STATIC AND DYNAMIC ANALYSIS OF PRO-

PORTIONAL FLUIDIC CIRCUITS

By

BEEGAMUDRE NAGABHUSHANA RAO MURALI

Bachelor of Engineering
University of Mysore
Mysore, India
1967

Master of Engineering
Indian Institute of Science
Bangalore, India
1969

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
DOCTOR OF PHILOSOPHY
December, 1973

STATIC AND DYNAMIC ANALYSIS OF PRO-

PORTIONAL FLUIDIC CIRCUITS

Thesis Approved:

Karl N. Reid

Thesis Adviser

Jerald D Parker

Charles M. Bacon

Henry R Sebesta

D D Durham

Dean of the Graduate College

ACKNOWLEDGMENTS

My graduate studies at Oklahoma State University would not be complete without proper acknowledgments to the many individuals who helped make my experience as a graduate student valuable and meaningful. I thank the entire faculty and staff of the Mechanical and Aerospace Department for maintaining the excellent atmosphere so essential for one's academic and professional development.

The individual most instrumental in my selecting OSU for graduate studies, and whom I greatly respect, is my major adviser, Professor Karl N. Reid. I am indebted to him for the invaluable guidance and encouragement he has provided throughout this study and many projects. His advice on the preparation of this thesis and other reports has been of great help to me.

I thank Professors H. R. Sebesta, J. D. Parker, and C. M. Bacon for serving on my thesis committee. Their valuable advice and criticisms are greatly appreciated.

Dr. Ganesh Rajagopal, as a personal friend, patiently discussed many problems of mutual interest. I wish to express my sincere thanks to him.

As a graduate student, I have received many stimulating ideas and words of encouragement from my collegues, Lynn Ebbesen, Ed Gostling, Syed Hamid, Jerry Robinson, and Mike Bowles. I wish to thank them all.

I am especially grateful to my parents without whose encouragement this graduate study would not have been possible. I thank Mrs. Barbara

Moore for her cooperation in typing the manuscript.

The work reported in this thesis was sponsored by the Center for Systems Science which is supported in part by the National Science Foundation under the grant GU-3160.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure		Page

CHAPTER I

INTRODUCTION

Background

The potential for applications of proportional fluidic components and circuits in the areas of control, computation, and sensing has been recognized for several years. Yet, with the exception of process controllers and a variety of computation circuits for some industrial and some aircraft and missile applications, proportional fluidic circuits have not been used nearly to the extent originally envisioned.

One of the primary deterrents to more widespread application of proportional circuits, appears to be related to the inherent complexity of the components. Most real fluidic components are highly nonlinear. Passive components do not actually exist which are pure, i.e., which are governed by a single energy storage or energy dissipation mechanism. Although nonlinear static and dynamic models exist for most passive components, adequate nonlinear dynamic models do not exist for active components.

Additionally, efficient procedures and tools for the static and dynamic analysis of proportional fluidic components and circuits do not exist. The practical design of these circuits is accomplished presently either by trial and error experimental methods or by hand calculation methods. Such methods are at best tedious, time consuming, and expensive; all are impractical for complex circuits. Moreover, the skill level required in analysis and synthesis prohibits most designers from even trying.

It seems clear that user-oriented, computer-aided analysis and synthesis tools are needed which can handle the dominant nonlinearities in fluidic circuits. In addition, some new or refined component models are needed which are compatible with these tools.

Scope and Results of Dissertation

This dissertation discusses the development and application of two general purpose, user-oriented computer programs for the analysis and synthesis of proportional fluidic circuits. The first program, termed the Fluidic Circuit Analysis Program (FCAP), allows a rapid and accurate determination of the static (D.C.) and dynamic (transient) performance of any complex proportional fluidic circuit. Included within FCAP is a component library which includes a number of common active and passive elements. A procedure is provided for adding new components or refined component models to the library. Both active and passive components are represented by generalized multi-port static and dynamic models. These models may be analytically derived or they may be based upon tabulated data derived from experimental measurements or manufacturer's catalog sheets. A procedure is presented whereby a phenomenological model may be derived to describe the nonlinear behavior of a component in case an adequate analytical (or empirical) model does not exist. This procedure is illustrated through the derivation of a nonlinear dynamic model for proportional amplifier based on measured static characteristics at the various ports; the result is believed to be of singular importance.

A static Network Optimization program (NETOPT) is presented also.

This program allows the determination of the optimum parameters of a fluidic circuit which minimize the power supplied to the active components, while the operation of the circuit meets certain parameter and output constraints. Parameters which may be optimized include: 1) supply pressures to the active components, 2) resistances of passive linear resistors, and 3) cross-sectional areas of certain nonlinear resistors.

Although both computer programs, FCAP and NETOPT, are intended for use in the design of proportional fluidic circuits, they may be adapted easily for use with mechanical, electrical and electronic circuits by suitable expansion of the component library. For some fluidic circuit analysis purposes, FCAP is definitely superior to other well known circuit analysis programs, e.g., ECAP, SCEPTRE, NET-1, etc.

CHAPTER II

LITERATURE SURVEY

Review of Analysis Techniques Used in Electrical Networks

With the introduction of electronic computers, considerable progress has been made in the analysis of electrical networks. Many computer-aided circuit analysis programs have been developed which perform DC, AC, transient, and in some cases, sensitivity and statistical analyses [1, 2]. Most of the programs include equation compilers; that is, the programs generate the necessary equations from node to node and element-type description of the network. Generally, the circuit description input is user-oriented. Some examples of user-oriented input language analysis programs are: NET-I [3], SCEPTRE [4], CIRCUS [5], CALAHAN [6], ECAP [7], CIRCAL [8], and AEDNET [9]. A few major features and capabilities of more general network analysis programs are summarized in Table I. Chua and Medlock [10] have developed computer program MECA to analyze nonlinear resistive networks. The program uses a piecewise-linear method to analyze multivalued electronic networks. One of the limitations of the program is that the models have to be in the canonical form. Walden [11] has presented a computer program for the dynamic analysis of nonlinear electronic circuits. A linear network analysis technique has been adapted to include the analysis of nonlinear networks. Nonlinear storage elements are represented on a piecewise-linear basis,

TABLE I

MAJOR FEATURES AND CAPABILITIES OF SOME CIRCUIT ANALYSIS PROGRAMS [1]

	CALAHAN	CIRCUS	ECAP	NET-1	POTTLE	PREDICT	SCEPTRE
Program input features							
Must use specific labels	No	Yes	Yes	Yes	Yes	Yes	Yes
Sequential node numbering	Yes	Yes	Yes	Yes	Yes	No	No
Voltage or current sources	Current Sources Only	Yes	Yes	Through a modification of the built-in model (not convenient)	Yes	Yes	Yes
Tabular input for branch descriptions	No	No	No	No	No	Yes	Yes
Tabular input for signal sources	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Analytic description of branch elements	No	No	No	No	No	Yes	Yes
Analytic description for signal sources	Yes (Available from a modified sub routine)	No (has several built-in signal sources)	No	No (has several built-in signal sources)	No	Yes	Yes
Automatic modifications of input (repeated runs)	Yes	No	Yes	Yes	No	No	Yes

TABLE I (Continued)

