-signal considerations.

As was stated earlier, the object of this study is to provide a foundation for the efficient modeling of active devices, in particular transistors, for use in circuit analysis using digital computer programs. To this end, the discussion of a transistor equivalent circuit will, in addition to developing a suitable model, discuss non-linear element representation. In particular, the development will be directed towards application to computer network analysis.

Transistor Models

One of the first steps in the analysis of any electrical network containing active devices is to describe the active devices by appropriate equivalent circuits. This generally consists of representing the device by RLC circuit elements and controlled voltage or current sources. The elements may or may not be linear.

Several considerations must be taken into account when representing an active device by an equivalent circuit. Among these considerations are:

1. Operating region. The device may perform only over a small range in voltages and currents, in which case, usually the parameters in the model may be regarded as approximately linear. Otherwise the non-linear nature of the device may require non-linear parameter representation.
2. Frequency range. When the analysis concerns itself only within a limited frequency range additional simplifications may be made in representing the device.
3. Accuracy. Generally in any analysis the degree of accuracy required in the results will greatly affect the complexity of the equivalent circuit used in the modeling.
4. Measurability of parameters. Not only must

numerical values for the circuit elements in the model be determined but their dependency on electrical and physical conditions must be found.

5. Efficiency in use. This will be taken to mean the relationship between computer solution times using a general time-domain network analysis program and the equivalent circuit parameter representation for a given solution accuracy.

The transistor equivalent circuit to be presented in this section for later use will be developed in three phases. These phases will consist of: (1) Small-signal equivalent circuit, (2) a static model, and (3) a transient analysis or dynamic model. The small-signal and static models will be presented separately and then combined into one equivalent circuit which will be used to predict large-signal behavior.

Small-signal Equivalent Circuit

The development of a small-signal equivalent circuit for the transistor will be accomplished in this section in three steps. Step one will consist of considering an idealized small-signal circuit model for the transistor. Then this will be extended to a Pi-configuration which contains first-order approximations. Finally second-order approximations will be discussed and added to the model of step two to characterize the transistor by means

of the well-known hybrid-Pi equivalent circuit. Throughout, the development will be in terms of a common emitter configuration.

This procedure is justified in that it describes the usual steps in the design process for electronic circuits. That is, in the initial stages of design, gross idealizations are made in the behavior of active devices and in the latter stages more detailed and precise models are used.

An idealization of the transistor in the common-emitter configuration is shown in Figure 2.

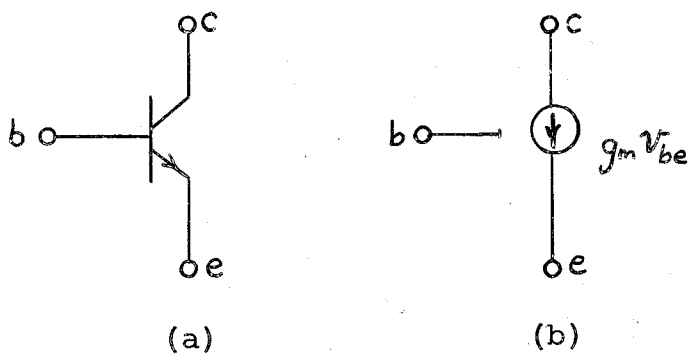


Figure 2. (a) Common-emitter transistor orientation and (b) idealized functional equivalent circuit.

Of course the infinite input and output impedances are gross approximations but the idealized model does present the common-emitter stage as a transconductance applifier which is quite correct. To this model we

now account for what might be called first-order effects.

With the emitter junction forward biased and the collector junction reverse biased, excess minority carriers are injected into the base region by the emitter. These carriers are transported across the base to the collector junction by drift and diffusion. At the collector junction they are swept into the collector body by the electric field in the space charge layer.

In refining the idealized model the influence from two circuit disturbances will be considered:

1) Effects of a small increment in the forward biased emitter junction, and
2) Effects of variations in the collector-to-emitter voltage.

To a good approximation, the distribution of excess minority carrier concentration in the base region can be assumed linear. Consequently an increase in the emitter-base voltage will result in a transient increase of base current to maintain charge neutrality within the base region. In addition, due to the higher concentration of minority carriers, recombination is higher in the base region and there will be a resulting increase in the constant component of the base current. Finally, as a result of the increased forward bias on the emitter there will be an increase in the concentration gradient

of minority carriers which, in turn, will produce an increase in collector current.

The increase in collector current is related to ΔV_{BE} by a proportionality constant called the tran-conductance and is designated by g_m. The relationship between g_m and I_c is (14, p. 134)

$$g_m = \frac{q}{kT} |I_c| \tag{1}$$

where $|I_c|$ indicates the magnitude of the collector current.

The total small-signal base current is

$$i_b = g_b v_{be} + C_b \frac{dv_{be}}{dt} \tag{2}$$

where the first term on the right side of the equation is due to the recombination and the second term to provide charge neutrality to a small increase in V_{BE} (or ΔV_{BE} or, because we are considering small signals, v_{be}).

Equations (1) and (2) can be used to modify the idealized model of Figure 2 (b).

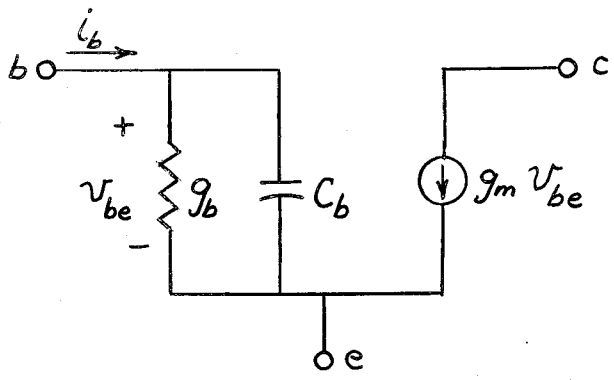


Figure 3. First-order small-signal equivalent circuit.

The equivalent circuit of Figure 3 has a simple expression for its voltage gain. It is

$$A_v = \frac{v_{ce}}{v_{be}} = \frac{-g_m v_{be} R_L}{v_{be}} = -g_m R_L \qquad (3)$$

This equation implies that the voltage gain is limited only by the magnitude of R_L and as R_L becomes exceedingly large so does the small-signal voltage gain. However, second-order effects come into play to limit the gain.

The width of collector-base space-charge layer depends on the magnitude of the collector-base voltage. For a positive increment in v_{cb} the width of the space-charger layer decreases, causing an incremental increase in the base width.

The increase in the base width has two effects (14):

(1) The collector current decreases because the slope of the minority carrier distribution decreases, and

(2) The base current increases because of the stored base charge and consequently the recombination increase.

These two effects may be incorporated in the model by adding two resistors and one capacitor. Because the change in base width with a change in v_{cb} is very small the resistors will be very large and the capacitor very small. These elements have been shown in Figure 4 as $r_{cb'}$, r_{ce}, and $C_{cb'}$.

Both the collector-base and emitter-base junctions have space-charge capacitances due to the depletion layer of the junctions. These capacitors are also voltage dependent, but the model being considered here applies only to small-signals at a constant quiescent operating point so these capacitors will be taken as constant and have been indicated in Figure 4 by C_{Tc} and C_{Te}. Their dependency on voltage will be discussed in the section on large-signal models.

One final element must be included in Figure 4. It is the base resistance, or sometimes referred to as the base spreading resistance or extrinsic base resistance. This resistance accounts for the voltage drop in the

base region by the flow of majority carriers. The minor-carriers flow from emitter to base and the majority-carrier base current flows at right angles to the minority current. This majority-current must flow through the inactive region of the base near the base contact and can be simulated by a resistor in series with an internal base lead.

The small-signal equivalent circuit shown in Figure 4 is commonly called the hybrid-Pi equivalent circuit.

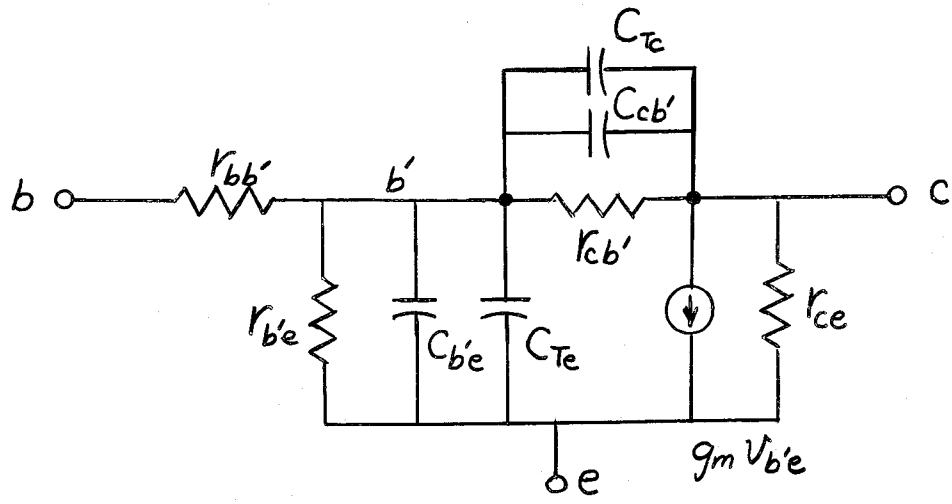


Figure 4. Small-signal transistor equivalent circuit.

The circuit in Figure 4 has been redrawn in Figure 5 with the parallel capacitors combined and $r_{cb'}$ omitted. It is only at quite low frequencies that $r_{cb'}$ becomes

important compared to the effects of $C_{cb'}$ and for this reason it is generally omitted in analyses.

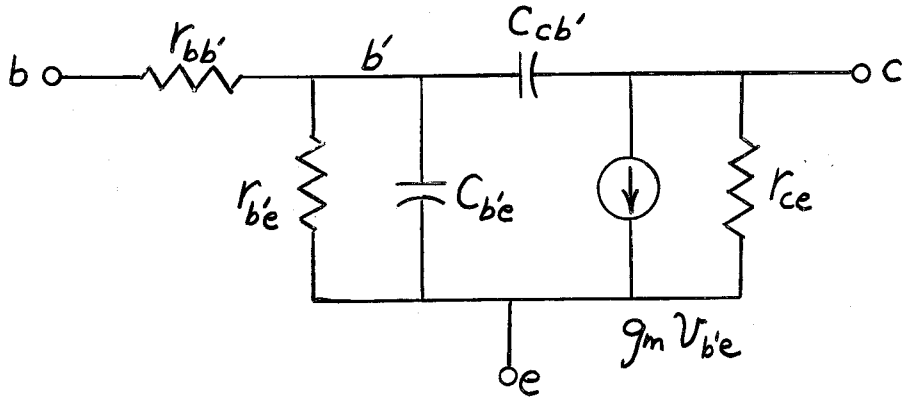


Figure 5. Hybrid-Pi equivalent circuit for the transistor.

Obviously the hybrid-Pi model for the transistor does not exhaust the possibilities for model configurations. However, this equivalent circuit has been shown to faithfully describe the small-signal behavior at all frequencies at which the device has reasonable gain. And for this reason it will be used in chapter four in investigating effects on computer solution times.

Static Transistor Model

In modeling transistors for large-signal transient analysis a convenient approach is to consider operation in two domains; the first being operation in the active region and characterized by a small signal model and the

other, operation due to non-linearities of the junctions and modeled by a static equivalent circuit. This section will consider the static behavior of junction transistors.

On the basis of linear diffusion in the absence of drift fields the diffusion equation

$$D_p \nabla^2 p - \frac{p-p_o}{\tau} = \frac{\partial p}{\partial t} \tag{4}$$

may be used to derive general analytical expressions for the p-n junction transistor. From these equations the gross non-linear behavior of the transistor can be determined. Ebers and Moll (12) have carried out this derivation and show that the equations describing a generalized junction transistor have the form

$$I_E = a_{11}\left(e^{\frac{q\phi_E}{kT}} - 1\right) + a_{12}\left(e^{\frac{q\phi_c}{kT}} - 1\right) \tag{a}$$

and (5)

$$I_c = a_{21}\left(e^{\frac{q\phi_E}{kT}} - 1\right) + a_{22}\left(e^{\frac{q\phi_c}{kT}} - 1\right) \tag{b}$$

where the currents and voltages are defined by Figure 6.

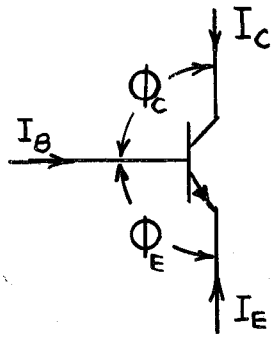


Figure 6. Voltage and current conventions.

The quantities ϕ_E and ϕ_C are the junction voltages and have the positive sense for a voltage drop from p to n material. Several assumptions were made by Ebers and Moll in deriving Equations (5 a,b). The most important are:

1. The resistivities of the semiconductor regions are low.
2. Injected current densities are low (linear diffusion equation).
3. Base modulation effects are negligible.
4. The emitter and collector junctions can be represented by an ideal diode.

The ideal diode of assumption 4 is a diode which has terminal relations described by

$$I = I_s \left(e^{\frac{qV}{kT}} - 1 \right) \qquad (6)$$

where $q/kT \simeq 0.026$ volts^{-1} at 25°C and I_s is the reverse

saturation current.

The coefficients a_{11}, a_{12}, a_{21}, and a_{22} are easily related to the four transistor parameters

I_{Eo} = emitter saturation current with $I_c = o$

I_{co} = collector saturation current with $I_E = o$

α_n = normal current gain

α_i = inverted current gain

which are easily measurable. When this is done, Equations (5 a,b) become

$$I_E = -\frac{I_{co}}{1-\alpha_n \alpha_i}\left(e^{\frac{q\phi_E}{kT}}-1\right) + \frac{\alpha_i I_{co}}{1-\alpha_n \alpha_i}\left(e^{\frac{q\phi_c}{kT}}-1\right) \qquad (7a)$$

$$I_c = \frac{\alpha_n I_{Eo}}{1-\alpha_n \alpha_i}\left(e^{\frac{q\phi_E}{kT}}-1\right) - \frac{I_{co}}{1-\alpha_n \alpha_i}\left(e^{\frac{q\phi_c}{kT}}-1\right). \qquad (7b)$$

These equations describe the static behavior of junction transistors provided the assumptions are fulfilled. The denominator term $1-\alpha_n \alpha_i$ in Equations (7a,b) may be eliminated by using the short-circuit saturated currents I_{cs} and I_{Es} where $I_{Eo} = I_{Es}(1-\alpha_n \alpha_i)$ and $I_{co} = I_{cs}(1-\alpha_n \alpha_i)$. Then Equations (7a,b) become

$$I_E = -I_{Es}\left(e^{\frac{q\phi_E}{kT}}-1\right) + \alpha_i I_{cs}\left(e^{\frac{q\phi_c}{kT}}-1\right) \qquad (8a)$$

$$I_c = \alpha_n I_{Es}\left(e^{\frac{q\phi_E}{kT}}-1\right) - I_{cs}\left(e^{\frac{q\phi_c}{kT}}-1\right). \qquad (8b)$$

These equations lead to the equivalent circuit in Figure 7(a). To more explicitly represent the diode nature of the junctions, the equivalent circuit of Figure 7(a) has been redrawn in Figure 7(b) where now, I_N and I_I are the forward (normal) and reverse (inverted) currents flowing through the ideal emitter and collector diodes respectively.

Several additional elements can be added to the model of Figure 7(b) to construct a model which more closely predicts actual device behavior. For example, although Equation (6) accurately describes the characteristics of an actual diode at small forward and reverse biases, deviations from the theoretical non-linear behavior occur at larger biases. These non-linearities may be very nearly accounted for over a wide range of variables by the addition of a small conductance in parallel with the diode to account for the finite slope of the reverse characteristic, and by the addition of a small series resistance which accounts for the voltage drop in the bulk semiconductor material. A characterization including these additional elements is shown in Figure 7(c).

The degree of complexity of the equivalent circuit used, of course, will depend on several considerations. Among these considerations are; (1) the accuracy

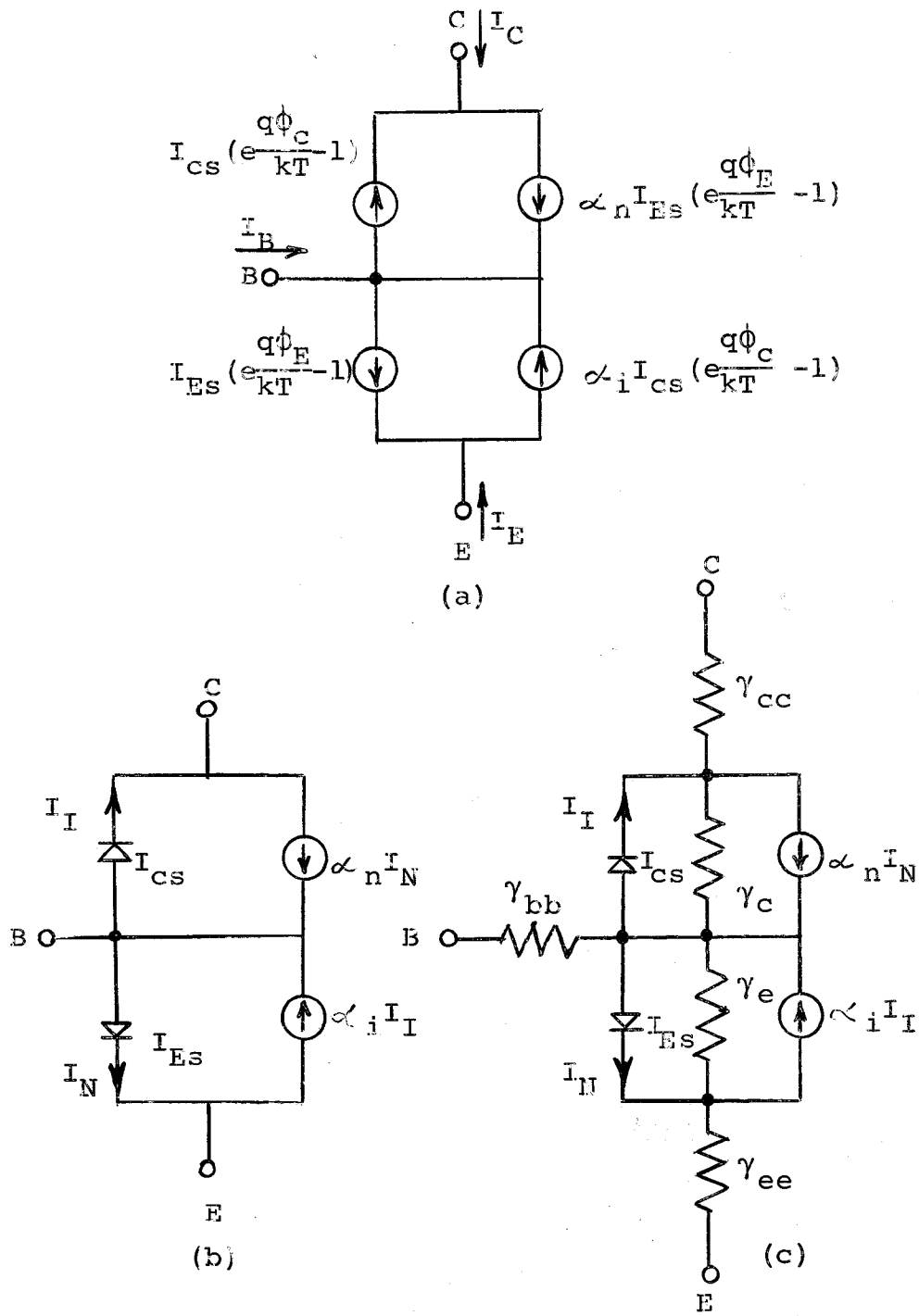


Figure 7. Ebers and Moll large-signal model for junction transistors.

desired in the final solution, (2) how well known or how easily measured the equivalent circuit parameters are, and (3) any restrictions on computer solution time available.

Dynamic Transistor Model

The dynamic behavior of transistors can be described by considering it to be a current-controlled device or by describing the behavior in terms of the density of minority carriers in the base region. The second method was introduced by Beaufoy and Sparkes in 1957 (4) and is generally referred to as charge-control operation.

The approach to be taken in this section in representing the transistor for dynamic operation will be to merge the small-signal Tee equivalent circuit and the large-signal static model to form a composite model. The hybrid-Pi equivalent circuit developed earlier may be easily transformed to the Tee circuit in the following way: First the voltage-controlled current generator is split into two generators -- one across $r_{b'e}$ and the other across $C_{b'c}$; the circuit is then as shown in Figure 8(a). The parallel combination of $r_{b'e}$ and $g_m V_{b'e}$ may be merged into a single conductance of value $g_m + 1/r_{b'e}$ which will be designated as r_e.

When the circuit of Figure 8(b) is compared with the circuit of Figure 7(b) several similarities may be

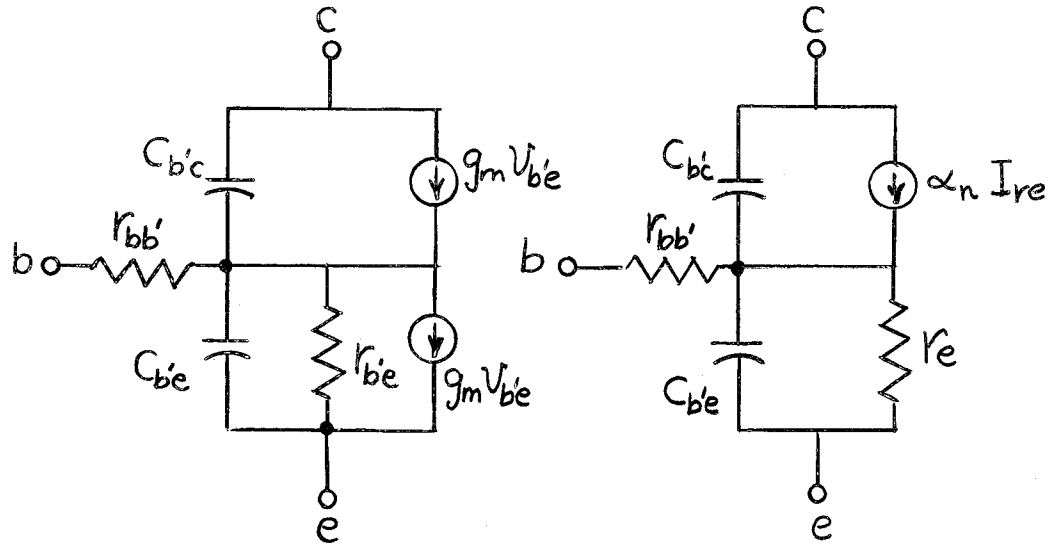


Figure 8. Development of a large-signal model.

noted: They are both Tee connections, the current generators in the collector circuits are both α_n times the current through the emitter diode. The current generators $I_{cs}\left(e^{\frac{q\phi_c}{kT}}-1\right)$ and $\alpha_i I_{cs}\left(e^{\frac{q\phi_c}{kT}}-1\right)$ do not appear in the second model because the diode nature of the collector junction was not considered in small-signal operation and because the term $(1+\alpha_n)I_{Eo}/(1-\alpha_n\alpha_i)$ was assumed small in comparison to I_E. However, to provide for operation in the inverted mode with the dynamic model, these terms will be retained. The dynamic model then becomes as indicated in Figure 9.

Resistors r_{ee}, r_e, r_c, and r_{cc} have already been mentioned. Capacitors C_e and C_c each actually consist

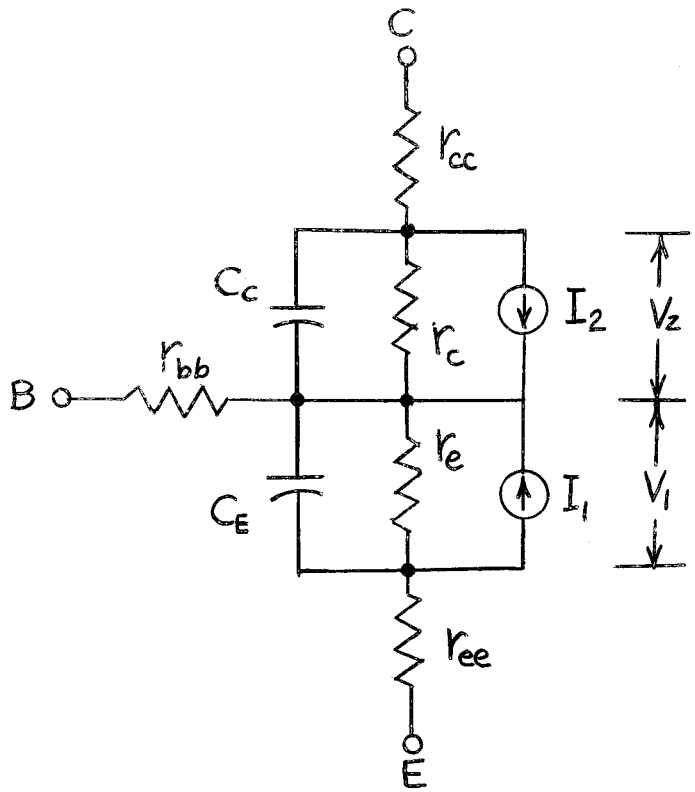


Figure 9. Dynamic model for large-signal analysis.

of two components, one due to the junction transition regions C_T and the other due to charge carriers in the base region C_D.

Transition capacitance, or space charge layer capacitance, is somewhat analogous to a parallel plate capacitor in that charges of opposite polarity are separated by a narrow region depleted of charge. Both the collector and emitter junction transition capacitances have the form (23, p. 108)

$$C_T = \frac{C_o}{(V_z - V)^n} \tag{9}$$

where C_o is evaluated by measuring C_T at a particular

value of V, V_z is the contact potential of the transistor material, and n is a constant between $\frac{1}{3}$ (graded junction) and $\frac{1}{2}$ (step junction). Equation (9) applies to both junctions although in general C_o and n will not be identical for the two junctions.

The emitter diffusion capacitance C_{De} arises from the transit time and dispersion of minority carriers across the base region. In the small-signal case the transit time T_e is related to the emitter diffusion capacitance by

$$T_e = C_{De}\, r_e \tag{10}$$

where r_e is the small-signal diode resistance. And for common-base operation it is further related to the radian α cutoff frequency through the relation

$$T_e = 1/\omega_\alpha \ . \tag{11}$$

In the large-signal case r_e is not a constant so C_{De} is given by

$$C_{De} = \frac{T_e}{\frac{\partial V_1}{\partial I_1}} \ . \tag{12}$$

Ashar et al. have suggested that to more closely represent the excess phase shift in the transport factor of modern diffused-junction transistors a delay factor be added in the single pole approximation for α. This is accomplished by using the effective base transit time rather than f_α or f_T (1).

In a similar way the collector diffusion capacitance may be found. This capacitance accounts for the storage time in normal operation and will be designated by C_s. It is given by

$$C_s = \frac{T_s}{\frac{\partial V_2}{\partial I_2}} \tag{13}$$

The storage (or saturation) time constant T_s can be obtained on the basis of charge-control equations (4) or from pulse measurements (1).

The elements r_{bb}, r_{ee}, and r_{cc} are due to the bulk semiconductor material and are generally represented by a constant value of resistance.

Equivalent-Circuit Element Representation

In the small-signal model excursions of the signal levels are assumed sufficiently small so that all elements can be assumed to be constant. This situation is altered drastically for the large-signal dynamic model. In this case the elements are explicit functions of the network voltages and currents. For example, the capacitor C_e of Figure 9 has the form

$$C_e = C_{Te} + C_{De}$$

$$= \frac{Co}{(V_z - V_1)^n} + \frac{Te}{\partial V_1 / \partial I_1}$$

$$= \frac{Co}{(V_z - V_1)^n} + \frac{q}{kT} \; T_e I_{Es} \; e^{\frac{qV_1}{kT}}$$

$$\simeq \frac{Co}{(V_z - V_1)^n} + \frac{q}{kT} \; \frac{I_{es}}{\omega_\alpha} \; e^{\frac{qV_1}{kT}} \qquad (14)$$

However, on specification sheets, C_{Te} is generally given by a plot of capacitance versus reverse bias voltage. Capacitor C_{De} must be computed from I_{es} (or I_{eo}) and f_α (or f_T). Generally, the junction capacitances are not given for forward-biases, but are obtained by extrapolation into the forward-bias region.

Generally the emitter and collector static characteristics are represented by curves of current versus voltage on device specification sheets, and these curves may be used in an analysis or the parameters for the analytic representation may be found from the curves. In the same manner the extrinsic base resistance R_{bb} changes with current level and to accurately represent the transient network response it may be necessary to represent it as a non-linear element in the analysis. A similar requirement might be necessary for R_{cc}, the saturation resistance.

Application to Computer Analysis

As pointed out in the last section there are a number of ways in which the electrical behavior of an

element may be described. However, in terms of network analysis by computer programs, these different representations can be grouped into four classifications:

1. A numerical constant,
2. a table,
3. an analytic expression,
4. or a combination of the above.

In performing an analysis, not all of the above four element representations will be equivalent in terms of program efficiency. For example, a table look-up will obviously require more time than extracting a single constant value for an unknown. Furthermore, to evaluate an analytic expression may require a number of lengthy program operations compared to a simple table look-up procedure.

In the solution for the network response several operations are involved in solving the algebraic matrix equations. The operations may require being done only once in a given problem or may be required for each solution step. The particular representation of a network element can have an influence on the solution process. The next chapter will show that variable R, L, and C elements force additional program operations at each solution time step. When other means can be found for representing non-linear passive elements the

solution will require less computer time. Various methods for accomplishing this will be discussed in the fourth chapter.

III. GENERAL NETWORK ANALYSIS

Network Analysis and Computer Solutions

The derivation of a general system of loop and node equations to be used in general network analysis has three purposes (29, p. 142). First it serves as a basis for the development of network theory, secondly it can always be used to provide the "right answer" whenever there is disagreement between "observed" and answers computed using simplifying approximations, and finally a general approach to the solution for the network behavior is necessary when solving problems using digital computers. That is to say, the development of a general network analysis computer program must be based on the framework of a mathematical description of the general loop and node system of equations. Otherwise the program will be reduced to solving special cases and would consequently be of limited usefulness.

It is the general approach to the formulation of network analysis that will be considered in this chapter. Within the context of this chapter, general network analysis will be defined as: The formulation and solution of a set of simultaneous independent equations which describe the response of an electrical network. Furthermore, the electrical network must satisfy Kirchhoff's

current and voltage laws and Ohm's law.

The description will be concerned initially with lumped linear networks and formalize the development of the network equations in matrix notation. The problem of writing the descriptive equations for a network, aside from solving them, can be in itself difficult, so to be most useful, a digital computer program for general network analysis should include an equation complier. That is to say, the program should be capable of deriving the descriptive network equations from a topological description of the network as input to the program. For that reason this chapter will also consider the question of choosing an appropriate set of independent network equations from the input data. Again the preference will be given to a method that is completely general and which always "works."

The organization of this chapter will begin with a description of the method for obtaining a set of simultaneous ordinary differential equations describing a general network. Following that, the question of choosing the set of network meshes will be covered and then the numerical procedure for solving the network equations will be described. The chapter will terminate by considering a few special programming techniques used in implementing a network analysis program on a computer.

Throughout, specific relations will be derived relating to the speed of problem solution to the operations involved in obtaining the solution. These relationships will be summarized at the end of the chapter.

It should be noted where the content and objective of this chapter differ from that of previous work. First the object here is to develop the necessary equations for the general approach to network analysis. This has been done before (28, p. 77-142) but not to the extent of presenting the complete elimination procedure as has been done in this chapter. Furthermore, the object in this chapter has not only been to develop the equations describing a general analysis but to express them in a manner that can be related to computer solution time. The sections on solution procedure and numerical integration are related more directly to the network analysis problem than previous reports. In particular, an effort to relate the details of a general analysis to the problem of excessive solution times has not appeared elsewhere.

General Loop and Node Equations

As mentioned in the introduction, the solution for the response of a network is based on solving a set of loop and node equations. This set of equations, in turn,

must satisfy Kirchhoff's current and voltage laws and Ohm's law. So we will begin by expressing these laws in matrix notation.

Kirchhoff's current law can be written as

$$\sum_{j=1}^{N_b} a_{k_j} i_j(t) = 0 \quad k = 1,2,\cdots N_v \tag{15}$$

where $a_{k_j} = \begin{cases} +1 \\ 0 \text{ if the j-th branch is } \begin{cases} \text{positively} \\ \text{not} \\ \text{negatively} \end{cases} \\ -1 \end{cases}$

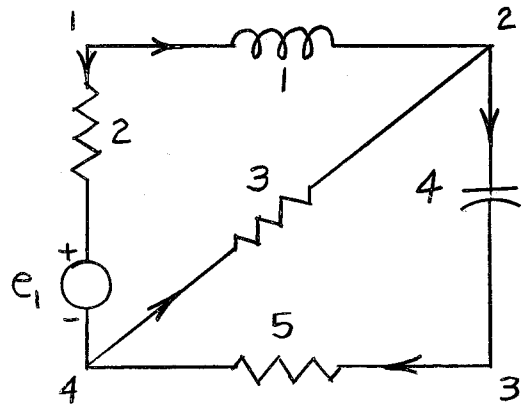
incident on the k-th node and N_v is the number of circuit nodes. This equation may be rewritten in matrix notation as

$$A_a i_b(t) = 0 \tag{16}$$

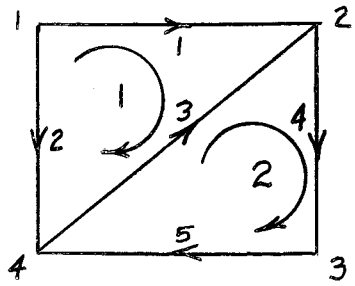
where $A_a = \left|a_{k_j}\right|$ and is of order (N_v, N_b) and $i_b(t)$ is a column matrix of N_b (number of branches) rows. The sign of a_{k_j} will be taken as positive when the current is directed away from node k (positively incidence).