	CALAHAN	CIRCUS	ECAP	NET-1	POTTLE	PREDICT	SCEPTRE
Modeling capabilities							
Built-in models	No	Yes	No	Yes (transistor diodes	No	No	Yes
Allows small-signal models	Yes	No	Yes	No (inconvenient)	Yes	Yes	Yes
Allows large-signal models	No	Yes (built-in)	Yes (inconvenient)	Yes (built-in)	No	Yes	Yes
Output options							
Transfer functions	Yes	No	No	No	Yes	No	No
Pole-zero locations	Yes	No	No	No	Yes	No	No
Symbolic expression for time response	Yes	No	No	No	Yes	No	No
Time-response output	Output node only- at fixed intervals	Node voltages and semiconductor voltages & currents	Selected variables at fixed intervals	Node voltages & semiconductor currents a at fixed intervals	Output node only at fixed intervals	Selected branch voltages & currents at every time step	Selected variables at fixed intervals
Steady-state solution	No	Yes	Separate analysis	Yes	No	Separate analysis	Yes
Programming features							
Programming language	FORTRAN II, IV	FORTRAN IV	FORTRAN II, IV	MAP & FAP	FORTRAN II, 63	FORTRAN II & FAP	FORTRAN IV
Memory capacity recommended	32,000	32,000 (uses overlays)	32,000	32,000 (chained)	32,000	32,000 (chained)	32,000 (uses overlays)

TABLE I (Continued)

	CALAHAN	CIRCUS	ECAP	NET-1	POTTLE	PREDICT	SCEPTRE
Network formulation	Topological	State-variable	Nodal	State-variable	State-variable	State-variable	State-variable
Primary integration routine	Runge-Kutta (& inverse Laplace)	Exponential	Implicit numerical integration	Predictor corrector	Runge-Kutta (& inverse Laplace)	Trapezoidal	Exponential

and nonlinear resistors and active elements are represented by dependent drivers.

A great deal of effort has been spent in developing user-oriented computer programs for the analysis of electrical networks. However, none of these programs can be used directly for the analysis of fluidic networks because of either the type of network formulation or other restrictions peculiar to the program. The tree-finding problem is the basic limitation of programs using a topological approach to network analysis. A practical network may have several hundred thousand trees, and with the present algorithms a prohibitive amount of computer time is used to find the trees. ECAP uses a piecewise-linear approximation to simulate nonlinearities in active devices and parameters. Nonlinearities in circuit responses are incorporated through use of switches which produce piecewise-linear time responses. This method of representation of nonlinearities becomes too involved and complicated if used to describe the characteristics of fluid amplifiers, which are generally functions of more than one variable. With the exception of SCEPTRE and PREDICT, other network analysis programs have a basic limitation in that they cannot handle tabulated data or analytical models for describing circuit branches [1].

The structure of SCEPTRE has some inherent limitations which limit its use for the analysis of fluidic circuits. The SCEPTRE program allows only a maximum of fifty state variables; the program modifications needed to increase this number are quite involved. This is a major limitation on the size of the fluidic circuit that can be analyzed using SCEPTRE, since even a simple model for a fluid transmission line may involve as many as eighteen state variables. Another limitation of

SCEPTRE is that there is no provision for integrating the differential equations of component models supplied through user written subprograms. These equations have to be integrated by the user, and the values of the static variables furnished by SCEPTRE. This calls for a considerable knowledge of both FORTRAN programming and numerical integration techniques on the part of the user.

Review of Analysis Techniques Used in Fluidics

The development of analysis techniques in fluidics technology has followed much the same pattern as those in other technologies, especially those in electronics. Many of the linear analysis methods used in electrical networks have been adapted for fluidics with fair success. Fluidic elements, unlike their electronic counterparts, are highly nonlinear. This limits the linear analysis techniques to only small signals around the operating point. The fluid circuit theory is more complicated than that of electrical circuit theory, and it has been pointed out by Kirshner [12] that there are no analogs of Kirchoff's circuit equations for the general case of fluid flow. The Kirchoff laws assume that there are no resistive or storage effects at a junction. This, in general, is not true for junctions in a fluid circuit. However, for cases where the fluid velocities are small and the temperature effects are negligible, the analogs of Kirchoff's circuit equations may still be used.

Most of the effort in developing techniques for the static analysis of fluid systems has been directed toward developing analysis methods for cascading and impedance matching of amplifiers. Lechner and Wambsganns [13] described the input and output characteristics of fluid amplifiers and their significance to the cascading problem. Dexter [14] described an "admittance" parameter for use in classifying and cascading components. Brown [15] applied a graphical technique to match a fluid amplifier with a load. Katz and Dockery [16] examined in detail the generalized performance characteristics for both proportional and digital amplifiers, and developed empirical relations for the input and output characteristics. They applied these characteristics to the analysis of two amplifiers in cascade. Shinn and Boothe [17] also described the problem of static matching of cascaded fluidic devices using the graphical approach to the static analysis of fluidic systems. Parker and Addy [18] developed a linear static matching method for analog fluidic circuits based on a particular type of proportional fluid amplifier. Interstage matching is achieved by the use of linear capillary resistances. In control systems design where gain is specified, the method described is useful for determining linear operating regions and matching resistances.

Unlike the development of static analysis techniques, which seems to be in only one direction (graphical approach), the progress in the development of dynamic analysis techniques has been varied. Belsterling and Tsui [19] took the initial step in the development of a method for the dynamic analysis of fluidic systems when they reported the "equivalent-circuit approach" for the small signal dynamic analysis. Brown [20] presented a frequency domain analysis technique for predicting the stability of fluid systems. Two methods were outlined; one employed admittance matrices, and the other used the scattering variables. Boothe [21] presented a lumped-parameter technique for predicting the dynamic performance of a proportional fluid amplifier. The dynamic analysis was based on normalized static characteristics of the amplifier. As a result, it was possible to predict the dynamic performance of the amplifier from simple static performance information. Healey [22] extended Boothe's work to include the effects of vent fluid, the inertia and friction of which provide additional dynamic lags. Manion [23] reported an analog computer simulation of a proportional fluidic amplifier which illustrated one more powerful technique for the dynamic analysis of fluidic systems.

In summary, the survey of the literature reveals the following:

1. No nonlinear models are known to exist which describe the behavior of active proportional fluid amplifiers.

2. The graphical load-line method and the experimental methods are the only available techniques for nonlinear analysis of the static performance of fluidic circuits. These methods are very time consuming and become involved as the complexity of the circuit increases.

3. The Belsterling, Brown, and Boothe techniques of dynamic analysis of fluidic circuits are limited to small signals. At present, no efficient, large signal, (nonlinear) dynamic analysis techniques are known to be available for the analysis of fluidic circuits.

CHAPTER III

STATIC ANALYSIS

Static analysis is concerned with two problems: 1) determination of the operating point of the circuit, and 2) determination of the linearized gain near a given operating point. The operating point determination consists of solving the resistive network for the given set of inputs and finding the values of all unknown port variables. The resistive circuit may consist of either linear or nonlinear elements which may be active or passive. Linearized gain analysis involves the determination of gain between any two specified points in the circuit for small signal changes.

A user-oriented Fluidic Circuit Analysis Program, FCAP has been developed for the static and dynamic analysis of fluidic circuits. The formulation and the computational procedure for the static analysis problem is discussed in this chapter. The dynamic analysis of fluidic circuits using FCAP is presented in Chapter IV.

The determination of the operating point consists of three basic steps. The first step is to describe the static behavior of the components; the next step is to formulate the system equations using the laws of interconnections and the component models; and the third step is to solve the resulting algebraic equations. The linearized gain analysis involves the determination of the operating point twice; first with the nominal values for the input, and then with the inputs perturbed. The gains may then be calculated by dividing the change in the port variables by the corresponding change in the input. These steps are explained next with particular reference to the analysis of fluidic systems.

Component Models

There are only two basic types of components that need to be considered for the static analysis. They are: 1) passive resistors, and 2) active fluid amplifiers.