As an example, the Kirchhoff current law equations for the network in Figure 10 are

$$
\begin{array}{c}
\text{branches} \longrightarrow \\
\text{nodes} \downarrow
\end{array}
\left|\begin{array}{ccccc}
1 & 1 & 0 & 0 & 0 \\
-1 & 0 & -1 & 1 & 0 \\
0 & 0 & 0 & -1 & 1 \\
0 & -1 & 1 & 0 & -1
\end{array}\right|
\left\|\begin{array}{c}
i_{b1}(t) \\
i_{b2}(t) \\
i_{b3}(t) \\
i_{b4}(t) \\
i_{b5}(t)
\end{array}\right\|
=
\left|\begin{array}{c}
0 \\
0 \\
0 \\
0 \\
0
\end{array}\right| .
$$



(a) Electrical circuit



(b) Linear graph of (a)

Figure 10. Example of a network.

The matrix A_a is known as the augmented branch-node incidence matrix. Notice each column contains a +1 and a -1 and the sum of the elements in each column is zero. Any one row may be expressed as the negative sum of the other rows or, in other words, the rows are linearly

dependent. We may consider any one row as redundant and delete it leaving N_v-1 rows. The node corresponding to the deleted row is called the datum or reference node (6). This is equivalent to saying that exactly N_v-1 of the equations in Equation (15) are linearly independent.

Then we can write Kirchhoff's current law as

$$A\, i_b(t) = 0 \tag{17}$$

in which A is called the reduced incidence matrix and corresponds in our example (deleting the third row) to

$$\begin{vmatrix} 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & -1 & 1 & 0 \\ 0 & -1 & 1 & 0 & -1 \end{vmatrix} \begin{vmatrix} i_{b1}(t) \\ \vdots \\ i_{b5}(t) \end{vmatrix} = \begin{vmatrix} 0 \\ \vdots \\ 0 \end{vmatrix} .$$

Next we turn to Kirchhoff's voltage law which can be stated as:

$$\sum_{j=1}^{N_b} b_{k_j}\, v_j(t) = 0 \qquad k = 1,2,\cdots N_m \tag{18}$$

where the coefficients b_{k_j} have the values $\pm 1,0$ depending on the voltage reference in the k-th loop. The term b_{k_j} is unity when branch j is in loop k and positive when the voltage reference is at the tail of the loop orientation arrow. Kirchhoff's voltage law written in matrix notation is:

$$B_a\, v_b(t) = 0 \tag{19}$$

where $B_a = \left| b_{k_j} \right|$ and is of order (N_m, N_b) i.e., (number

of meshes, number of branches) and $v_b(t)$ a column matrix.

There are $N_b - N_v+1$ independent mesh equations for a connected network (28, p. 73) so the reduced form of Equation (19) becomes

$$B\ v_b(t) = 0 \tag{20}$$

and matrix B is called the fundamental-loop matrix (8).

For the example in Figure 10, Equation (20) becomes

$$\begin{array}{l} \text{nodes} \\ \downarrow \end{array} \overset{\text{branches} \longrightarrow}{\begin{vmatrix} 1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{vmatrix}} \begin{vmatrix} v_{b1}(t) \\ v_{b2}(t) \\ v_{b3}(t) \\ v_{b4}(t) \\ v_{b5}(t) \end{vmatrix} = \begin{vmatrix} 0 \\ 0 \end{vmatrix}$$

v-i Relationships

Kirchhoff's laws are concerned only with the branch voltages and currents and with the topology of the network elements. They do not specify anything about the nature of the branch elements themselves (resistor, capacitor, inductor, sources) or the linearity of the branch element. Kirchhoff's laws are associated purely with the topology of the network.

Physically, each branch of an electrical network may consist of three distinct electrical devices: 1) a passive element; 2) an ideal voltage source in series with the passive element; and 3) an ideal current source in

parallel with the passive element.

The relative positions and orientations of these electrical devices for a general r-th branch are shown in Figure 11.

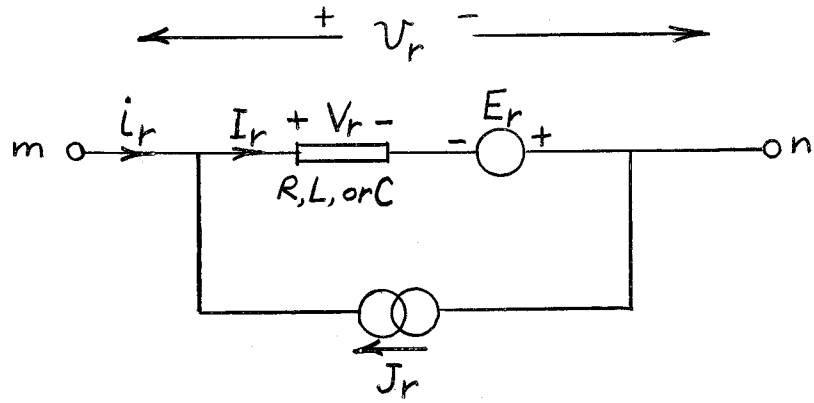


Figure 11. A general network branch.

The explicit inclusion of the time variable t has been dropped in Figure 11 and shall often be omitted in the following development.

From this diagram the following assignments can be made for the voltage and current vectors (i.e., column matrices):

E = vector of source voltages

J = vector of source currents

V = vector of element voltages

I = vector of element currents

v = vector of branch voltages

i = vector of branch currents.

The branch voltage v_r is the voltage difference between the initial node m and the final node n. Branch current i_r is the positive current which enters the branch at node m. The voltage source is orientated so that

$$E_r = V_r - v_r \tag{21}$$

and the current source so that

$$J_r = I_r - i_r \ . \tag{22}$$

Then, using the above assignments, Kirchhoff's current and voltage laws, Equations (17) and (20), can be written as

$$AI - AJ = 0 \tag{23}$$

and

$$BV - BE = 0 \tag{24}$$

In a network of b branches the self-impedance of the r-th branch may be defined as follows (6):

$$V_r = Z_{rr}I_r \text{ with } I_q = 0 \text{ for } q \neq r \text{ and } q = 1,2,\cdots b \tag{25}$$

and the self-admittance as

$$I_r = Y_{rr}V_r \text{ with } V_q = 0 \text{ for } q \neq r \text{ and } q = 1,2,\cdots b. \tag{26}$$

Likewise, the trans-impedance Z_{rs} and trans-admittance Y_{rs} between the r-th and s-th elements may be defined as:

$$V_r = Z_{rs}I_s \text{ with } I_q = 0 \text{ for } q \neq s \text{ and } q = 1,2,\cdots b \tag{27}$$

and

$$I_r = Y_{rs}V_s \text{ with } V_q = 0 \text{ for } q \neq s \text{ and } q = 1,2,\cdots b. \tag{28}$$

Then the relationship between the voltage and current in a single branch element is a generalization of Ohm's law or can be stated as

$$V_r = \sum_{s=1}^{b} Z_{rs} I_s \tag{29}$$

and

$$I_r = \sum_{s=1}^{b} Y_{rs} V_s \quad . \tag{30}$$

Or in general, Equations (29) and (30) are represented as

$$E + v = Z(J+i) \tag{31}$$

and

$$J + i = Y(E+v) \tag{32}$$

where the matrices Z and Y consist of complex numbers in the most general case, and are mutually inverse, that is, $YZ = ZY = U$ where U is a unit matrix. Equations (31) and (32), written in matrix form become

$$\begin{vmatrix} V \\ I \end{vmatrix} = \begin{vmatrix} Z & 0 \\ 0 & Y \end{vmatrix} \begin{vmatrix} I \\ V \end{vmatrix} \quad . \tag{33}$$

Ohm's law is entirely independent of the way in which the network branches are interconnected. In other words, Z and Y are related only to the "hardware" from which the network is to be constructed.

A capacitor obeys a differential equation like

$$I(t) = C \frac{dV(t)}{dt} = C\dot{V}(t) \quad , \tag{34}$$

an inductor obeys a differential equation like

$$V(t) = L \frac{dI(t)}{dt} = L\dot{I}(t) \quad , \tag{35}$$

and a resistor obeys an algebraic equation like

$$V(t) = RI(t) \ . \tag{36}$$

Equations (34), (35), and (36) can be combined into one matrix equation as

$$Y_c \ V = Z_{LR} \ I \tag{37}$$

For the network in Figure 10, Equation (33) is

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \end{bmatrix} = \begin{bmatrix} L_1\frac{d}{dt} & & & & & & & & & \\ & R_2 & & & & & & & & \\ & & R_3 & & & & & & & \\ & & & \frac{1}{C_4}\int_o^t dt & & & & & & \\ & & & & R_5 & & & & & \\ & & & & & \frac{1}{L_1}\int_o^t dt & & & & \\ & & & & & & \frac{1}{R_2} & & & \\ & & & & & & & \frac{1}{R_3} & & \\ & & & & & & & & C_4\frac{d}{dt} & \\ & & & & & & & & & \frac{1}{R_5} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \end{bmatrix}$$

and Y_c and Z_{LR} are

$$Y_c = \begin{vmatrix} 1 & . & . & . & . \\ . & 1 & . & . & . \\ . & . & 1 & . & . \\ . & . & . & C_4\frac{d}{dt} & . \\ . & . & . & . & 1 \end{vmatrix} \ , \ Z_k = \begin{vmatrix} L_1\frac{d}{dt} & & & & \\ & R_2 & & & \\ & & R_3 & & \\ & & & 1 & \\ & & & & R_5 \end{vmatrix} .$$

Network System of Equations

The three fundamental systems of equations, i.e., Kirchhoff's and Ohm's laws, constitute the starting point for the development of the loop and node systems of equations. Ohm's law leads to a system of ordinary integro-differential equations with constant coefficients and are generally solved using the Laplace transform method.

The normal procedure for solving a system of simultaneous ordinary integrodifferential equations by numerical means would be first to express the system as a higher-order set of differential equations by removing the integrals through differentiation. Then the system of higher-order differential equations is reduced to a set of first-order equations by making a simple change of variable. This is illustrated by the following example. An n-th-order equation,

$$y^{(n)} = f\,(x, y, y', y'', \cdots, y^{(n-1)})$$

may be transformed by letting

$$\begin{aligned}
y &= y_o \\
y^{1} &= y_1 \\
y^{11} &= y^{1}_{1} = y_2 \\
y^{111} &= y^{11}_{1} = y^{1}_{2} = y_3 \\
&\vdots \\
y^{(n)} &= \cdots = y^{1}_{n-1} = f(x, y_o, y_1, y_2, \cdots, y_{n-1})
\end{aligned}$$

The computation proceeds in parallel; one begins at some starting point and makes one step for each equation before proceeding to the next increment. When the derivative functions are evaluated, the current values of the array of the functional values (the y's) are used.

An alternative development, and the one to be followed here, consists of choosing an appropriate set of network variables that leads automatically to a set of first-order differential equations. When voltages across capacitors and currents through inductances are used as dependent variables and time as the independent variable, a set of first-order differential equations is obtained of the form:

$$\dot{x} + Ax = y$$

in which x is the vector of independent variables and y represents the forcing functions or sources (2,8). The A matrix consists of scalar elements which are combinations of the inductances, capacitances or resistances of the network.

The minimal set of branch currents and voltages whose instantaneous values are sufficient to determine completely the instantaneous state of the network is called a 'complete set of dynamically-independent network variables' or a state variable set. The process used to arrive at a complete set of dynamically-independent

network variables depends upon forming the branch equations in a particular manner. This form involves the classification of the network branches into the following six groups with respect to any proper C-tree, that is, a network tree which contains the maximum number of capacitor branches and the minimum number of inductor branches;

class	description
α	capacitive links
β	resistive links
γ	inductive links
λ	capacitive branches
σ	resistive branches
μ	inductive branches.

This branch classification is now used to subdivide, or partition, the network vectors and matrices into sub-vectors and sub-matrices.

First the branch voltage and current are partitioned as

$$i = \begin{vmatrix} i_\alpha \\ i_\beta \\ i_\gamma \\ i_\lambda \\ i_\sigma \\ i_\mu \end{vmatrix} \quad \text{and} \quad e = \begin{vmatrix} e_\alpha \\ e_\beta \\ e_\gamma \\ e_\lambda \\ e_\sigma \\ e_\mu \end{vmatrix} \tag{38}$$

and then the element and source vectors are partitioned into link contained (subscript L) or branch contained (subscript T) vectors as follows

$$E = \begin{vmatrix} E_L \\ E_T \end{vmatrix} \qquad J = \begin{vmatrix} J_L \\ J_T \end{vmatrix}$$

$$V = \begin{vmatrix} V_L \\ V_T \end{vmatrix} \qquad I = \begin{vmatrix} I_L \\ I_T \end{vmatrix} \tag{39}$$

Each of the subvectors of Equation (39) are then further partitioned as follows

$$E_L = \begin{vmatrix} E_\alpha \\ E_\beta \\ E_\gamma \end{vmatrix} \qquad E_T = \begin{vmatrix} E_\lambda \\ E_\sigma \\ E_\mu \end{vmatrix}$$

$$J_L = \begin{vmatrix} J_\alpha \\ J_\beta \\ J_\gamma \end{vmatrix} \qquad J_T = \begin{vmatrix} J_\lambda \\ J_\sigma \\ J_\mu \end{vmatrix}$$

$$V_L = \begin{vmatrix} V_\alpha \\ V_\beta \\ V_\gamma \end{vmatrix} \qquad V_T = \begin{vmatrix} V_\lambda \\ V_\sigma \\ V_\mu \end{vmatrix} \tag{40}$$

and

$$I_L = \begin{vmatrix} I_\alpha \\ I_\beta \\ I_\gamma \end{vmatrix} \qquad I_T = \begin{vmatrix} I_\lambda \\ I_\sigma \\ I_\mu \end{vmatrix}$$

The state variables, I_γ and V_λ, as mentioned previously can be used to completely describe the state of a network. Then I_γ and V_λ as dependent variables can describe the network performance by a system of first-order differential equations.

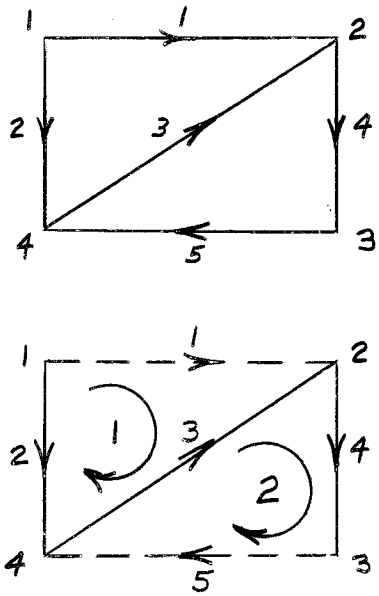
In addition to Equations (38), (39), and (40), the branch node matrix A and the fundamental-loop matrix B may be partitioned as

$$A = \left| A_L \quad A_T \right| \tag{41}$$

$$B = \left| U \quad B_T \right| . \tag{42}$$

The subscripts L and T refer to links (or chords) and tree branches, respectively and U is again the unit matrix.

Referring again to the network in Figure 10 a tree may be chosen as shown in Figure 12.



(a) Linear graph of Figure 10.

(b) Tree of (a), (solid lines).

Figure 12. Linear graph and tree of Figure 10.

For the tree of Figure 12(b) Equation (41) becomes (again eliminating the equation at node 3)

$$A = \left| A_L \quad A_T \right| = \begin{array}{c} \text{branches} \longrightarrow \\ \text{nodes} \downarrow \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 5 & 2 & 3 & 4 \end{array} \\ \left| \begin{array}{ccccc} 1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & -1 & 1 \\ 0 & -1 & -1 & 1 & 0 \end{array} \right| \end{array} \begin{array}{l} 1 \\ 2 \\ 4 \end{array}$$

and Equation (42) becomes

$$B = \left|U \quad B_T\right| = \begin{array}{c} \\ 1 \\ 2 \end{array} \left| \begin{array}{ccccc} 1 & 5 & 2 & 3 & 4 \\ 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{array} \right| .$$

Besides the reduced incidence matrix A and the fundamental circuit matrix B there is an additional matrix of interest in the following development. It is the fundamental cut-set matrix and may be constructed from either the A or B matrices. A cut-set is a set of edges of a connected graph such that the removal of the set of edges separates the graph into two pieces. The fundamental cut-set matrix is defined as the system of cut-sets in which each cut-set includes exactly one branch of a tree of the graph. The use of fundamental cut-sets provides a completely general method of choosing an appropriate set of network equations, i.e., a sufficient number of independent equations.

The fundamental cut-set matrix is formed as follows: The fundamental cut-set matrix $C = \left[c_{ij}\right]$ has one row for each branch of a tree of the graph and one column for each tree branch and link.

For example, from Figure 12(b) the fundamental cut-set matrix is

$$C = \begin{array}{c} \\ 2 \\ 3 \\ 4 \end{array} \left| \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 \end{array} \right|$$

The fundamental cut-set matrix C equations are analogous to the fundamental loops. In fact, the cut-set matrix contains the incidence matrix A where the rows of A are expressible as linear combinations of rows of C (29, p. 96). Since each fundamental cut-set contains a tree branch which is in no other cut-set the Kirchhoff current law equations within C are linearly independent. This fact makes the cut-set matrix more attractive to use than the incidence matrix because linearly independent equations are insured, whereas they are not in the reduced incidence matrix.

Using the fundamental cut-set matrix, Kirchhoff's current law expression of Equation (23) becomes

$$C\ I - C\ J = 0\ . \tag{43}$$

If the columns of the A, B, and C matrices are ordered according to branches and links as was done in Equations (41) and (42) then the fundamental cut-set matrix can be partitioned as

$$C = \left| C_L \quad U \right| \tag{44}$$

and the following relations result (29, p. 98)

$$C = A_T^{-1} A \quad \text{and} \quad C_L = -B_T' = A_T^{-1} A_L\ . \tag{45}$$

Using Equation (45), Equation (44) becomes

$$C = \left| -B_T' \quad U \right|\ . \tag{46}$$

Next the B and C matrices are partitioned according

to the six classes described on page 49 (8,9)

$$B = \begin{vmatrix} 1 & 0 & 0 & B_{\alpha\lambda} & B_{\alpha\sigma} & B_{\alpha\mu} \\ 0 & 1 & 0 & B_{\beta\lambda} & B_{\beta\sigma} & B_{\beta\mu} \\ 0 & 0 & 1 & B_{\gamma\lambda} & B_{\gamma\sigma} & B_{\gamma\mu} \end{vmatrix}$$

and (47)

$$C = \begin{vmatrix} -B_{\alpha\lambda}' & -B_{\beta\lambda}' & -B_{\gamma\lambda}' & 1 & 0 & 0 \\ -B_{\alpha\sigma}' & -B_{\beta\sigma}' & -B_{\gamma\sigma}' & 0 & 1 & 0 \\ -B_{\alpha\mu}' & -B_{\beta\mu}' & -B_{\gamma\mu}' & 0 & 0 & 1 \end{vmatrix}$$

where, for example, the columns correspond to branches and the rows to links for the circuit matrix.

In the construction of the network tree it follows that the loops defined by the capacitive links must be purely capacitive, hence

$$B_{\alpha\sigma} = 0$$

and

$$B_{\alpha\mu} = 0 \quad .$$

Further, the loops defined by the resistive links can contain no inductors, so

$$B_{\beta\mu} = 0 \quad .$$

Then it follows from the manner followed in constructing the network tree that Equations (47) can be written

$$B = \begin{vmatrix} 1 & 0 & 0 & B_{\alpha\lambda} & 0 & 0 \\ 0 & 1 & 0 & B_{\beta\lambda} & B_{\beta\sigma} & 0 \\ 0 & 0 & 1 & B_{\gamma\lambda} & B_{\gamma\sigma} & B_{\gamma\mu} \end{vmatrix}$$

and (48)

$$C = \begin{vmatrix} -B'_{\alpha\lambda} & -B'_{\beta\lambda} & -B'_{\gamma\lambda} & 1 & 0 & 0 \\ 0 & -B'_{\beta\sigma} & -B'_{\gamma\sigma} & 0 & 1 & 0 \\ 0 & 0 & -B'_{\gamma\mu} & 0 & 0 & 1 \end{vmatrix} . \qquad (48) \text{ (cont'd)}$$

In addition the coefficient matrices for the branch capacitance and resistance will be denoted by

$$C = \begin{vmatrix} C_{\alpha\alpha} & 0 \\ 0 & C_{\lambda\lambda} \end{vmatrix}$$

and (49)

$$R = \begin{vmatrix} R_{\beta\beta} & 0 \\ 0 & R_{\sigma\sigma} \end{vmatrix}$$

and the matrix of the self and mutual inductances by

$$L = \begin{vmatrix} L_{\gamma\gamma} & L_{\gamma\mu} \\ L_{\mu\gamma} & L_{\mu\mu} \end{vmatrix} . \qquad (50)$$

Finally we can construct a combined matrix equation describing the network equations. To do this, Ohm's law, Equations (34), (35), (36), (49), and (50), Kirchhoff's voltage law, Equation (24), and Kirchhoff's current law in terms of the fundamental cut-set matrix, Equation (43) are used.

The single matrix representing the network equations become

$$\begin{vmatrix} 0 & B \\ C & 0 \\ Z_{RL} & -Y_C \end{vmatrix} \begin{vmatrix} I \\ V \end{vmatrix} = \begin{vmatrix} BE \\ CJ \\ 0 \end{vmatrix} . \qquad (51)$$

This equation in expanded form is (52)

$$
\begin{array}{l|cccccccccccc|c|c}
 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & 0 & 0 & B_{\alpha\lambda} & 0 & 0 & I_\alpha & BE \\
\text{Kirchhoff voltage laws} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 1 & 0 & B_{\beta\lambda} & B_{\beta\sigma} & 0 & I_\beta & BE \\
 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 1 & B_{\gamma\lambda} & B_{\gamma\sigma} & B_{\gamma\mu} & I_\gamma & BE \\
\hline
 & -B'_{\alpha\lambda} & -B'_{\beta\lambda} & -B'_{\gamma\lambda} & 1 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & I_\lambda & CJ \\
\text{Kirchhoff current laws} & 0 & -B'_{\beta\sigma} & -B'_{\gamma\sigma} & 0 & 1 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & I_\sigma & CJ \\
 & 0 & 0 & -B'_{\gamma\mu} & 0 & 0 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & I_\mu & CJ \\
\hline
 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & -C_\alpha \frac{d}{dt} & \cdot & \cdot & \cdot & \cdot & \cdot & V_\alpha & 0 \\
 & \cdot & R_{\beta\beta} & \cdot & \cdot & \cdot & \cdot & \cdot & -1 & \cdot & \cdot & \cdot & \cdot & V_\beta & 0 \\
\text{Ohm's law equations} & \cdot & \cdot & L_{\gamma\gamma}\frac{d}{dt} & \cdot & \cdot & L_{\gamma\mu}\frac{d}{dt} & \cdot & \cdot & -1 & \cdot & \cdot & \cdot & V_\gamma & 0 \\
 & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & C_\lambda \frac{d}{dt} & \cdot & \cdot & V_\lambda & 0 \\
 & \cdot & \cdot & \cdot & \cdot & R_{\sigma\sigma} & \cdot & \cdot & \cdot & \cdot & \cdot & -1 & \cdot & V_\sigma & 0 \\
 & \cdot & \cdot & L_{\mu\gamma}\frac{d}{dt} & \cdot & \cdot & L_{\mu\mu}\frac{d}{dt} & \cdot & \cdot & \cdot & \cdot & \cdot & -1 & V_\mu & 0
\end{array}
$$

The vector on the right side of Equations (51) and (52) are column vectors representing the loop voltage generators and the nodal current generators.

Equations (51) and (52) summarize the three basic laws of electrical networks and Branin (6) has made a formal statement of the electrical network problem as follows:

Given: 1) an electrical network whose linear graph determines the matrices A, B, and C,
2) the impedance matrix Z or its inverse Y, and
3) the arbitrary voltage and current sources E and J;

Find: 4) the branch voltages and currents, v and i, so that the three basic laws of electrical networks hold true.

There are several methods for solving this problem, of which the two classical methods are the node and mesh methods. In node analysis method the node-to-reference voltages are used as auxiliary variables and in the mesh method the mesh currents are used. In any case the interest is in obtaining a complete set of dynamically independent network variables whose instantaneous values are sufficient to determine completely the state of the network at an instant in time.

Elimination Procedure

Starting with the network equations of Equation (51) or (52) the procedure to be followed is to eliminate variables until a set of dynamically-independent variables is obtained. The process of elimination depends on forming the branch equations in a certain form. This form involves classifying the network branches into the six classes stated earlier. This produces twelve classes for the branch voltages and currents as in Equation (38). The elimination process consists of expressing ten of these classes algebraically in terms of the remaining two classes. It has been shown (2) that by using the voltages across capacitors and the currents through inductors as the set of independent variables, a set of first-order differential equations describe the response

of a network.

The branch equations used in the elimination process are formed by using a particular tree construction (9, 31). Tree-branch elements are chosen in the order; capacitors, resistors, and inductors. Then to carry out the elimination process the variables

$$I_\alpha \ , \ I_\beta \ , \ I_\lambda \ , \ I_\sigma \ , \ I_\mu \ , \ V_\alpha \ , \ V_\beta \ , \ V_\gamma \ , \ V_\sigma \ , \ V_\mu$$

are expressed in terms of the state variables

$$I_\gamma \quad \text{and} \quad V_\lambda$$

together with the various generators and their differentials.

The elimination procedure to be carried out below will consist of expressing Kirchhoff's laws in terms of the six voltage and current classes. Then Ohm's law will be applied to resistors to solve for the resistor voltages and currents and finally the voltages and currents for the inductors and capacitors will be solved for in terms of I_γ and V_λ .

Kirchhoff's voltage and current laws from Equations (24) and (43), or (51), are

$$BV - BE = 0 \qquad (24)$$

$$CI - CJ = 0 \qquad (43)$$

and using Equations (42) and (46) then

$$B = \left| U \quad B_T \right| \qquad (42)$$

$$C = \left| -B_T' \quad U \right| \qquad (46)$$

partitioning according to Equation (38) Kirchhoff's voltage law may be written as

$$B\,V = B\,E$$

$$\left|U\ B_T\right| V = \left|U\ B_T\right| E$$

$$\left|U\ B_T\right| \begin{vmatrix} V_\alpha \\ V_\beta \\ V_\gamma \\ V_\lambda \\ V_\sigma \\ V_\mu \end{vmatrix} = \left|U \cdot B_T\right| \begin{vmatrix} E_\alpha \\ \cdot \\ \cdot \\ \cdot \\ \\ E_\mu \end{vmatrix}$$

$$\begin{vmatrix} V_\alpha \\ V_\beta \\ V_\gamma \end{vmatrix} + B_T \begin{vmatrix} V_\lambda \\ V_\sigma \\ V_\mu \end{vmatrix} = \begin{vmatrix} E_\alpha \\ E_\beta \\ E_\gamma \end{vmatrix} + B_T \begin{vmatrix} E_\lambda \\ E_\sigma \\ E_\mu \end{vmatrix} .$$

Next using Equation (48)

$$\begin{vmatrix} V_\alpha \\ V_\beta \\ V_\gamma \end{vmatrix} + \begin{vmatrix} B_{\alpha\lambda} & 0 & 0 \\ B_{\beta\lambda} & B_{\beta\sigma} & 0 \\ B_{\gamma\lambda} & B_{\gamma\sigma} & B_{\gamma\mu} \end{vmatrix} \begin{vmatrix} V_\lambda \\ V_\sigma \\ V_\mu \end{vmatrix} = \begin{vmatrix} E_\alpha \\ E_\beta \\ E_\gamma \end{vmatrix} + \begin{vmatrix} B_{\alpha\lambda} & 0 & 0 \\ B_{\beta\lambda} & B_{\beta\sigma} & 0 \\ B_{\gamma\lambda} & B_{\gamma\sigma} & B_{\gamma u} \end{vmatrix} \begin{vmatrix} E_\lambda \\ E_\sigma \\ E_\mu \end{vmatrix}$$

which, when expanded, gives

$$V_\alpha + B_{\alpha\lambda} V_\lambda = E_\alpha + B_{\alpha\lambda} E_\lambda \qquad (53a,b,c)$$

$$V_\beta + B_{\beta\lambda} V_\lambda + B_{\beta\sigma} V_\sigma = E_\beta + B_{\beta\lambda} E_\lambda + B_{\beta\sigma} E_\sigma$$

$$V_\gamma + B_{\gamma\lambda} V_\lambda + B_{\gamma\sigma} V_\sigma + B_{\gamma\mu} V_\mu = E_\gamma + B_{\gamma\lambda} E_\lambda + B_{\gamma\sigma} E_\sigma + B_{\gamma\mu} E_\mu .$$

By a similar process Kirchhoff's current law may be developed as

$$C\ I = C\ J$$

$$\left|-B_T' U\right| I = \left|-B_T' U\right| J$$

$$-B_T' \begin{vmatrix} I_\alpha \\ I_\beta \\ I_\gamma \end{vmatrix} + \begin{vmatrix} I_\lambda \\ I_\sigma \\ I_\mu \end{vmatrix} = -B_T' \begin{vmatrix} J_\alpha \\ J_\beta \\ J_\gamma \end{vmatrix} + \begin{vmatrix} J_\lambda \\ J_\sigma \\ J_\mu \end{vmatrix} .$$

Using Equation (48) gives

$$-B'_{\alpha\lambda} I_\alpha - B'_{\beta\lambda} I_\beta - B'_{\alpha\mu} I_\gamma + I_\lambda = -B'_{\alpha\lambda} J_\alpha - B'_{\beta\lambda} J_\beta - B'_{\gamma\mu} J_\gamma + J_\lambda$$

$$- B'_{\beta\sigma} I_\beta - B'_{\gamma\sigma} I_\gamma + I_\sigma = -B'_{\beta\sigma} J_\beta - B'_{\gamma\sigma} J_\gamma + J_\sigma$$

$$- B'_{\gamma\mu} I_\gamma + I_\mu = -B'_{\gamma\mu} J_\gamma + J_\mu .$$

With rearranging

$$I_\lambda = B'_{\alpha\lambda} I_\alpha + B'_{\beta\lambda} I_\beta + B'_{\gamma\mu} I_\gamma - \left[B'_{\alpha\lambda} J_\alpha + B'_{\beta\lambda} J_\beta + B'_{\gamma\mu} J_\gamma\right] + J_\lambda$$

$$I_\sigma = B'_{\beta\sigma} I_\beta + B'_{\gamma\sigma} I_\gamma - \left[B'_{\beta\sigma} J_\beta + B'_{\gamma\sigma} J_\gamma\right] + J_\sigma$$

$$I_\mu = B'_{\gamma\mu} I_\gamma - B'_{\gamma\mu} J_\gamma + J_\mu . \qquad (54a,b,c)$$

Next Ohm's law for resistors will be used to solve for I_β . From Equation (52)

$$R_{\beta\beta} I_\beta = V_\beta \qquad (55)$$

and

$$R_{\sigma\sigma} I_\sigma = V_\sigma . \qquad (56)$$

Substituting V_β from Equation (53b) into Equation (55) and solving for I_β yields

$$I_\beta = R_{\beta\beta}^{-1}\left[E_\beta + B_{\beta\lambda} E_\lambda + B_{\beta\sigma} E_\sigma - B_{\beta\lambda} V_\lambda - B_{\beta\sigma} V_\sigma\right] . \qquad (57)$$

Equations (56) and (54b) yield

$$V_\sigma = R_{\sigma\sigma}\ B'_{\beta\sigma} I_\beta + B'_{\gamma\sigma} I_\gamma - (B'_{\beta\sigma} J_\beta + B'_{\gamma\sigma} J_\gamma) + J_\sigma$$

and substituting this value for V_σ into Equation (57) produces

$$I_\beta = R_{\beta\beta}^{-1}\Big[E_\beta + E_{\beta\lambda}E_\lambda + B_{\beta\sigma}E_\sigma - B_{\beta\lambda}V_\lambda - B_{\beta\sigma}\Big[R_{\sigma\sigma}\Big[B'_{\beta\sigma}I_\beta + B'_{\gamma\sigma}I_\gamma - (B'_{\beta\sigma}J_\beta + B'_{\gamma\sigma}J_\gamma) + J_\sigma\Big]\Big]\Big]$$

solving for I_β

$$I_\beta = \Big[R_{\beta\beta} + B_{\beta\sigma}R_{\sigma\sigma}B'_{\beta\sigma}\Big]^{-1}\Big[E_\beta + B_{\beta\lambda}E_\lambda + B_{\beta\sigma}E_\sigma - B_{\beta\lambda}V_\lambda - \Big[B_{\beta\sigma}R_{\sigma\sigma}\Big]\Big[B'_{\gamma\sigma}I_\gamma + J_\sigma - B'_{\beta\sigma}J_\beta + B'_{\gamma\sigma}J_\gamma\Big]\Big]. \quad (59)$$

Then with a knowledge of the state variables I_γ and V_λ all of the terms on the right side of Equation (59) are known and thus I_β may be computed. The value of I_σ may be computed once I_β is known by using Equation (54b) and by then using Equations (55) and (56) V_β and V_σ can be calculated.

Table I summarizes the solution at this point in the derivation. Starting with the state variables the

TABLE I. SOLUTION VARIABLES

I_γ	state variable
V_λ	state variable
I_β	Equation (59)
I_σ	I_β and Equation (54b)
V_β	I_β and Equation (55)
V_σ	I_σ and Equation (56)
I_α, I_λ, I_μ, V_α, V_γ, V_λ, V_μ remain to be derived.	

current in the tree branch resistors and then the tree chord resistor currents are calculated. Knowing the

resistor currents, the resistor voltages may be found. The next step is to solve for the inductor voltages.