Orifices, nozzles and capillary tubes constitute the most commonly used forms of resistances in fluid control systems [24]. Fluid resistance can be represented either by a single port or by a two port model. The single port representation can be used only when the resistance is a function of pressure drop across the resistor and the flow through it. The model for a nonlinear resistor may, however, be a function of the individual pressures at the ends of the resistor rather than the pressure drop. Fluid resistors are normally connected between two components in a circuit. Hence there are two points in a resistor where power transfer occurs. Therefore, from a mere connection point of view, and also from the definition of a port, a series resistor would require a two port representation.

Figure 1 shows a two port representation of a resistor and its typical characteristics. Analytical models for some known forms of fluid resistors are given in Table II. If experimentally measured data which do not conform with any of the models in Table II are encountered, then the graphical characteristic may be represented analytically by using polynomials. It has been found that the characteristic of nonlinear resistors generally follow a parabolic form. Equations which

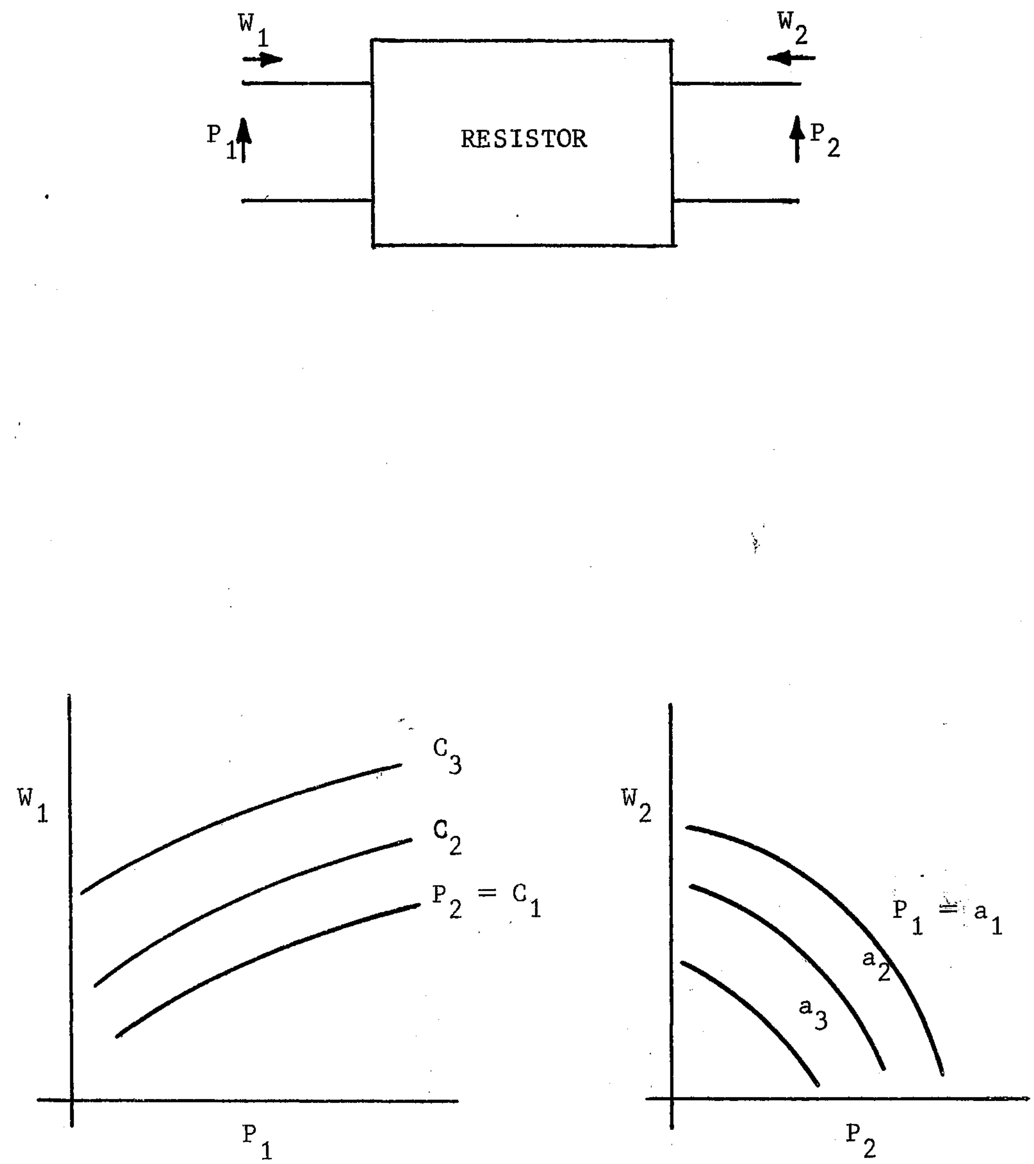


Figure 1. Two-port Model of a Resistor and its Graphical Characteristics

express the behavior of a general resistor using the polynomial representation may be written in the form:

$$W_1 + W_2 = 0 \tag{1}$$

$$W_1 - \text{Sign}\,(P_1 - P_2) \sum_{i=1}^{NX+1} C_i \{|P_1 - P_2|\}^{(i-1)} = 0 \tag{2}$$

where NX is the lowest order of the polynomial that fits the measured characteristics in a least square sense, and C_i are the constants of the polynomial.

TABLE II

MATHEMATICAL MODELS FOR RESISTORS

Type	Section	Model*
1. Capillary Tube (Laminar)	Circular	$W_1 + W_2 = 0$ $W_1 = \frac{\pi d^4}{128\mu L} (P_1 - P_2)$
2. Capillary Tube	Rectangular	$W_1 + W_2 = 0$ $W_1 = \frac{\omega b^3}{12\mu L} (P_1 - P_2)$
3. Orifices and Nozzles	Circular and Rectangular	$W_1 + W_2 = 0$ $W_1 = \text{Sign}(P_1 - P_2)\, K\, (\lvert P_1 - P_2\rvert)^{1/2}$

*See Figure 1 for sign convention.

Except for open circuit and closed circuit models, generalized static input-output models do not exist for active proportional fluidic components. Reliance must be placed on experimentally determined characteristics. Three sets of characteristics are required:

1. Power nozzle (supply jet) characteristics
2. Normalized input characteristics
3. Normalized output characteristics

Typical measured static characteristics of a proportional amplifier are shown in Figure 2. The power nozzle characteristics describe the resistive behavior of the supply port. These characteristics are non-linear in nature. Analytically, the power nozzle characteristics may be expressed by the equation,

$$W_1 - \sum_{i=1}^{NP+1} C_{1_i} P_1^{(i-1)} = 0 \tag{3}$$

where NP is the order of the lowest order polynomial required to fit the power nozzle characteristics in a least square sense, and C_{1_i}, $i = 1, \ldots, NP+1$ are constants.