I_γ is a state variable, therefore, by using Equation (54c) I_μ can be found directly. The inductor voltages V_γ and V_μ can be found by using Kirchhoff's laws from Equations (53c) and (54c) and Ohm's law relationships from Equation (52), repeated below

$$L_{\gamma\gamma}\frac{d}{dt} I_\gamma + L_{\gamma\mu}\frac{d}{dt} I_\mu = V_\gamma \tag{60}$$

$$L_{\mu\gamma}\frac{d}{dt} I_\gamma + L_{\mu\mu}\frac{d}{dt} I_\mu = V_\mu \tag{61}$$

By using the four equations and eliminating I_μ, V_γ, and V_μ, $\dot{I}_\gamma$ may be solved for. A dot is used to denote differentiation with respect to time. The procedure is to differentiate Equation (54c) with respect to time producing

$$\dot{I}_\mu = B'_{\gamma\mu}\dot{I}_\gamma - B'_{\gamma\mu}\dot{J}_\gamma + \dot{J}_\mu \ . \tag{62}$$

Equation (62) is then substituted into Equations (60) and (61) to eliminate $\dot{I}_\mu$. The resulting expressions for V_γ and V_μ are then used in Equation (53c) which is finally solved for $\dot{I}_\gamma$, yielding

$$\begin{aligned}\dot{I}_\gamma = &\left[L_{\gamma\gamma} + L_{\gamma\mu}B'_{\gamma\mu} + B_{\gamma\mu}L_{\mu\gamma} + B_{\gamma\mu}L_{\mu\mu}B'_{\gamma\mu}\right]^{-1} \\ &\Big[E_\gamma + B_{\gamma\lambda}E_\lambda + B_{\gamma\sigma}E_\sigma + B_{\gamma\mu}E_\mu - \left[B_{\gamma\lambda}V_\lambda + B_{\gamma\sigma}V_\sigma\right] \\ &+ \left[B_{\gamma\mu}L_{\mu\mu} + L_{\gamma\mu}\right]\left[B_{\gamma\mu}\dot{J}_\gamma - \dot{J}_\mu\right]\Big] \ .\end{aligned} \tag{63}$$

$\dot{I}_\mu$ can be solved for by substituting $\dot{I}_\gamma$ into the derivative of Equation (54c). Finally the inductor

voltages V_γ and V_μ are evaluated from Equations (60) and (61). The equations are as follows:

$$\dot{I}_\mu = B'_{\gamma\mu}\,\dot{I}_\gamma - B'_{\gamma\mu}\,\dot{J}_\gamma + \dot{J}_\mu \tag{64}$$

$$V_\gamma = L_{\gamma\gamma}\,\dot{I}_\gamma + L_{\gamma\mu}\,\dot{I}_\mu \tag{65}$$

$$V_\mu = L_{\mu\gamma}\,\dot{I}_\gamma + L_{\mu\mu}\,\dot{I}_\mu \quad . \tag{66}$$

Since the capacitor voltage V_λ is a state variable, V_α can be computed directly from Equation (53a). The capacitor currents I_α and I_λ can be found from a process similar to that for the inductor voltages.

Beginning with Ohm's law relationships from Equation (52)

$$C_\alpha\,\dot{V}_\alpha = I_\alpha \tag{67}$$

$$C_\lambda\,\dot{V}_\lambda = I_\lambda \tag{68}$$

and using Equations (53a) and (54a) the solution for I_α follows

$$I_\alpha = \left[C_\alpha^{-1} + B_{\alpha\lambda}C_\lambda^{-1}\,B'_{\alpha\lambda}\right]^{-1}\left[\dot{E}_\alpha + B_{\alpha\lambda}\dot{E}_\lambda - B_{\alpha\lambda}C_\lambda^{-1}\left[B'_{\beta\lambda}I_\beta + B'_{\gamma\mu}I_\gamma - (B'_{\alpha\lambda}J_\alpha + B'_{\beta\lambda}J_\beta + B'_{\gamma\mu}J_\gamma) + J_\lambda\right]\right] . \tag{69}$$

Current I_λ follows directly from Equation (54a) and $\dot{V}_\lambda$ from Equation (68).

All of the variables in Table I can now be expressed in terms of I_γ and V_λ. To summarize, the terms present in each variable are shown in Figure 13. Beginning with the state variables, the "flow" of the solution is

depicted with the dependence of each variable on the others indicated. Equations used in each step are indicated on the path arcs.

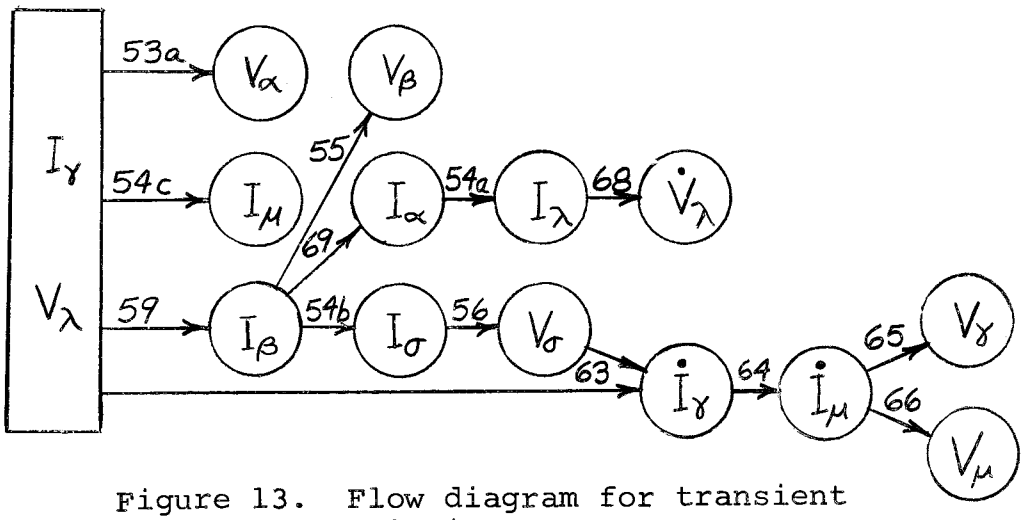


Figure 13. Flow diagram for transient solution.

Solution Procedure

The set of first-order differential equations established from topological circuit considerations can be evaluated at each value of circuit time provided the values of the state variables are known at each time. The normal procedure for obtaining a time history of the circuit response is to start with time equal to zero and evaluate the network initial conditions to establish values for the state variables.

Solution of the network equations is then accomplished through matrix operations and produces state variable derivatives. These derivatives must be

numerically integrated to produce new state variables for the evaluation at the next time step. That is, the state variable S at time i+1 is evaluated from

$$S_{(i+1)} = S_{(i)} + \Delta t \dot{S}_{(i)} \tag{70}$$

where Δt is the time step between evaluations of the network equations. After the computed derivatives of the state variables are numerically integrated, the

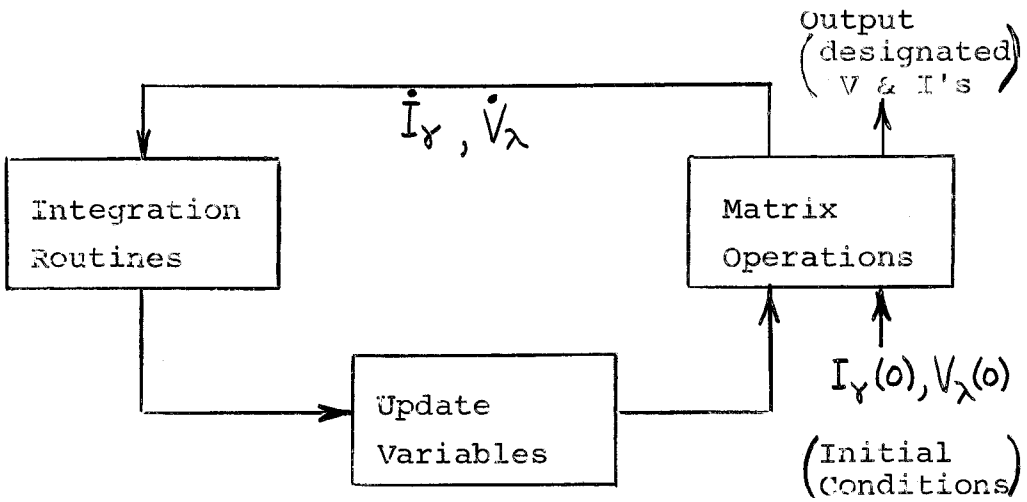


Figure 14. Transient solution procedure.

variable elements and sources are updated and the system of network equations are evaluated at the next time step. The value of each element voltage and current is available at each time step as output. This solution procedure is shown schematically in Figure 14. A repetitive evaluation of the transient solution and subsequent integration of the state variable derivatives produces the time history of the network response.

Numerical Integration

The solution of the network equations, among other things, depends on integrating the differential equations describing the state variables at each time step. The rate at which the solution progresses is dependent on the method of performing this integration. There are two basic categories of methods available for this process:

1. One-step methods which do not depend on the past history of the solution. The most widely used one-step method is the Runge-Kutta.
2. Multi-step methods in which the next solution point is obtained by using previous solutions. Most methods of this type are called predictor-corrector.

Basically the process can be described as follows: We take the starting point, compute the slope, and move in that direction for a short distance, and select the next point. Using this as a new starting point the slope is again evaluated and so it proceeds. No control of the errors between the true solution and the computed solution is achieved by this method, so generally a slightly different procedure is followed.

In general, the solution to the first-order differential equation

$$\frac{dx}{dt} = f(x,t) \tag{71}$$

effectively replaces the result of truncating a Taylor-series expansion of the form

$$x_{n+1} = x_n + h\dot{x}_n + \frac{h^2}{2!}\ddot{x}_n + \frac{h^3}{3!}\dddot{x}_n + \cdots \tag{72}$$

by an approximation in which x_{n+1} is calculated from a formula of the type (17, p. 233)

$$x_{n+1} = x_{n-p} + h(a_{-1}\dot{x}_{n-1} + a_o\dot{x}_n + a_1\dot{x}_{n-1} + \cdots + a_r\dot{x}_{n-r}). \tag{73}$$

In the special case where $f(x,t) = Ax$, where A is a constant the preceding equation takes the form

$$(1-a_{-1}Ah)x_{n+1} = x_{n-p} + Ah(a_o x_n + a_1 x_{n-1} + \cdots + a_r x_{n-r}) \tag{74}$$

where at least one of the coefficients of a_r is nonzero. The general solution to this equation is (17, p. 203)

$$x_n = c_o p_o^n + c_1 p_1^n \cdots + c_r p_r^n \tag{75}$$

where the c coefficients are determined by the initial values of x_o, x_1, $\cdots$ x_{r-1} and p_1, p_2 $\cdots$ p_r are their roots of the characteristic equation obtained by substituting the particular solution $x_n = p^n x_o$ into Equation (74) (5).

The principal solution of Equation (75) will be generated by one of the roots and the other r-1 roots represent parasitic solutions. These parasitic

solutions correspond to the fact that the order of the difference equation exceeds the order of the approximated differential equation by r. If any one of the parasitic roots is greater in magnitude than unity, the corresponding term in Equation (75) will increase without limits as n increases (17, p. 205). This situation arises when the integration interval h is too large and is called numerical instability.

A frequently used pair of predictor-corrector formulas for numerical integration uses Milne's method and is

predictor:

$$x_{n+1} = x_{n-3} + \frac{4h}{3}(2\dot{x}_n - \dot{x}_{n-1} + 2\dot{x}_{n-2}) + \frac{14}{45}h^5 x^{(5)}(\mu) \qquad (76)$$

corrector:

$$x_{n+1} = x_{n-1} + \frac{h}{3}(\dot{x}_{n+1} + 4\dot{x}_n + \dot{x}_{n-1}) - \frac{h^5}{90} x^{(5)}(\mu) \qquad (77)$$

where the values of μ lie between the largest and smallest argument involved in the formulas and, in general, are not equal.

To avoid numerical instability in the use of the above integration formulas, and if Equation (71) is rewritten as $\dot{x} = -ax+b$, then the integration step h must satisfy the relationship $ha \leq 0.83$ (10). In network

equations a will be a matrix and consequently the above restriction on the integration step size requires that h be less than the smallest natural time constant of the network (5). Although these roots contribute least to the solution they force the numerical integration to proceed at a rate dependent on them and consistent with the allowable error in the solution.

Then in designing a system for solving the set of differential equations it is desirable to use as large a value for h as possible. Besides the formulas for integrating each step, it is necessary to have formulas for halving or doubling the integration interval and criteria for when to do so. The criteria for doubling or halving is provided through a comparison of predicted and corrected values. These formulas are given in Hamming (16, p. 208).

In contrast to the predictor-corrector methods the Runga-Kutta methods do not use information from previously calculated points and consequently are self-starting and, in fact, are generally utilized in starting the methods that do require previous solution points.

In both predictor-corrector and Runga-Kutta integration methods automatic step-size control is possible. Generally predictor-corrector methods require two derivative evaluations per time step as against four

for the Runga-Kutta method, but Warten (30) has shown that from stability considerations the extra derivative evaluations per single time step for the Runga-Kutta method do not appear as serious a drawback as might first appear.

Warten's method estimates the local truncation error in the Taylor-series expansion of the solution and then, using the error, prescribes smaller steps during the transient region and larger steps in the steady state portion. This allows the solution to progress at a rate that depends on the numerical integration errors and greatly speeds the solution process.

Chapter Summary

Figures 13 and 14 provide a summary of the transient solution procedure. The equations used in the solution process are summarized below:

$$V_\alpha = B_{\alpha\lambda}\, E_\lambda - B_{\alpha\lambda} V_\lambda + E_\alpha \tag{53a}$$

$$I_\mu = B'_{\gamma\mu}\, I_\gamma - B'_{\gamma\mu}\, J_\gamma + J_\mu \tag{54c}$$

$$I_\beta = \left[R_{\beta\beta} + B_{\beta\sigma} R_{\sigma\sigma} B'_{\beta\sigma}\right]^{-1} \left[E_\beta + B_{\beta\lambda} E_\lambda + B_{\beta\sigma} E_\sigma - B_{\beta\lambda} V_\lambda - \left[B_{\beta\sigma} R_{\sigma\sigma}\right]\left[B'_{\gamma\sigma} I_\gamma + J_\sigma - B'_{\beta\sigma} J_\beta + B'_{\gamma\sigma} J_\gamma\right]\right] \tag{59}$$

$$V_\beta = R_{\beta\beta} I_\beta \tag{55}$$

$$I_\alpha = \left[C_\alpha^{-1} + B_{\alpha\lambda} C_\lambda^{-1} B'_{\alpha\lambda}\right]^{-1} \left[\dot{E}_\alpha + B_{\alpha\lambda} \dot{E}_\lambda - B_{\alpha\lambda} C_\lambda^{-1} \left[B'_{\beta\lambda} I_\beta + B'_{\gamma\mu} I_\gamma - \left[B'_{\alpha\lambda} J_\alpha + B'_{\beta\lambda} J_\beta + B'_{\gamma\mu} J_\gamma\right] + J_\lambda\right]\right] \tag{69}$$

$$I_\lambda = B'_{\alpha\lambda} I_\alpha + B'_{\beta\lambda} I_\beta + B'_{\gamma\mu} I_\gamma - \left[B'_{\alpha\lambda} J_\alpha + B'_{\beta\lambda} J_\beta + B'_{\gamma\mu} J_\gamma\right] + J_\lambda \tag{54a}$$

$$\dot{V}_\lambda = C_\lambda^{-1} I_\lambda \tag{68}$$

$$I_\sigma = B'_{\beta\sigma} I_\beta + B'_{\gamma\sigma} I_\gamma - \left[B'_{\beta\sigma} J_\beta + B'_{\gamma\sigma} J_\gamma \right] + J_\sigma \tag{54b}$$

$$V_\sigma = R_{\sigma\sigma} I_\sigma \tag{56}$$

$$\dot{I}_\gamma = \left[L_{\gamma\gamma} + L_{\gamma\mu} B'_{\gamma\mu} + B_{\gamma\mu} L_{\mu\gamma} + B_{\gamma\mu} L_{\mu\mu} B'_{\gamma\mu} \right]^{-1} \left[E_\gamma + B_{\gamma\lambda} E_\lambda + B_{\gamma\sigma} E_\sigma + B_{\gamma\mu} E_\mu - \left[B_{\gamma\lambda} V_\lambda + B_{\gamma\sigma} V_\sigma \right] + \left[B_{\gamma\mu} L_{\mu\mu} + L_{\gamma\mu} \right] \left[B_{\gamma\mu} \dot{J}_\gamma - \dot{J}_\mu \right] \right] \tag{63}$$

$$\dot{I}_\mu = B'_{\gamma\mu} \dot{I}_\gamma - B'_{\gamma\mu} \dot{J}_\gamma + \dot{J}_\mu \tag{64}$$

$$\dot{V}_\gamma = L_{\gamma\gamma} \dot{I}_\gamma + L_{\gamma\mu} \dot{I}_\mu \tag{65}$$

$$V_\mu = L_{\mu\gamma} \dot{I}_\gamma + L_{\mu\mu} \dot{I}_\mu \tag{66}$$

The solution of these equations at each time step produces state variable derivatives ($\dot{I}_\gamma$ and $\dot{V}_\lambda$) which are numerically integrated providing the starting point for the next time increment.

The first term in brackets in Equations (59) and (69) will be diagonal matrices when only simple passive elements appear in the network. The corresponding term in Equation (63) will be non-diagonal if there are mutual inductive terms. The fact that the inverse is required for these three terms at each time step can be a significant factor in the time required in the solution process. This will be discussed in more detail in the next chapter.

When integrating the state variable derivatives in preparation for the next solution pass, it is highly desirable to use the largest integration time step

possible consistent with numerical stability. In order to do this the time step should be variable as suggested earlier. Procedures for accomplishing this are given by Ralston (25, p. 102) for predictor-corrector methods and by Warten (30) for Runga-Kutta integration methods.

The elimination process leading to the equations used in the final solution process was different from, but entirely equivalent to, Bryant's method (9). With but minor differences the equations summarized in this section are identical to those used by the PREDICT circuit analysis program in solving for the time response of a network.