The input characteristics represent the impedance seen by the input (control) signal. For most commercially available proportional amplifiers, these input characteristics are practically independent of the output conditions. The characteristics are, however, dependent on the bias (the average level of the inputs). It is assumed in this analysis that the normalization of the input characteristics with respect to the supply conditions is valid. The characteristics may be expressed mathematically in the form

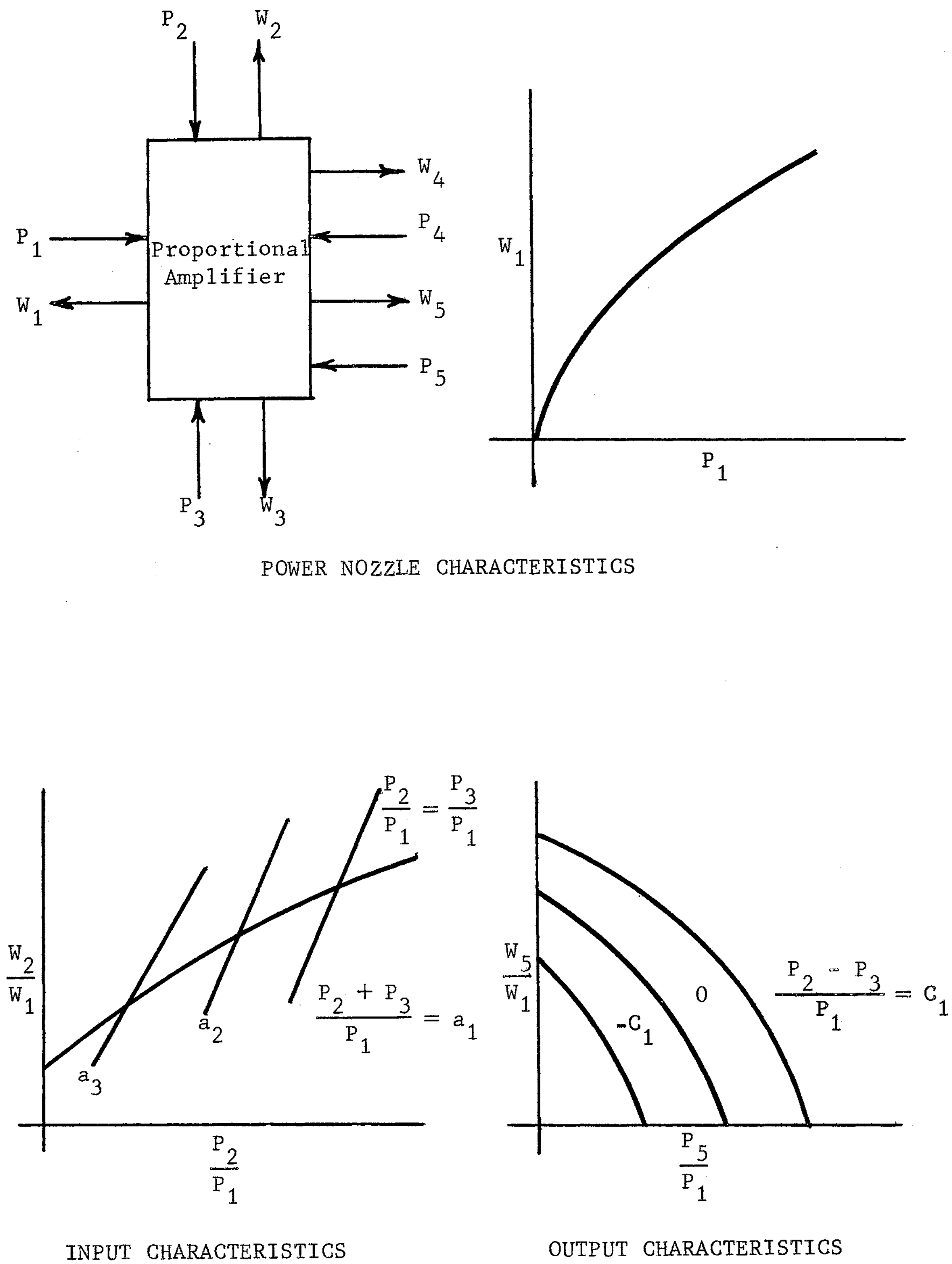


Figure 2. Static Characteristics of a Proportional Amplifier

$$\frac{W_2}{W_1} - \sum_{j=1}^{NY+1} \sum_{i=1}^{NX+1} C_{2_{i,j}} \left(\frac{P_2}{P_1}\right)^{(i-1)} \left(\frac{P_2 + P_3}{P_1}\right)^{(j-1)} = 0 \qquad (4)$$

where NX and NY are the lowest order polynomials which fit the data in a least square sense and $C_{2_{i,j}}$ are constants. For a symmetrical amplifier, the input characteristics of each control are identical.

The output characteristics describe the variation of output flow with the output pressure as the load is varied from near zero impedance to near infinite impedance. Each control input difference, which may be flow or pressure difference, gives rise to a specific output characteristic. It is assumed that the output characteristics are independent of the input bias level and that the characteristics can be normalized with respect to the supply conditions. It is also assumed that the output characteristics of one output port are independent of the loading of the other output port. These assumptions are true for most of the proportional amplifiers presently available. Analytically, the output characteristics may be expressed by the relation,

$$\frac{W_5}{W_1} - \sum_{j=1}^{NY+1} \sum_{i=1}^{NX+1} C_{5_{i,j}} \left(\frac{P_5}{P_1}\right)^{(i-1)} \left(\frac{P_2 - P_3}{P_1}\right)^{(j-1)} = 0 \qquad (5)$$

where NX and NY are the minimum orders of the polynomials which best fit the data in a least square sense.

Other proportional fluidic amplifiers including rectifiers can also be modeled using a similar approach.

Formulation of Operating Point Analysis Problems

The operating point analysis involves the determination of the values of the various port variables of a circuit for a given set of inputs. The problem consists of two major steps. The first step is to formulate the system equations using laws of interconnection (which depend upon the circuit configuration), and the component models; and the second step is to solve the resulting algebraic equations.

A simplified flow diagram of the algorithm for the operating point analysis is shown in Figure 3. The configuration of the circuit to be analyzed, the tabulated data for the static characteristics of the component, parameters for analytic models, and other input information are furnished to the Fluidic Circuit Analysis Program, FCAP (See Appendix A). The static analysis section of FCAP "picks up" the subroutines for each component in the circuit from the component library, and connects the proper ports to form the circuit to be analyzed. Next, FCAP fits the tabulated data, if any, with polynomials of the lowest order that best describe the data in a least square sense. The algebraic equations from all the component subroutines are compiled into a single set of simultaneous equations of the form:

$$\underline{G}\ (\underline{U},\ \underline{V}) = 0 \tag{6}$$

where,

G is the functional form of the algebraic equations,

U is the independent port variable,

V is the dependent port variable.

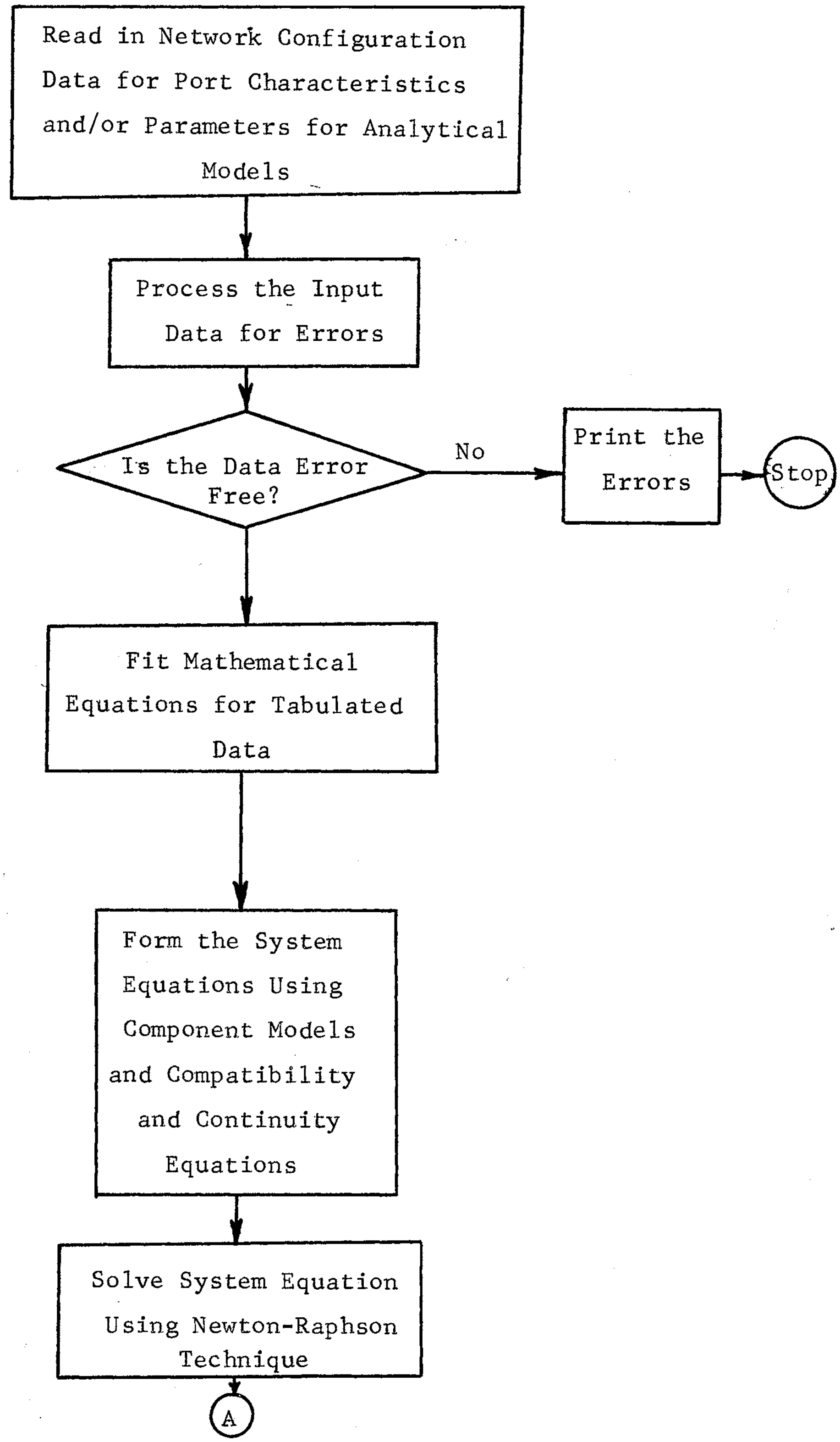


Figure 3. Flow Diagram of the Static Analysis Section of FCAP

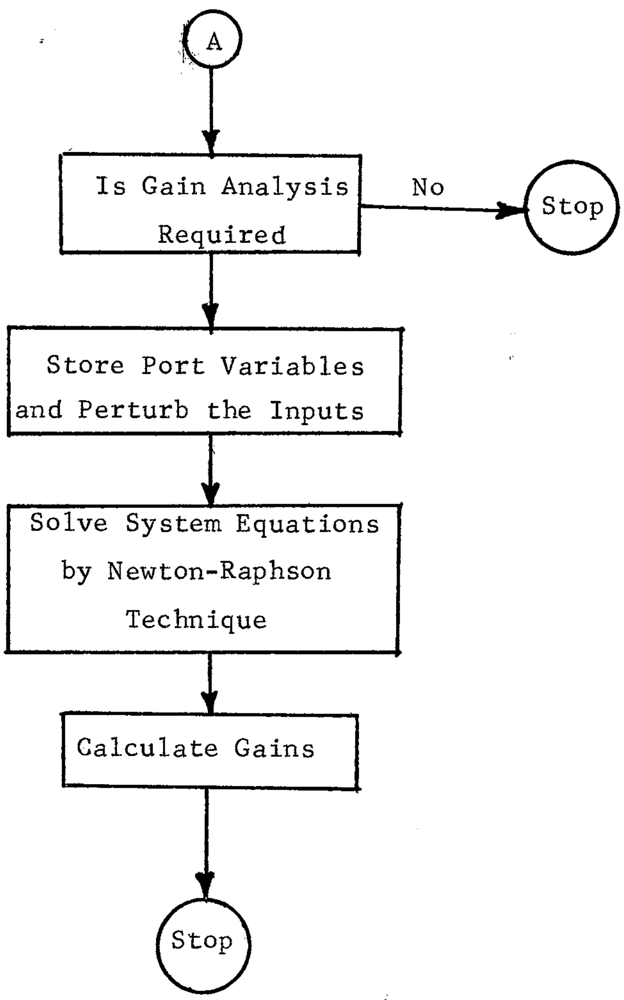


Figure 3 (Continued)

Depending upon the network configuration, FCAP generates the necessary continuity and compatibility equations. The continuity and compatibility equations express the relations between the independent variable at a port of a component and the dependent variable at the port of another component connected to it. These equations have the general form:

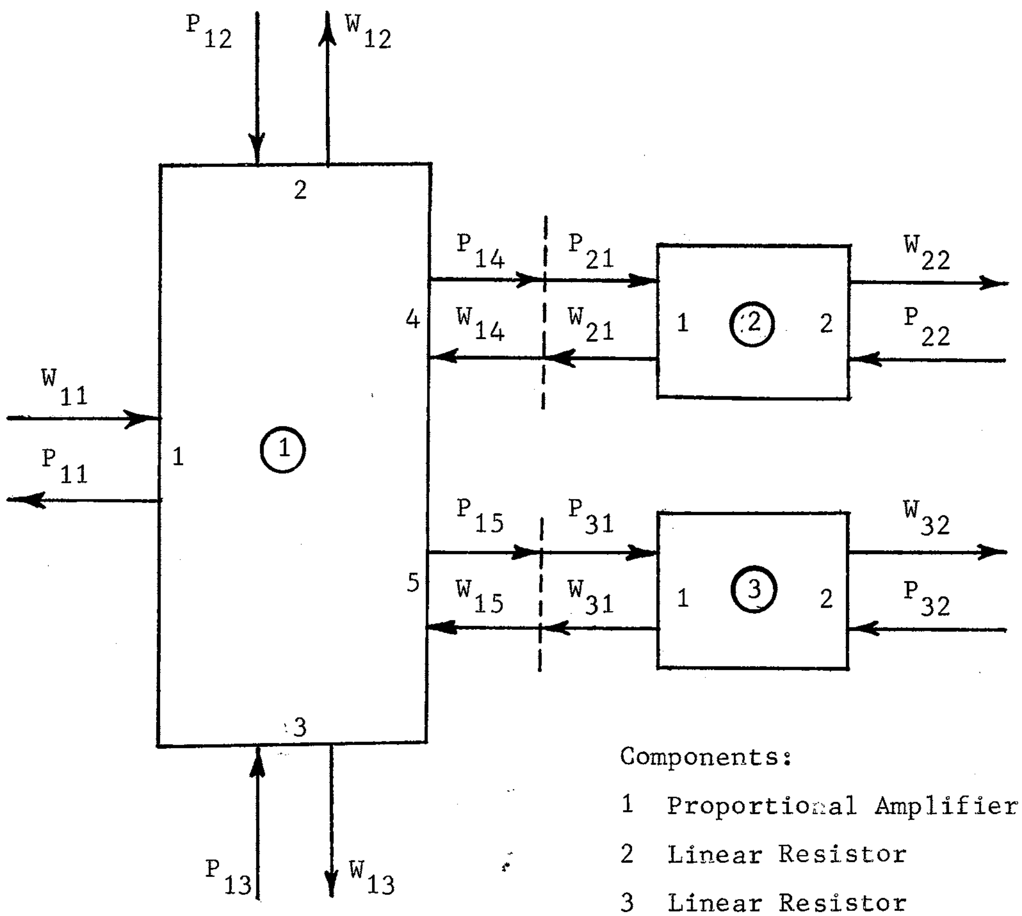
$$\underline{L}\ (\underline{U}, \underline{V}) = 0 \tag{7}$$

where,

L is the functional form of the compatibility and continuity equations. FCAP uses the continuity and compatibility equations to eliminate the independent port variables in the set of simultaneous equations of algebraic models, in favor of dependent port variables. The resulting set of equations includes as variables only dependent port variables and inputs to the network. The standard Newton-Raphson technique is used to solve this algebraic set of equations.