IV. MODELING AND COMPUTER RUNNING TIMES

Introduction

The previous two chapters have, in turn, provided a brief background for modeling transistors for large-signal analysis and derived the algebraic matrix equations describing a general time domain analysis. In this chapter both of these developments will be utilized in establishing efficient modeling techniques for large-signal network analysis.

To establish the relationships between the model and the resulting computer running time two means will be utilized: (1) The analytical equations from the preceding chapter will be used, and (2) a general network analysis program will be used. The computer program that will be used is the PREDICT circuit analysis program developed by IBM Corporation, Space Guidance Center, Owego, New York under contract to the U.S. Air Force Weapons Laboratory (32).

The PREDICT program is a large, general purpose, network analysis program which automatically solves, from a description of the network topology, differential and algebraic matrix equations which completely characterize the behavior of a complex network. The program is the product of several years of investigation and development

of general network analysis programs by the IBM Corporation. The mathematics of the PREDICT program is formulated much in the same way as the development in the preceding section on general network analysis. PREDICT is written in both the FORTRAN II and FAP languages expressedly for the IBM 7090 - 7094 computer. A description of this program is given in the PREDICT manual (32).

Although most of the results presented in this chapter was derived using a specific program running on a particular computer the nature of the PREDICT program and the data are general. Consequently the results and conclusions will be valid for most general time-domain network analysis programs formulated within the same framework as in chapter three.

In terms of equivalent circuits and network representation, several factors are important in determining the amount of computer time that will be required in performing an analysis of a network. These factors are:

(1) The network size, i.e., the number of circuit nodes and branches;

(2) Network time constants;

(3) Network natural frequencies; and

(4) Element representation.

Each of the above will be investigated in the following subsections and will be related to network

modeling.

Network Size and Running Time

In solving the algebraic network equations several matrix operations are required. In particular, these operations are:

matrix addition,
matrix subtraction,
matrix transposition,
matrix inversion, and
matrix multiplication.

Each of the above operations are performed numerous times at each integration time step. The time required to complete each matrix operation is a function of the matrix size. Figure 15 depicts how the amount of computer time required for each operation varies as a function of matrix size or order for rather simple, unsophisticated matrix programs written in FORTRAN and run on a CDC 3600 computer. Timing for the operations were obtained from an internal clock on the computer and repeating each operation 1000 times to obtain a reasonable time difference.

Each point on the curves in Figure 15 represents one computer run. However, the statistics of the points were investigated in two ways. First a nearly diagonal tenth-order matrix was timed for 1000 repeated inversions

on ten separate computer runs. The ten samples were identical with the exception of one which was one second less (nine runs required 54 seconds and one run required 53 seconds). The clock can only be interrogated to the closest second, providing a possible plus or minus one second spread in timing for any particular operation.

Again using the matrix inversion program, ten tenth-order matrices were constructed using a random number table. In this case the spread for ten samples was from 79 to 82 milliseconds per operation. For this particular program, which uses a Gauss-Jordan elimination procedure (16, p. 366), the minimum inversion time will occur with a diagonal matrix (45 ms for a matrix of order ten). No effort was made to find the maximum time but presumedly it would be a matrix containing no zero elements.

The other matrix routines contain no statements that branch on a zero test so will exhibit no dependence on element values as does the matrix inversion program.

From Figure 15 it is immediately obvious that compared to matrix multiplication and inversion the other matrix operations require a negligible amount of computer time in the total solution process. A log-log plot of the time per operation versus the order of the matrix shows that for multiplication the time varies as the matrix order to the 2.7 power and for matrix inversion

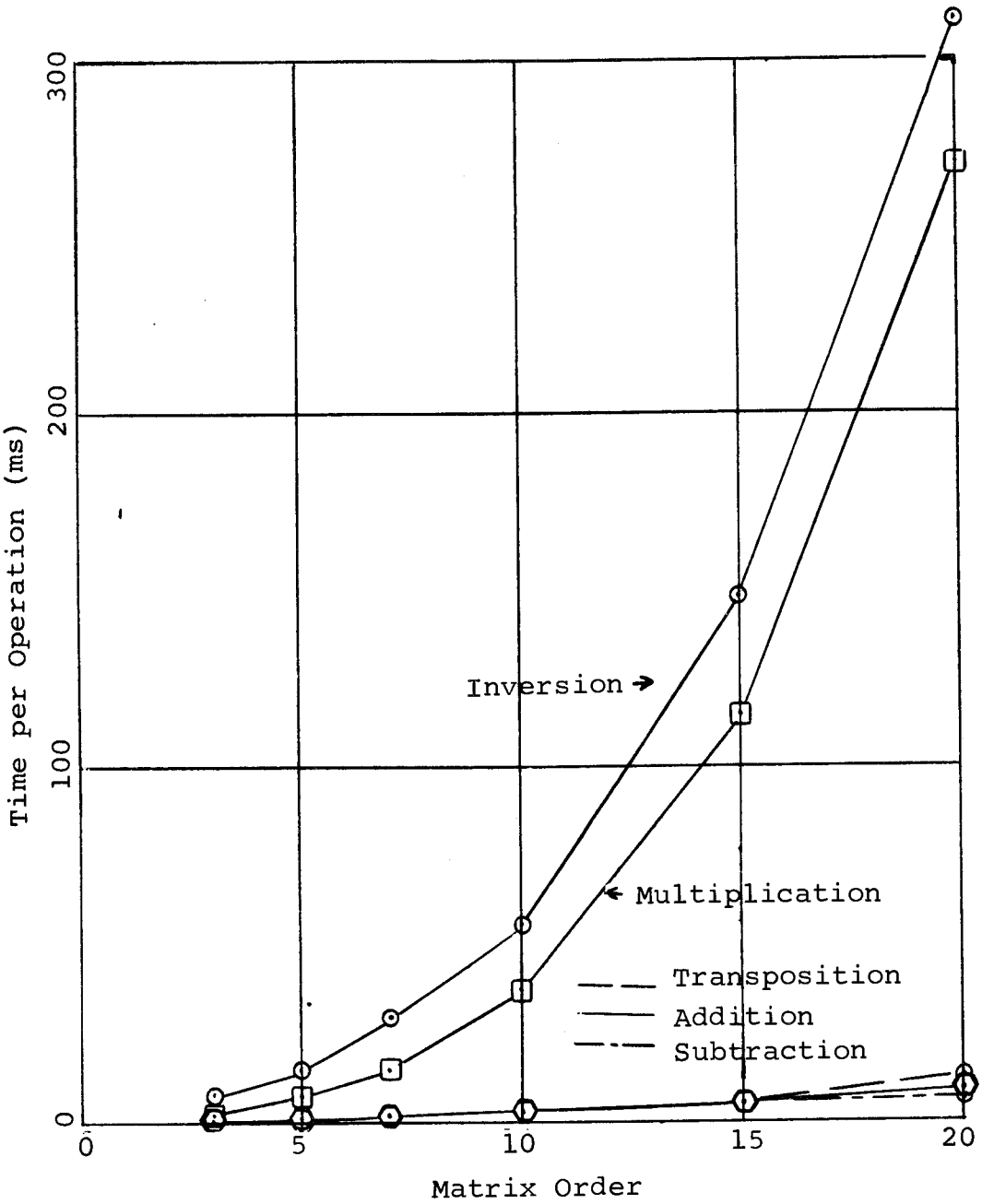


Figure 15. Time per matrix operation versus matrix size.

as the matrix order to the 2.5 power. As an approximation, taking the inverse of a matrix requires about twice the amount of time as multiplication requires up to matrix sizes of approximately ten by ten.

If the assumption is made that for a typical solution pass the number of matrix operations is approximately the same for each operation, then the matrix multiplications and inversions will constitute the major part of the solution time.

As the size of the network increases, i.e., contains a greater number of network branches and nodes, the size of the matrices will increase. Consequently the solution time will increase also and in approximately the same manner as the matrix inversion and multiplication time does. That is, the solution time should depend on the network size to approximately the same power as inversion and multiplication.

Two networks were used to verify the above statement. The first consisted of a common-emitter amplifier in which identical stages were successively added in cascade. Each stage added two nodes to the network. A plot of the program solution time versus the number of nodes for this circuit is shown in curve (a) of Figure 16. For this curve the following relationship holds:

$$\text{Program solution time} \propto (\text{Number of network nodes})^{2.1} \tag{78}$$

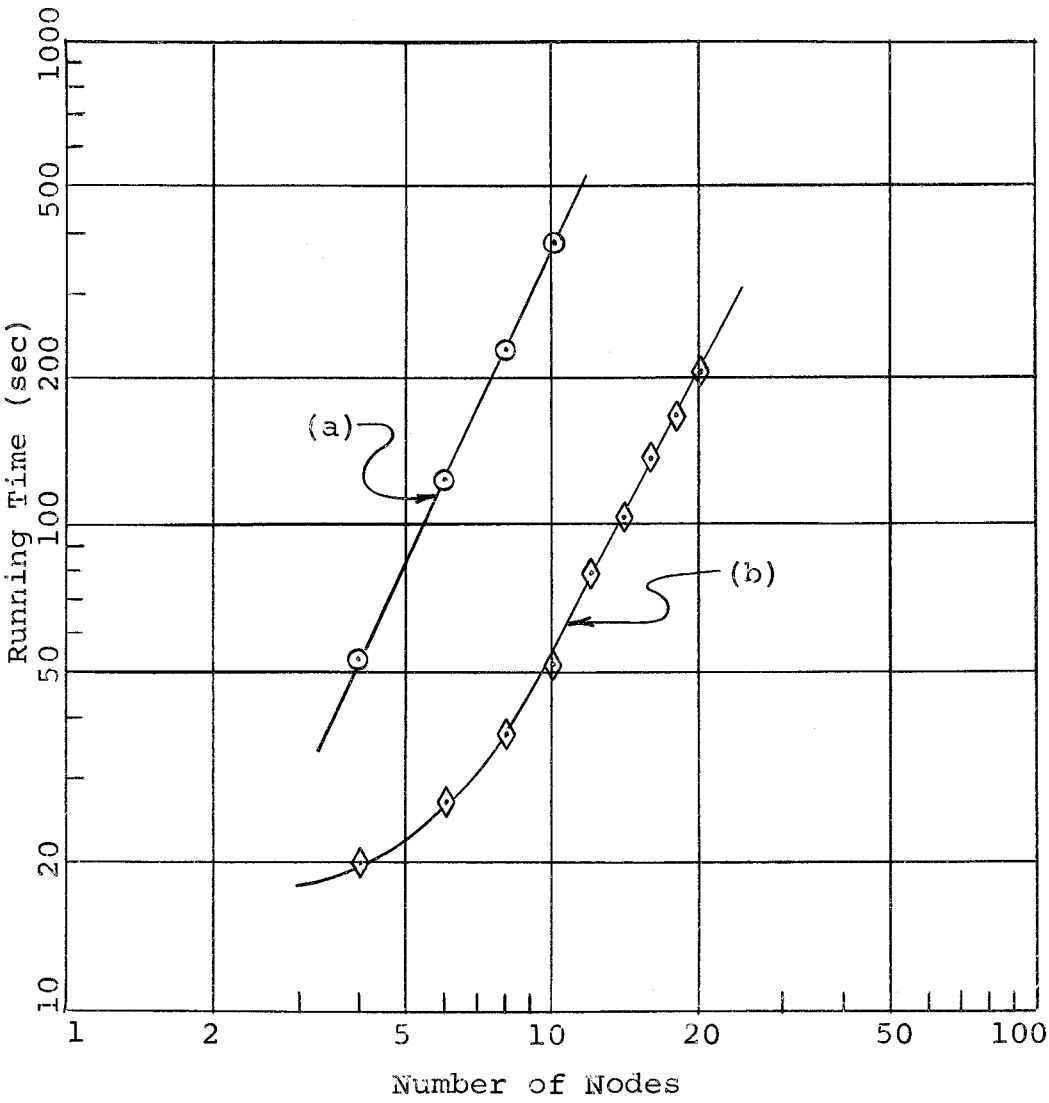


Figure 16. Computer running time versus number of network nodes.

where $\propto$ means 'proportional to'.

The second case consisted of cascading identical stages of a series-parallel RLC network. The solution time versus the number of network nodes is shown in Figure 16 curve (b). Below solution times of approximately 50 seconds, the influence of the faster matrix operations (addition, subtraction) and the programming transfer operations (calls to subroutines) cause a departure from Equation (78). Solution times above 50 seconds show a relationship in very close agreement with Equation (78).

Interpreted simply, Equation (78) means that doubling the circuit size and other factors remaining constant will have a resulting effect of increasing the running time by a factor of approximately four.

Network Time Constants and Natural Frequencies Versus Running Time

In addition to the computer solution time arising from the solving of the matrix equations themselves, the network time constants and natural frequencies play a fundamental role in determining the solution time. The distinction between time constant and natural frequency being that the time constant is equal to the reciprocal of the real part, and the natural frequency is equal to the imaginary part of a root of the network

characteristic equation. In terms of RLC elements, for a parallel RC network the time constant is the RC product and, for a parallel LC network the natural period is $2\pi\sqrt{LC}$.

As previously mentioned, the integration step size h must satisfy the equation $h \leq 0.83a$ where a is the coefficient matrix in the differential equation $\dot{x} = -ax+b$. It is easily shown by writing the network equations that the smallest value for a in the set of differential equations will correspond to the smallest natural time constant of the network, or alternatively to the largest natural frequency.

Program running time or solution time is defined as the amount of time required for the repeated solution of the network equations. It does not include the computer time used in formulating the network equations in matrix form or the reading out of the computed results by the computer.

Two examples will be used to show the nature of the variation of solution time to network time constants and network natural frequencies. The first will be a simple two-stage RC circuit (curve (a) in Figure 17) in which the analysis is performed for different values of the predominant RC time constant. The solution time versus the problem time divided by the smallest time constant

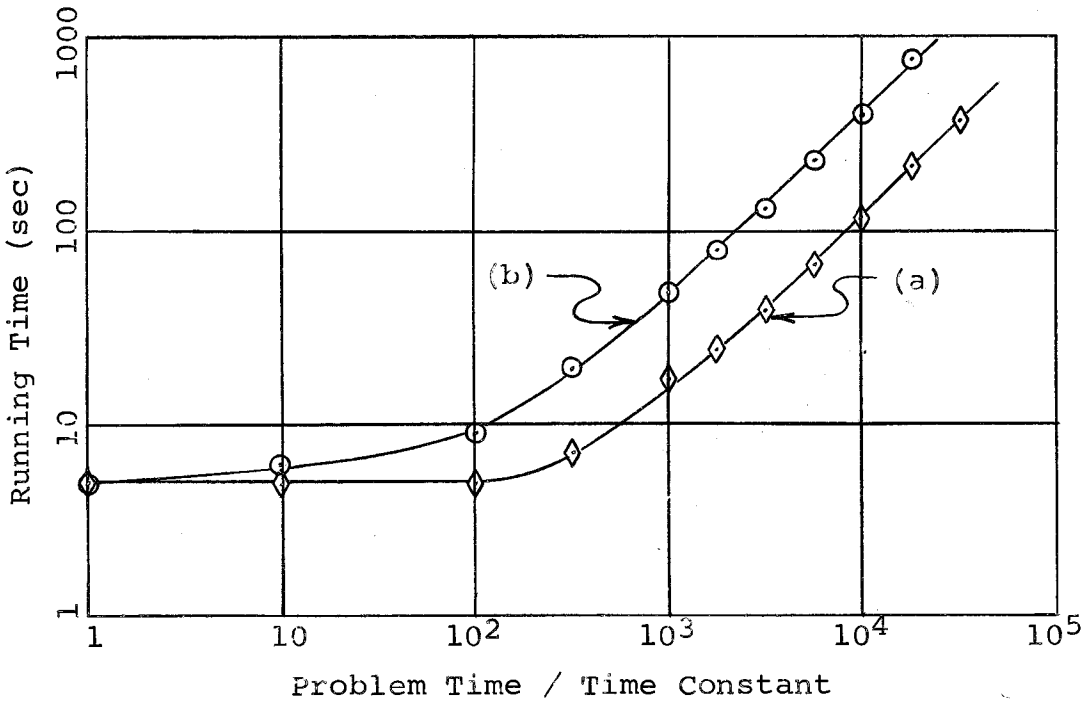


Figure 17. Computer running time versus the problem time divided by the smallest circuit time constant.

is shown in Figure 17.

For solution times less than approximately six seconds the majority of the time is taken up by the mechanics of the computer program, for example in the transfer of subroutine arguments in call and return statements. The solution time, for times greater than six seconds, exhibits a near linear relationship to the abscissa. From curve (a) of Figure 17 this relationship is

$$\text{Program running time} \propto \left[\frac{\text{Problem time}}{\text{Smallest time constant}}\right]^{0.9} . \tag{79a}$$

In Equation (79a) a proportionality exists rather than an equality because, in general, a network to be analyzed may contain any number of elements and nodes. In that case the expression could be generalized to a form

$$\text{Program running time} = k\left[\frac{\text{Problem time}}{\text{Smallest time constant}}\right]^{0.9} \tag{79b}$$

where k is different for each network and depends on the complexity and size of the network. Equation (79b) assumes the circuit contains a time constant that predominates over the other time constants (i.e., is less than) and over other circuit natural frequencies. Two values of k are depicted by curves (a) and (b) in

Figure 17. Curve (b) represents the analysis results for a network containing seven more branches than that for the network of curve (a). However, both networks have a single time constant that governs the analysis time. From the previous section it is expected the larger network, for the same value of minimum time constant, will require more analysis time which is the case for Figure 17. The important thing to note, however, is that both curves have the same slope indicating the same exponent for Equation (79b). The value of k is different though, for the two cases. In curve (a) $k = 0.023$ and in curve (b) $k = 0.097$.

When the smallest natural period is less than the smallest time constant, then the roots of the characteristic equation influencing the solution time the most are controlled by the natural frequencies of the circuit. The smallest natural period then determines the integration step size and consequently the solution time required for a given problem time. To illustrate this behavior a series LC circuit driven by a voltage step was analyzed for different values of the LC product. From the curve of Figure 18 the following empirical relationship can be obtained

$$\text{Program running time} \propto \left[\frac{\text{Problem time}}{\text{Smallest natural period}}\right]^{1.3} . \tag{79c}$$

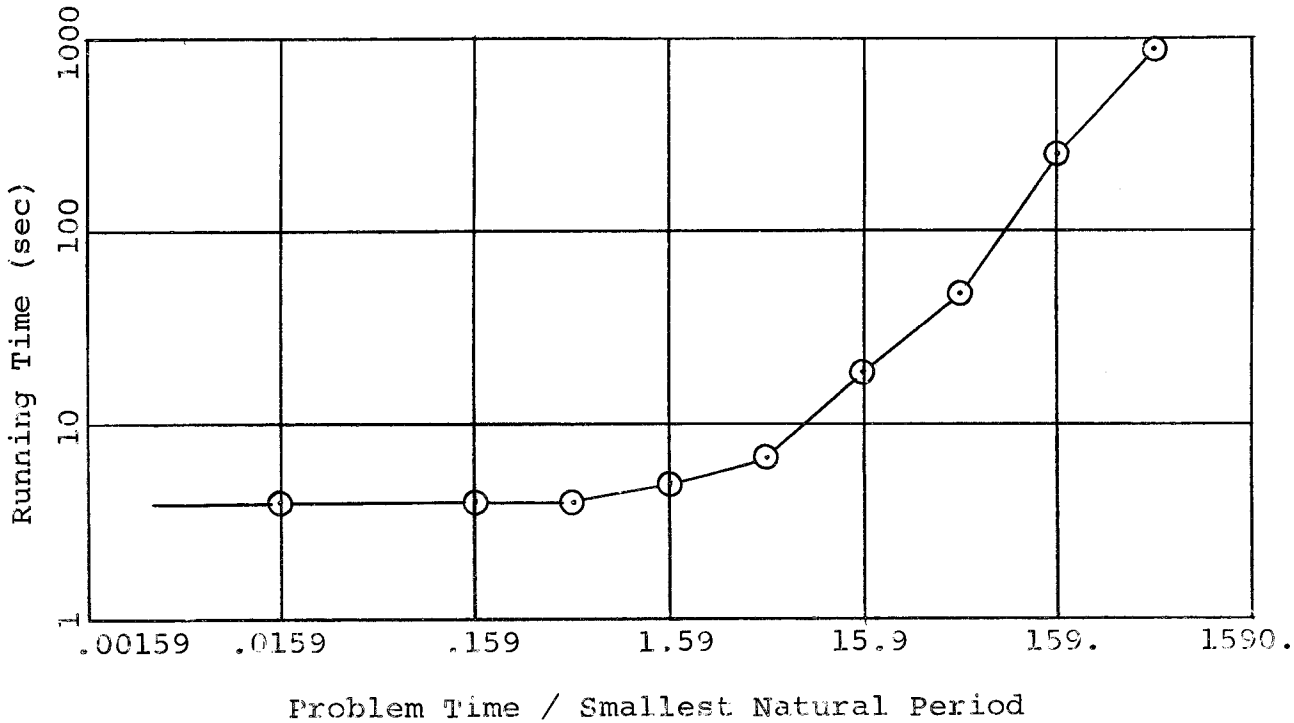


Figure 18. Computer running time versus the problem time divided by the smallest natural period.

Again this relationship could also be generalized to include circuit size and other effects. Instead Equations (79a) and (79c) will be expressed as a single equation; Equation (80), in which the value of the integration step size, for a given circuit, will be determined by whichever is smaller -- the smallest time constant or the smallest natural period.

$$\text{Program running time} \propto \left[\frac{\text{Problem time}}{\text{Smallest time constant or smallest natural period}}\right]^{1.1} \tag{80}$$

The exponent of Equation (80) is the average of the exponents from the previous examples. A value of unity for the exponent would be accurate enough for modeling purposes in most cases.

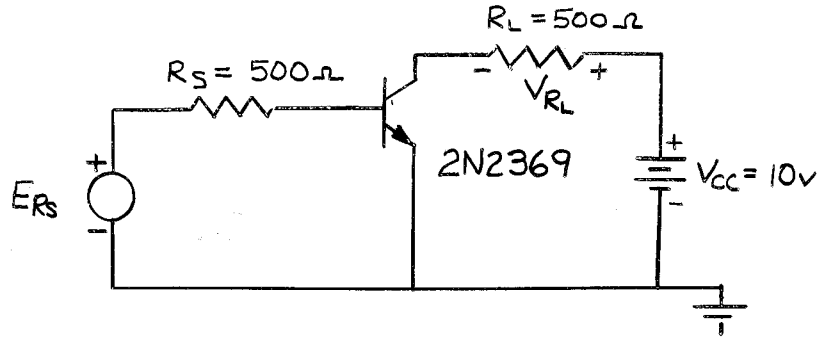
Large-Signal Transistor Model and Element Representation

Having established in the previous sections that the network size and the network time constants and natural frequencies are important parameters in determining the amount of computer time that will be used in performing an analysis, this section will experimentally relate transistor element representation to solution times. Again the solution time will be considered as the real time spent in solving for the network response for a

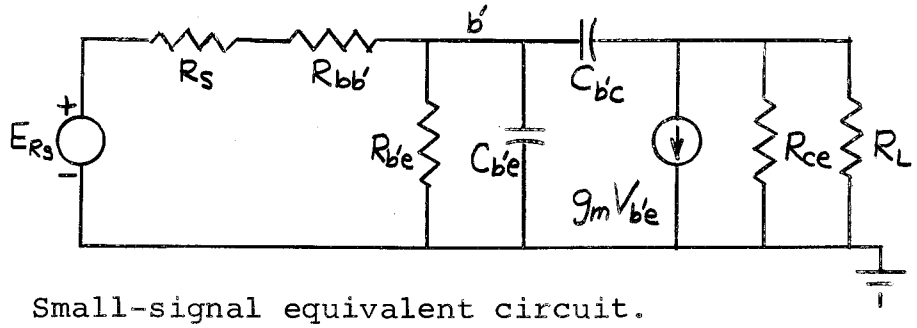
given problem time and solution accuracy. This time will not include the time consumed in reading in network data, formulating the network matrix equations, generating function tables for variables, or in reading the results out onto magnetic tape or other output devices.

As an example of the solution time differences in a small-signal analysis and a large-signal analysis, consider the circuit shown in Figure 19(a). The small-signal circuit was analyzed for a problem time of 75 ns using small-signal equivalent circuit data for a 2N2369 biased at I_C = 1ma. The analysis accuracy was held to one percent and the solution took 317 integration time steps in 28 seconds. In contrast to this the large-signal equivalent circuit was analyzed for a problem time of 75 ns and the same solution accuracy but required 327 integration steps and 42 seconds. These two analyses illustrate that some care is needed in modeling active devices to insure that the analysis will be accomplished in a reasonable amount of computer time.

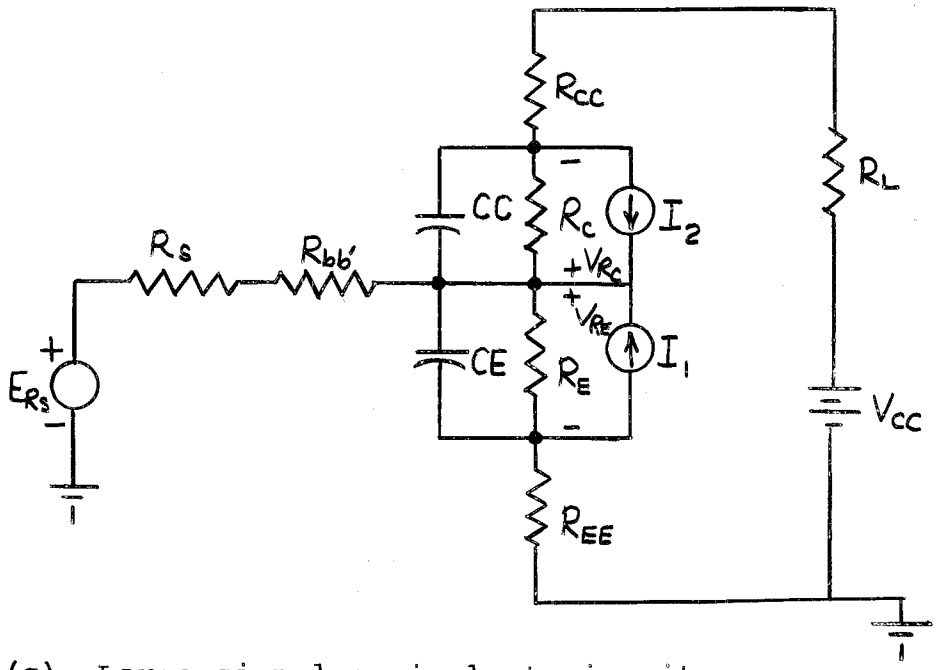
It was demonstrated earlier that in comparison to addition, subtraction, and transposition, matrix multiplication and inversion require far more time for a given matrix size. In the previous chapter it was shown that at each integration time step several algebraic matrix equations must be evaluated. Included among these



(a) Simple common-emitter inverter circuit.



(b) Small-signal equivalent circuit.



(c) Large-signal equivalent circuit.

Figure 19. Example for analysis.

equations are the following three equations that must be solved at each time step:

$$I_\beta = \left[R_{\beta\beta}+B_{\beta\sigma}R_{\sigma\sigma}B'_{\beta\sigma}\right]^{-1} \cdot \left[E_\beta +B_{\beta\lambda}E_\lambda +B_{\beta\sigma}E_\sigma -B_{\beta\lambda}V_\lambda - \left[B_{\beta\sigma}R_{\sigma\sigma}\right]\cdot\left[B'_{\gamma\sigma}I_\gamma +J_\sigma -B'_{\beta\sigma}J_\beta +B'_{\gamma\sigma}J_\gamma\right]\right] \quad (81)$$

$$\dot{I}_\gamma = \left[L_{\gamma\gamma}+L_{\gamma\mu}B'_{\gamma\mu}+B_{\gamma\mu}L_{\mu\gamma}+B_{\gamma\mu}L_{\mu\mu}B'_{\gamma\mu}\right]^{-1} \cdot \left[E_\gamma +B_{\gamma\lambda}E_\lambda +B_{\gamma\sigma}E_\sigma +B_{\gamma\mu}E_\mu - \left[B_{\gamma\lambda}V_\lambda +B_{\gamma\sigma}V_\sigma\right] + \left[B_{\gamma\mu}L_{\mu\mu}+L_{\gamma\mu}\right]\cdot\left[B_{\gamma\mu}\dot{J}_\gamma -\dot{J}_\mu\right]\right] \quad (82)$$

and

$$I_\alpha = \left[C_\alpha^{-1}+B_{\alpha\lambda}C_\gamma^{-1}B'_{\alpha\lambda}\right]^{-1} \cdot \left[\dot{E}_\alpha +B_{\alpha\lambda}\dot{E}_\lambda -B_{\alpha\lambda}C_\lambda^{-1}\left[B'_{\beta\lambda}I_\beta +B'_{\gamma\mu}I_\gamma -\left[B'_{\alpha\lambda}J_\alpha +B'_{\beta\lambda}J_\beta +B'_{\gamma\mu}J_\gamma\right] + J_\lambda\right]\right] . \quad (83)$$

These three equations represent the three cases where a matrix inverse must be taken in solving for the network response. In each expression the inverse involves the elements R, L, and C and the fundamental loop matrix. The fact that these are the only equations where the inverse is required is a result of the manner in which the problem was formulated.

When the resistive, capacitive, and inductive elements for a particular network to be analyzed are linear then the terms

$$\left[R_{\beta\beta}+B_{\beta\sigma}R_{\sigma\sigma}B'_{\beta\sigma}\right] ,$$

$$\left[L_{\gamma\gamma}+L_{\gamma\mu}B'_{\gamma\mu}+B_{\gamma\mu}L_{\mu\gamma}+B_{\gamma\mu}L_{\mu\mu}B'_{\gamma\mu}\right] ,$$

and

$$\left[C_\alpha^{-1} + B_{\alpha\lambda}C_\gamma^{-1}B'_{\alpha\lambda}\right] \quad (84\ a,b,c)$$

need only be inverted once during the entire analysis. However, when the network contains non-linear resistors, capacitors, and inductors the above terms must be inverted at each time step or solution pass. This fact means that for large circuits, the non-linear passive elements will greatly affect the amount of computer time the analysis requires. For this reason it is highly desirable to avoid the use of non-linear passive elements in the network representation if possible.

The network in Figure 19(c) will be used to illustrate the effects of when non-linear resistors are present in the circuit. In both analyses the problem time was 75 ns and the solution accuracy one percent. Non-linear resistors require a matrix inversion at each solution step but in many cases a non-linear resistor may be transformed to a voltage dependent current source. A voltage or current source representation does not require a matrix inversion, as is easily verified by inspecting the summary of solution equations in the preceding chapter.

In illustrating this effect by the circuit of Figure 19(c), the current source I_1, which represents the non-linear behavior of the emitter-base junction, will be represented by a non-linear resistor. The capacitors CE and CC will be analytic expressions as in

Equations (87a,b).

The source I_1 has the form

$$I_1 = -2.61\text{x}10^{-6}(e^{40.1\text{xVRE}}-1)+4.01\text{x}10^{-6}(e^{29.4\text{xVRC}}-1). \quad (85)$$

The large-signal resistance (R'_E) of the junction becomes then

$$R'_E = \frac{VR_E}{I_1} = V_{R_E} \Big/ \left[2.61\text{x}10^{-6}(e^{40.1 \cdot V_{R_E}}-1) - 4.01\text{x}10^{-6}(e^{29.4V_{R_C}}-1)\right] . \quad (86)$$

The results of the analyses, in terms of the solution times, were

TABLE II. SOLUTION TIME AND E-B JUNCTION REPRESENTATION

E-B Junction Representation	Problem Time	Integration Steps	Solution Time(sec)
Voltage dependent current generator	75 ns	327	42
Variable resistor	75 ns	414	107

It is obvious that the representation of non-linear resistive elements by voltage dependent current sources will greatly speed the solution process even when the circuit is of limited size.

In the solution process, after the state variables are updated through numerical integration, the variable

elements are then updated prior to the next algebraic matrix operations. The variable elements are, in general, either time variable or network voltage or current dependent, or perhaps a function of both time and voltage or current. Updating these elements can consist of a table lookup procedure or an equation solving process. With efficient programming either process will require approximately the same amount of time in terms of total computer running time.

The same effect, i.e., increasing the solution time by variable elements, could also be illustrated by the emitter and collector capacitances. However, as depicted by Figure 15, the matrix order must be larger than two before the effects of matrix inversion times become significant. In place of the non-linear representation of the elements, the time constants become the controlling influence on the solution time.

Nevertheless it is worth while investigating the variation of CE and CC of Figure 19(c) to see the effect on the circuit response. While a sensitivity factor could be defined relating circuit response to parameter variation, the intent of this report is to investigate the influence on computer running times for device characterization; therefore the intent here is only in showing that considerable care must be taken in the

modeling of active devices.

When CE and CC were expressed as variables the following analytical expressions were used:

$$CE = \frac{3.7}{(1.1-VRE)^{.34}} + 3.32\times10^{-5}\ e^{40.1\times VRE}$$

$$CC = \frac{3.3}{(1.1-VRC)^{.10}} + 1.58\times10^{-3}\ e^{29.4\times VRC} \quad . \qquad (87a,b)$$

The values are from measurements on 2N2369 transistors.

Both CE and CC were represented as constants while the other capacitor was expressed analytically by either Equation (87a) or (87b). The values for each capacitor and the resulting solution times are given in Table III.

TABLE III. CAPACITOR VALUES FOR CE AND CC VERSUS SOLUTION TIMES

CE Constant			CC Constant		
CE	CC	Solution Time	CE	CC	Solution Time
Analytic	Analytic	42 sec.	Analytic	Analytic	42 sec.
100	"	48	"	5	41
50	"	80	"	3	55
10	"	368	"	2	70

Aside from the increased solution times due to smaller time constant values in each case, the circuit response and its sensitivity on parameter value is interesting. Figure 20 shows the circuit response for the three cases where CE is the element varied. For

this circuit the delay is the principal response parameter affected by the emitter-base capacitance. Figure 21 reveals that the collector-base capacitance primarily controls the large-signal risetime for the circuit of Figure 19(c).

Not only do Figures 20 and 21 show the extreme care that is necessary in modeling active devices accurately, but that the networks can be very sensitive in terms of running time to the element representation and element values.