Example Formulation

To illustrate the formulation of the system equations explained above, consider the fluidic circuit shown in Figure 4. The first subscript of the port variables in Figure 4 refers to the component number and the second subscript refers to the port number. The circuit configuration, the tabulated data for the port characteristics of the proportional amplifier, the values of the linear resistors, and the independent variables at the ports which are free are furnished to the FCAP program. FCAP first picks up the proportional amplifier and the resistor subroutines from the component library and builds the circuit



Components:
1 Proportional Amplifier
2 Linear Resistor
3 Linear Resistor

Figure 4. A Fluidic Circuit With a Proportional Amplifier Loaded by Linear Resistors

shown in Figure 4. It should be noted here that a subroutine for a particular type of component can represent more than one component of the same type. The curve and surface fitting programs fit polynomials of the form given in Equations (3), (4), and (5) to the static characteristics of the proportional amplifier. The models for the resistors are analytic in this example and hence require only the value of the resistance.

Three sets of algebraic equations exist, one in each component describing the component's static characteristics. FCAP picks up these three sets of equations and compiles them into a single set of simultaneous equations. The set of simultaneous equations for the present example problem is:

$$G_{11} = W_{11} + \sum_i C_{11_i} \; P_{11}^{(i-1)} = 0, \tag{8}$$

$$G_{12} = \frac{W_{12}}{W_{11}} + \sum_i \sum_j C_{12_{i,j}} \left(\frac{P_{12}}{P_{11}}\right)^{(i-1)} \left(\frac{P_{12} + P_{13}}{P_{11}}\right)^{(j-1)} = 0, \tag{9}$$

$$G_{13} = \frac{W_{13}}{W_{11}} + \sum_i \sum_j C_{13_{i,j}} \left(\frac{P_{13}}{P_{11}}\right)^{(i-1)} \left(\frac{P_{12} + P_{13}}{P_{11}}\right)^{(j-1)} = 0, \tag{10}$$

$$G_{14} = \frac{W_{14}}{W_{11}} - \sum_i \sum_j C_{14_{i,j}} \left(\frac{P_{14}}{P_{11}}\right)^{(i-1)} \left(\frac{P_{12} - P_{13}}{P_{11}}\right)^{(j-1)} = 0, \tag{11}$$

$$G_{15} = \frac{W_{15}}{W_{11}} - \sum_i \sum_j C_{15_{i,j}} \left(\frac{P_{15}}{P_{11}}\right)^{(i-1)} \left(\frac{P_{12} - P_{13}}{P_{11}}\right)^{(j-1)} = 0, \tag{12}$$

$$G_{21} = W_{21} + W_{22} = 0, \tag{13}$$

$$G_{22} = P_{21} - P_{22} - R_2W_{22} = 0, \tag{14}$$

$$G_{31} = W_{31} + W_{32} = 0, \tag{15}$$

$$G_{32} = P_{31} - P_{32} - R_3W_{32} = 0. \tag{16}$$

FCAP then generates the continuity and compatibility equations. For the circuit given in Figure 4, the continuity and compatibility equations are:

$$L_1 = W_{14} + W_{21} = 0, \tag{17}$$

$$L_2 = W_{15} + W_{31} = 0, \tag{18}$$

$$L_3 = P_{14} - P_{21} = 0, \tag{19}$$

$$L_4 = P_{15} - P_{31} = 0. \tag{20}$$

FCAP eliminates the dependent port variables W_{21}, W_{31}, P_{14} and P_{15} in Equations (8) through (16) implicitly by using Equations (17) through (20). The resulting equations can be solved for the nine unknown dependent variables since the input variables P_{11}, P_{12}, P_{13}, P_{22}, and P_{32} are known. FCAP uses the standard Newton-Raphson technique to solve this resulting algebraic equation set.

Formulation of Linearized Gain Analysis Problem

Gain analysis involves the determination of the ratio of the change in the value of a certain specified port variable(s) to the corresponding change in the input variable. The change in the input variable is maintained small so that linearized assumptions hold.

The simplified flow diagram for the linearized gain analysis is shown in Figure 3. The operating point of the circuit is determined for the given set of inputs using the method described in the previous section. The port variables required for gain evaluation are then stored. The inputs to the system are then slightly perturbed. (The amount of perturbation of the input is user controlled.) The system is again solved for the operating point with the new values of the inputs. The differences between the new value and the previously stored value of port variables are computed. The gain is then calculated from the equation:

$$G = \frac{\Delta V_o}{\Delta V_i} \tag{21}$$

where G is the gain,

ΔV_o is the change in value of the variable at the output port of interest,

ΔV_i is the change in the value of the variable at the control input port of interest.

The FCAP Program

The Fluidic Circuit Analysis Program, FCAP, is a user-oriented program for predicting the static and dynamic analysis of fluidic circuits. The user needs to specify the circuit configuration and some other related data in order to initiate the FCAP program. No real programming knowledge is required on the part of the user. FCAP has some commonly used fluidic component models stored in its "component library" (See Appendix A). No attempt has been made to include all possible models available from literature. A procedure is outlined whereby the user can supply additional or refined models, depending on the particular problem requirements. FCAP presently includes models for the following types of components:

1. Proportional amplifier,
2. Rectifier,
3. Resistor,
4. Tee,
5. Fluid capacitor,
6. Transmission line,
7. Function generator.

Models for the first two components are generalized and require the user to supply experimental data or data available from manufacturers' catalogs when suitable. Analytic models are provided for the remaining components, although phenomenological models can be developed (internal to the program) if the user provides measured data. Provision is made for adding additional component types or additional models for existing components. Each type of component is described by a separate subroutine. The static and the dynamic models representing the behavior of a component are arranged in the multi-port format within the particular component subroutine. Besides the component subroutines, FCAP includes other computational and organizational subroutines. The computational subroutines include curve fitting programs for fitting equations to tabulated data for the static characteristics of components, a program which utilizes the Newton-Raphson technique for solving the nonlinear algebraic equations in the static and dynamic models, and a Runge-Kutta numerical integration program for integrating the dynamic equations.

The organizational subroutines check the input data for errors and set up the temporary memory of control vectors required for the particular problem. The computational subroutines along with the organizational and component subroutines can simulate any proportional fluidic circuit composed of any number of components (limited at present to 20).

The basic steps a user is required to follow in order to initiate the FCAP program for either static or dynamic analysis are:

1. Write the block diagram of the circuit following certain FCAP rules,
2. Assign unique numbers between 1 and 20 to each component,
3. Fill a first table which designates interconnections of the various components, indicates the variable to be printed, and the type of analysis to be performed.
4. Fill a second table which specifies the tabulated data for the components which are represented by measured terminal characteristics,
5. Fill a third table which indicates the ports of different components which have identical characteristics,
6. Fill a fourth table which specifies values of the independent variables at those ports which are not connected to any other components (free ports),
7. If gain analysis is required, fill a fifth table which furnishes the data for the gain analysis,
8. Fill a sixth table which specifies the parameter values for the components which are represented by analytic models.

The six tables referred to above are transferred onto computer cards to form the input data for the solution of a particular problem using the FCAP program. The next section discusses an example problem.

Example Problem 1

The circuit to be analyzed is shown in Figure 4. It is required to determine the operating point of the circuit and the linearized pressure gains associated with ports 4 and 5 of the proportional amplifier when the differential input to the proportional amplifier is varied by 0.005 psig. Other input data are given in Tables III through VII.

The step by step procedure delineated in the section on FCAP program (also see Appendix A) is followed in setting up the input data for this problem.

Step 1. The block diagram of the circuit is shown in Figure 5.

Step 2. The proportional amplifier is assigned 1 and the resistors are assigned 2 and 3 respectively.

Step 3. Table III shows the type of analysis to be performed and the data for circuit interconnections.

Step 4. Figures 6, 7, and 8 show the manufacturer's static characteristics for the proportional amplifier. These characteristics are expressed as tabulated data as shown in Table IV.

Step 5. Not applicable

Step 6. Table V shows the data for free ports.

Step 7. The data for the gain analysis is presented in Table VI.

Step 8. The components 2 and 3 require the value of one parameter (resistance). This is entered in Table VII.

The input information in Tables III through VII was transferred to

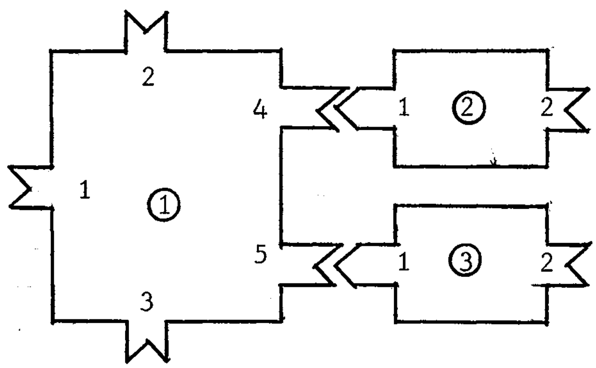


Figure 5. Block Diagram for
Example Problem 1

TABLE III

CIRCUIT DATA FOR EXAMPLE PROBLEM 1

1. Type of analysis: Gain   Type of Gain: Pressure

2. Circuit Connection

COMP	TYPE	PORT 1				PORT 2				PORT 3				PORT 4				PORT 5			
		A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
1	1	1			1	1			1	1			1		2	1	1		3	1	1
2	5		1	4	1	1															
3	5		1	5		1															

A: Is port free or connected? = 1 if free

B: Component number of the port it is connected to

C: Port number of the component it is connected to

D: Should port variable be printed? = 1 if yes

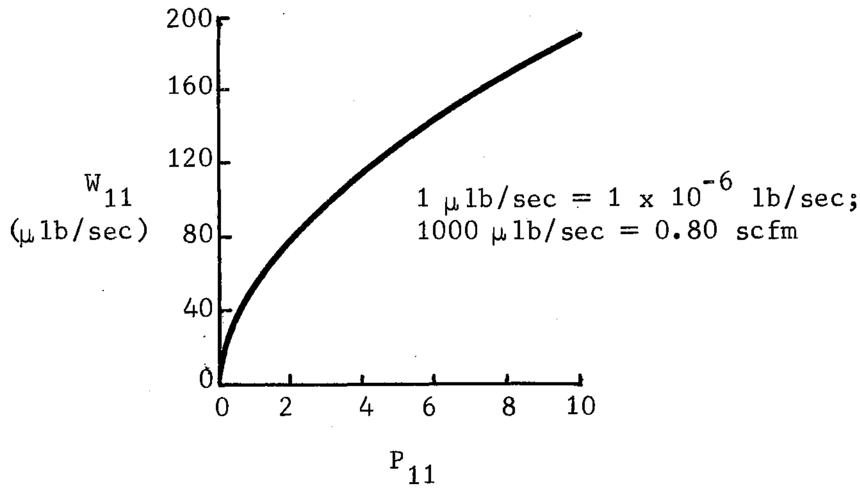


Figure 6. Supply Pressure--Flow Characteristics of GE AW 32

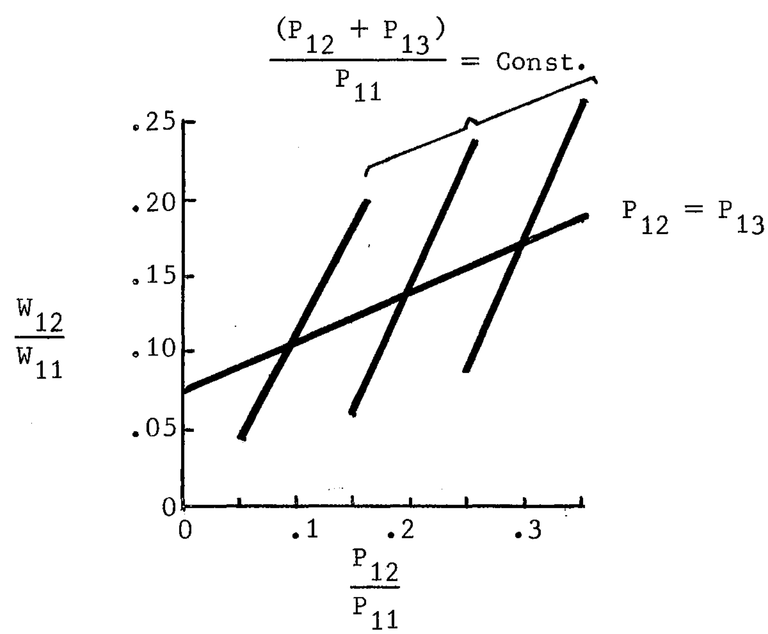


Figure 7. Input Characteristics of GE AW 32

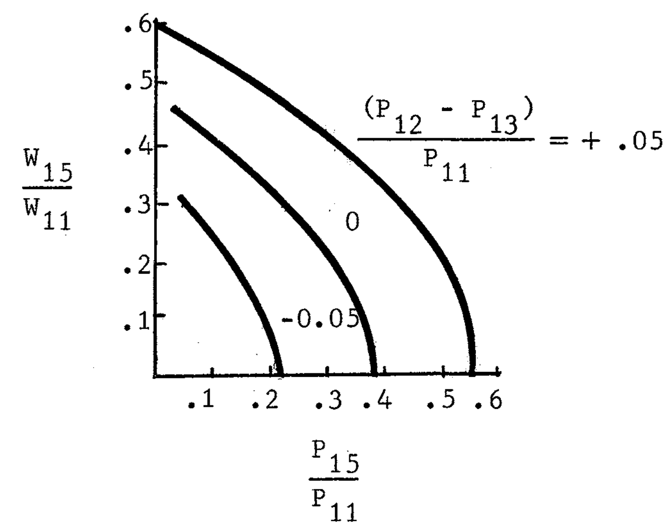


Figure 8. Output Characteristics of GE AW 32

TABLE IV

PORT DATA FOR EXAMPLE PROBLEM 1

Component	Port	Variable 1 (W_{11}) x 10^6 lbm/sec	Variable 2 (P_{11}) Psi	Variable 3
1	1	0.	0.0	
		60.	1.0	
		80.	2.0	
		100.	3.0	
		120.	4.0	
		133.	5.0	
		146.	6.0	
		160.	7.0	
		173.	8.0	
		186.	9.0	
		200.	10.0	

Component	Port	Variable 1 $\frac{P_{12}}{P_{11}}$	Variable 2 $(\frac{P_{12} + P_{13}}{P_{11}})$	Variable 3 $\frac{W_{12}}{W_{13}}$
1	3	0.0	0.03	0.2
		0.02	0.04	0.2
		0.045	0.06	0.2
		0.078	0.08	0.2
		0.11	0.10	0.2
		0.136	0.12	0.2
		0.165	0.14	0.2
		0.195	0.16	0.2
		0.225	0.18	0.2
		0.25	0.20	0.2
		0.0	0.115	0.4
		0.04	0.14	0.4
		0.075	0.16	0.4
		0.11	0.18	0.4
		0.14	0.2	0.4
		0.175	0.22	0.4
		0.21	0.24	0.4
		0.24	0.26	0.4
		0.27	0.28	0.4
		0.31	0.30	0.4
		0.0	0.19	0.6
		0.5	0.22	0.6
		0.08	0.24	0.6
		0.11	0.26	0.6
		0.15	0.28	0.6
		0.18	0.30	0.6