Extension to Integrated Circuit Analysis

Although the preceding material has had for its framework the modeling of transistors and transistor circuits, it is directly applicable to other active devices and circuit elements. The only restrictions placed on the nature of the circuit to be analyzed occur in the formulation of the analysis procedure itself. In chapter two the analysis problem was based on ordinary RLC passive elements and VI sources. Therefore any analysis must also be formulated with similar elements. As an example of the types of problems one encounters with integrated circuits consider the cross-sectional view of a diffused resistor for hybrid integrated circuits depicted in Figure 22(a).

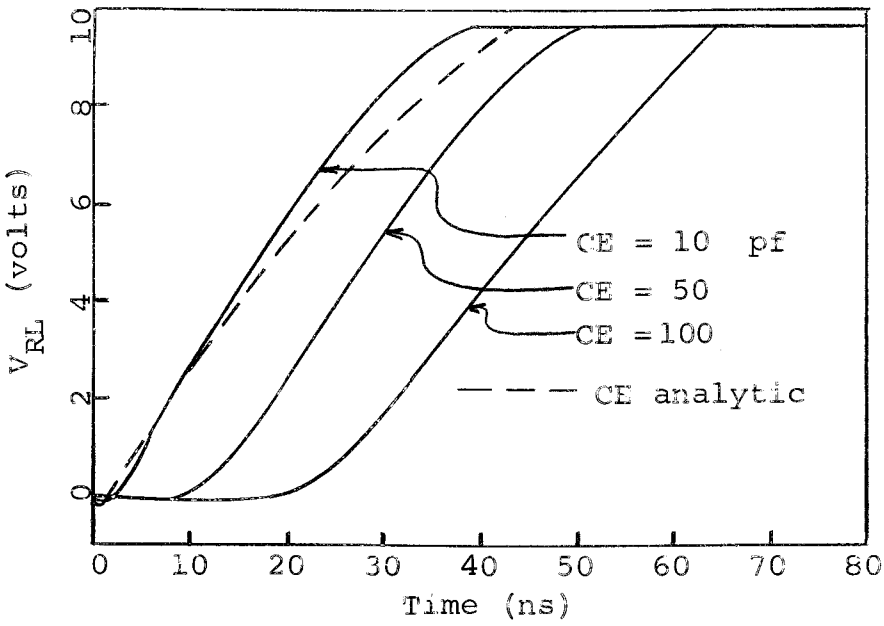


Figure 20. Response sensitivity to changes in CE.

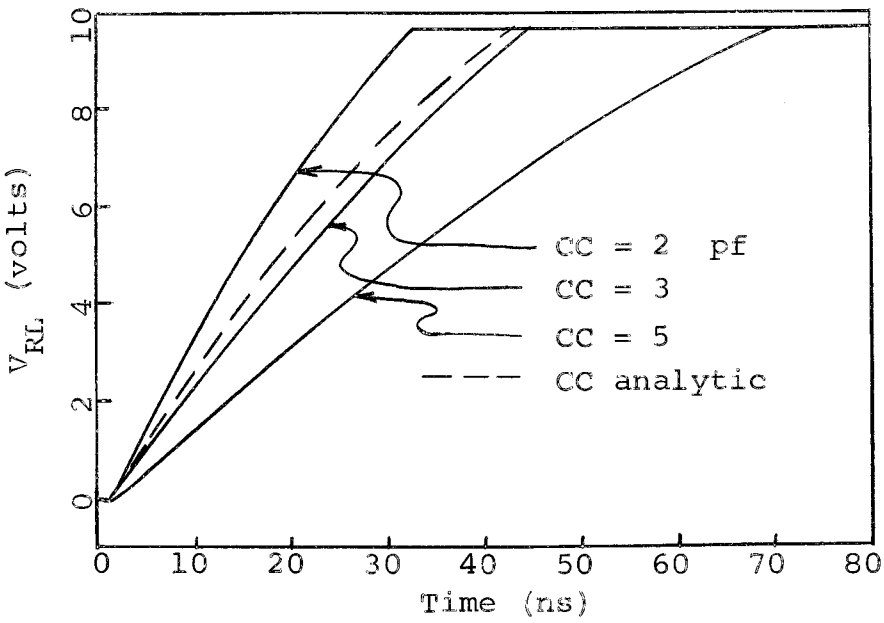
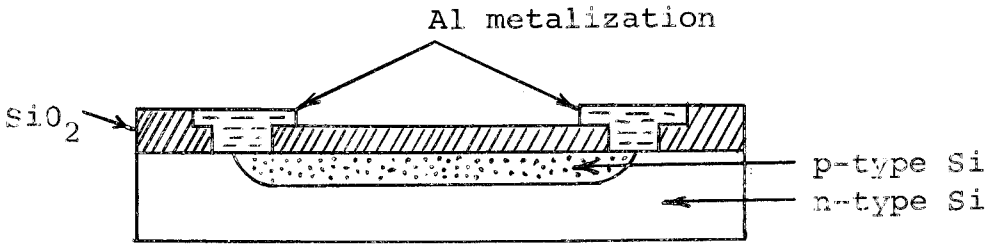
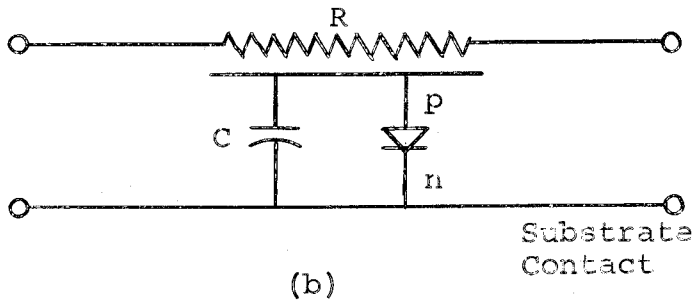


Figure 21. Response sensitivity to changes in CC.



(a)

Cross-sectional view of diffused resistor for hybrid integrated circuits.



(b)

Equivalent circuit of diffused resistor for hybrid integrated circuits.

Figure 22. Diffused resistor for integrated circuit.

An equivalent circuit for this resistor is shown in Figure 22(b) where, in addition to the resistor, a distributed capacitance and diode must be associated with the resistor due to the substrate material. Among other effects the distributed capacitance will form a distributed shunt across the resistor at high frequencies. This distributed capacitance may be represented by a series of resistor-capacitor stages, or both the resistor and capacitor may be modeled by analytical

expressions which express their distributed effect directly.

The point is that integrated circuits may be analyzed by general network analysis techniques provided adequate equivalent circuits are used in which the elements are expressible as RLC elements. Several approximations may be necessary in such equivalent circuits. Narud and Meyer (21) have discussed the characterization of integrated circuits and the nature of these approximations.

Chapter Summary

In this chapter several aspects of equivalent circuit representation have been presented and related to computer solution times for general network time domain analysis programs. Network size, network time constants and natural frequencies, and network element representation have been related to the amount of computer time required to solve for the network response. These relationships are a result of the mathematical formulation of the general network analysis problem as applied to computer programs. It is not possible to establish exact relationships or equations to predict how much computer time will be required for an analysis of a network of a given size because of the great number of

possible element representations, network connections, and branching paths in any particular solution. It is possible, however, to generalize from a few simple analysis examples and formulate guidelines in modeling that, in most cases, will be valid. This will be accomplished in the next chapter.

V. CONCLUSIONS

Important Results

In the foregoing chapters the primary purpose has been to establish a relationship between the modeling process (especially large-signal transistor equivalent circuits) and the models' effect on the amount of computer time that would be required by a general time domain network analysis computer program. In relating computer solution times to the model, both the mathematical expressions for the solution and experimental results show that three considerations are important and should be considered in modeling to avoid excess solution times. They are:

(1) The network size or in other terms the order of the matrices describing the network topology;

(2) The network time constants and natural frequencies; and

(3) The representation of non-linear equivalent circuit elements.

For networks consisting of linear elements a measure of the effect of circuit size on the resulting computer time is shown in Figure 16. The amount of computer time varies, to a good approximation, as the second power of

the number of nodes. In circuits containing non-linearities this relationship will tend to have higher powers due two factors: (1) the need to update variable elements at each integration step, and (2) the extra amount of time required for matrix inversion when variable resistors, capacitors, or inductors are present.

Figure 17 and 18 show that the smallest natural periods or the smallest time constants are of fundamental importance in determining the amount of computer time that will be required for a specified problem time. This is a result of the necessity for keeping the integration step size small enough to avoid numerical instability.

Tables II and III illustrate the importance of element representation. In particular, when circuit elements are expressed as variables it becomes necessary to take the inverse of the solution matrices at each time step, thus increasing the solution time.

Guidelines in Modeling

From the results of the preceding chapter several guidelines can be established as an aid in modeling active devices and network elements for computer time domain analysis. In particular, the specific purpose of the guidelines is to help avoid equivalent circuits that will force a general purpose transient analysis computer

program to use an excessive amount of computer time for a given problem time. These guidelines will be stated as RULES but it should be understood they are rules only in a general sense.

RULE 1: Try to limit the size of the circuit.

For the PREDICT circuit analysis program the following approximation holds:

$$\text{Running time} = a \cdot (\text{Number of nodes})^2 \text{ (sec)} \quad (88)$$

where

$$a \approx 3.7 \text{ sec} \quad (89)$$

for reasonable amounts of computer time on the IBM 7094 computer (reasonable would be about ten minutes). Another interpretation of Rule 1, besides the obvious interpretation, is that the result of the analysis should be insight to the problem posed for analysis, not merely numerical answers. Often the same problem expressed in simpler terms or divided into sub-networks can provide more insight.

RULE 2: Try to avoid large spreads in circuit time constants or wide ranges in network natural frequencies.

As shown in chapter four, the running time for a general transient analysis program with a variable integration period is:

$$\text{Running time} \propto \left(\frac{\text{Problem time}}{b}\right) \text{(sec)} \qquad (90)$$

where b can be either the smallest time constant of the circuit or the reciprocal of the highest natural frequency. Again, in terms of computers in the 7094 class (operating cycle time = 2 microseconds) this means when

$$\left(\frac{\text{Problem time}}{b}\right) < 6\text{x}10^4 \qquad (91)$$

the machine time will be reasonable (less than ten minutes for an IBM 7094).

RULE 3: Try to avoid variable R,L,C elements.

Variable passive elements require a matrix inversion at each integration step in the solution process. This rule becomes especially important as the size of the network increases (see Figure 15).

Suggestions for Future Work

It is only relatively recent that digital computer programs capable of performing transient analyses of large, non-linear networks have been available. Notable among those programs that are available is the NET-1 program (20) and the PREDICT program (32). Several groups are presently working on transient analysis programs (26,30) and it is expected that considerable further work will be supported in this area.

One of the primary difficulties in transient analysis programs is speed. Although much can be gained in program refinements in the areas of integration methods and more efficient matrix operations, what is needed is orders-of-magnitude increases in the program speeds. Several approaches have been taken towards improving the speed of solutions. One method is to integrate the linear and non-linear sets of differential equations separately and join the solutions periodically. This has been successfully used in transient analysis programs (3,5) but could be further developed.

Certaine has developed a transformation method of numerical integration that uses time-dependent relaxation methods that shows promise of reducing solution times of circuits with mixed time constants (25, p. 128-132). This method, however, has been found to present mathematical difficulties with many types of circuits, making it less powerful than was hoped (33, p. 13).

The problem of speeding computation has been, for the most part, directed towards reducing the amount of work to be done by the program, that is, to find algorithms which produce some result in the least number of operations, or to improve the rate of convergence of a given algorithm for a given accuracy. Another approach has been proposed by Neivergelt (22) which improves the

speed of solution by parallel computation. Redundancy in computation is introduced so that several subtasks can be performed in parallel and then joined at intervals. This approach could conceivably lead to faster network analysis programs.

Besides the general area of improvement of program speeds as an area for further study another topic is suggested. It is the area of active device modeling for large-signal analysis. At present the analysis approach is usually either through the Ebers and Moll model or the charge-control model to an estimate of circuit speeds, delays, and stability. It is very difficult to estimate the relationship between parameter values and resulting response for the non-linear case. It would be worth while to study these interrelationships in more detail. Perhaps meaningful equivalent circuit element sensitivity factors could be established for various circuit configurations.

BIBLIOGRAPHY

1. Ashar, K. G. et al. Transient analysis and device characterization of ACP circuits. IBM Journal of Research and Development 17:207-223. 1963.

2. Bashkow, T. R. The A matrix, a new network description. Institute of Radio Engineers Transactions on Circuit Theory CT-4:117-120. 1957.

3. Beaudette, J. H. and P. A. Honkanen. PE CANS, circuit analysis of non-linear systems. Poughkeepsie, International Business Machines Corporation, n.d. 17p. (Scientific Computations Department).

4. Beaufoy, R. and J. J. Sparkes. The junction transistor as a charge-controlled device. Automatic Telephone and Electric Company Journal 13:310-327. 1957.

5. Branin, Franklin H., Jr. D-C and transient analysis of networks using a digital computer. 1962 Institute of Radio Engineers International Convention Record Part 2:236-256. 1962.

6. Branin, Franklin H., Jr. The relation between Kron's method and the classical methods of network analysis. 1959 Institute of Radio Engineers WESCON Convention Record Part 2:3-28. 1959.

7. Brown, David P. Derivative-explicit differential equations for RLC graphs, Journal of the Franklin Institute 275:503-514. 1963.

8. Bryant, P. R. The explicit form of Bashkow's A matrix. Institute of Radio Engineers Transactions on Circuit Theory CT-9:303-306. 1962.

9. Bryant, P. R. The order of complexity of electrical networks. Proceedings of the Institute of Electrical Engineers 106:174-188. 1959. (Part C, Monograph 335E).

10. Chase, P. E. Stability properties of predictor-corrector methods for ordinary differential equations. Journal of the Association for Computing Machinery 9:457-468. 1962.

11. Domenico, R. J. Simulation of transistor switching circuits on the IBM 704. Institute of Radio Engineers Transactions on Electronic Computers EC-6:242-247. 1957.

12. Ebers, J. and John L. Moll. Large-signal behavior of junction transistors. Proceedings of the Institute of Radio Engineers 42:1761-1772. 1954.

13. Falk, Howard. Computer programs for circuit design. Electro-Technology 73:101-164. Mar. 1964.

14. Gray, P. E. et al. Physical electronics and circuit models of transistors. New York, Wiley, 1964. 262 p.

15. Hamilton, Douglas J., Fred A. Lindholm, and Jan A. Narud. Comparison of large-signal models for junction transistors. Proceedings of the Institute of Electrical and Electronics Engineers 52:239-248. 1964.

16. Hamming, Richard W. Numerical methods for scientists and engineers. New York, McGraw-Hill, 1962. 212 p.

17. Hildebrand, F. B. Introduction to numerical analysis. New York, McGraw-Hill, 1956. 511 p.

18. Kuh, Ernest S. and Ronald A. Rohrer. The state-variable approach to network analysis. Proceedings of the Institute of Electrical and Electronics Engineers 53:672-686. 1965.

19. Linvill, John G. Lumped models of transistors and diodes. Proceedings of the Institute of Radio Engineers 46:949-957. 1958.

20. Malmberg, Allan F. and Fred L. Cornwell. NET-1 network analysis program. Los Alamos, Los Alamos Scientific Laboratory, Sept. 1964. II-31 p. (Los Alamos Scientific Laboratory Report LA-3119).

21. Narud, Jan A. and Charles S. Meyer. Characterization of integrated logic circuits. Proceedings of the Institute of Electrical and Electronics Engineers 52:1551-1564. 1964.

22. Nievergelt, J. Parallel methods for integrating ordinary differential equations. Communications of the Association for Computing Machinery 7:731-733. Dec. 1964.

23. Phillips, Alvin B. Transistor engineering. New York, McGraw-Hill, 1962. 373 p.

24. Pritchard, R. L. Electric-network representation of transistors - a survey. Institute of Radio Engineers Transactions on Circuit Theory CT-1:5-21. 1956.

25. Ralston, A. and H. S. Wilf (eds.) Mathematical methods for digital computers. New York, Wiley, 1960. 293 p.

26. Reid, Richard J. A system analysis program. Poughkeepsie, International Business Machines Corporation, n.d. 23 p. (Product Development Laboratory Technical Report No. 1).

27. Searle, C. L. et al. Elementary circuit properties of transistors. New York, Wiley, 1964. 306 p.

28. Seshu, Sundaram and Norman Balabanian. Linear network analysis. New York, Wiley, 1959. 571 p.

29. Seshu, Sundaram and Myril B. Reed. Linear graphs and electrical networks. Reading, Addison-Wesley, 1961. 315 p.

30. Warten, R. M. Automatic step-size control for Runge-Kutta integration. IBM Journal of Research and Development 7:340-341. 1963.

31. Wilson, R. L. and W. A. Massena. An extension of Bryant-Bashkow A matrix. Institute of Electrical and Electronics Engineers Transactions on Circuit Theory CT-12:120-122. 1965.

32. U.S. Air Force Weapons Laboratory. Automatic digital computer program for determining response of electrical systems to transient nuclear radiation. Kirtland AFB, New Mexico, 1964. 128 p. (Technical Documentary Report No. WL-TDR-64-62 Vol. II Predict circuit analysis program).

33. U.S. Air Force Weapons Laboratory. Manual for circuit analysis and design-TREE. Kirtland AFB, New Mexico, 1964. 137 p. (Technical Documentary Report No. WL-TDR-64-60 Vol. I).