TABLE IV (Continued)

Component	Port	Variable 1 $\frac{P_{12}}{P_{11}}$	Variable 2 $(\frac{P_{12} + P_{13}}{P_{11}})$	Variable 3 $\frac{W_{12}}{W_{13}}$
		0.21	0.32	0.6
		0.24	0.34	0.6
		0.0	0.25	0.76
		0.042	0.28	0.76
		0.078	0.30	0.76
		0.11	0.32	0.76
		0.14	0.34	0.76
		0.17	0.36	0.76
		0.2	0.38	0.76
		0.23	0.40	0.76
		0.265	0.42	0.76
		0.30	0.44	0.76
		0.0	0.3	0.9
		0.025	0.32	0.9
		0.055	0.34	0.9
		0.09	0.36	0.9
		0.12	0.38	0.9
		0.15	0.40	0.9
		0.18	0.42	0.9
		0.215	0.44	0.9
		0.25	0.46	0.9
		0.28	0.48	0.9
		0.31	0.50	0.9

Component	Port	Variable 1 $\frac{W_{15}}{W_{11}}$	Variable 2 $(\frac{P_{12} - P_{13}}{P_{11}})$	Variable 3 $\frac{P_{15}}{P_{11}}$
1	5	0.62	0.0	0.05
		0.58	0.05	0.05
		0.55	0.10	0.05
		0.52	0.15	0.05
		0.48	0.20	0.05
		0.45	0.25	0.05
		0.42	0.30	0.05
		0.375	0.35	0.05
		0.325	0.40	0.05
		0.275	0.45	0.05
		0.2	0.50	0.05
		0.05	0.55	0.05
		0.48	0.0	0.0
		0.45	0.05	0.0
		0.41	0.10	0.0
		0.375	0.15	0.0

TABLE IV (Continued)

Component	Port	Variable 1 $\frac{W_{15}}{W_{11}}$	Variable 2 $(\frac{P_{12} - P_{13}}{P_{11}})$	Varialbe 3 $\frac{P_{15}}{P_{11}}$
		0.325	0.20	0.0
		0.275	0.25	0.0
		0.225	0.3	0.0
		0.15	0.35	0.0
		0.0	0.38	0.0
		0.36	0.0	-0.05
		0.30	0.05	-0.05
		0.25	0.1	-0.05
		0.175	0.15	-0.05
		0.06	0.2	-0.05
		0.0	0.22	-0.05

TABLE V

FREE PORTS DATA FOR EXAMPLE 1

Component	Port	Value
1	1	1.5
1	2	0.05
1	3	0.05
2	2	0.0
3	2	0.0

TABLE VI

GAIN ANALYSIS DATA FOR EXAMPLE PROBLEM 1

Input		Gain From		Gain To		
Component	Port	Component	Port	Component	Port	Tolerance
1	Differential	1	Differential	1	4	0.005
				1	5	

TABLE VII

PARAMETERS FOR EXAMPLE PROBLEM 1

Component	Parameter Values
2	0.1285 (Resistance $\frac{lbf}{in^2}$ / $\frac{lbm}{sec}$)
3	0.0133 (Resistance $\frac{lbf}{in^2}$ / $\frac{lbm}{sec}$)

computer cards and furnished to the FCAP program.

FCAP fit the following polynomials for the static characteristics of the proportional amplifier:

1. Supply pressure-flow characteristics

$$W_{11} = \sum_{i=1}^{5} C_{11_i} P_{11}^{(i-1)} \tag{22}$$

where,

$\underline{C}_{11} = [0.303\ E01 \quad 0.600\ E02 \quad -0.256\ E01 \quad 0.140\ E01 \quad -0.559\ E{-01}]$.

2. Input characteristics

$$\frac{W_{12}}{W_{11}} = \sum_{j=1}^{2} \sum_{i=1}^{2} C_{12_{ij}} \left(\frac{P_{12}}{P_{11}}\right)^{(i-1)} \left(\frac{P_{12}+P_{13}}{P_{11}}\right)^{(j-1)} \tag{23}$$

where,

$$\underline{C}_{12} = \begin{bmatrix} 0.730\ E{-01} & -0.616\ E00 \\ 0.156\ E01 & 0.259\ E{-01} \end{bmatrix}$$

3. Output characteristics

$$\frac{W_{15}}{W_{11}} = \sum_{j=1}^{3} \sum_{i=1}^{3} C_{15_{ij}} \left(\frac{P_{15}}{P_{11}}\right)^{(i-1)} \left(\frac{P_{12}+P_{13}}{P_{11}}\right)^{(j-1)} \tag{24}$$

where,

$$\underline{C}_{15} = \begin{bmatrix} 0.382\ E00 & 0.340\ E01 & 0.252\ E01 \\ -0.152\ E{-01} & 0.190\ E01 & -0.375\ E02 \\ -0.163\ E01 & -0.336\ E02 & 0.130\ E03 \end{bmatrix}$$

Since data were not furnished for ports 3 and 4 of the proportional amplifier, FCAP assumed the amplifier to be symmetric and transferred the proper coefficients to the equations for ports 3 and 4. The Newton-Raphson method converged in 5 iterations. The values of the various port variables were:

$W_{11} = -69.491$ lbm/sec, $W_{12} = -5.834\ \mu$lbm/sec, $W_{13} = -5.834\ \mu$lbm/sec

$P_{15} = 0.303$ psi, $W_{15} = 22.804\ \mu$lbm/sec, $P_{14} = 0.563$ psi

$W_{14} = 4.378$ lbm/sec, $W_{21} = 4.378\ \mu$lbm/sec, $W_{32} = 22.804\ \mu$lbm/sec

The gain values for ports 4 and 5 of the proportional amplifier when the differential input was perturbed by 0.005 psi were:

$G_{14} = -3.380$ $\qquad$ $G_{15} = 1.327$

where G_{14} and G_{15} are the gain the individual ports 4 and 5 of component 1. The differential for proportional amplifier 1 is 4.707.

A graphical analysis yielded

$P_{15} = 0.3$ psig $\qquad$ $W_{15} = 22.9\ \mu$lbm/sec

$P_{14} = 0.57$ psig $\qquad$ $W_{14} = 4.2\ \mu$lbm/sec

$G_{14} = -3.35$ $\qquad$ $G_{15} = 1.32$

The above example illustrates the use of FCAP for the static analysis of fluidic circuits. More complex problems may be solved just as easily as the one presented above. Static analysis of systems other than fluidic can also be performed using FCAP. This would require the addition of component subroutines describing the static behavior of the new components. Detailed information on the use of FCAP is presented in Appendix A.

CHAPTER IV

DYNAMIC ANALYSIS

The problem considered in this chapter is that of dynamic simulation of nonlinear fluidic circuits. The components that form the circuits may be either active or passive, and their models may be linear or nonlinear.

Component Models

Analytic models, derived using basic principles of fluid mechanics, have been reported for some fluidic components, e.g., the enclosed volumes (or capacitors), capillary tubes, orifices, and transmission lines. Static and dynamic models are available for these components which adequately describe their isolated terminal behavior over the range of frequencies normally encountered in fluidic control systems. There are, however, no known mathematical models that describe the nonlinear dynamic behavior of active proportional fluid amplifiers. The next section describes a method for deriving a nonlinear lumped parameter model for proportional fluid amplifiers.

Dynamic Modeling of Proportional Fluid Amplifiers

The silhouette of a typical proportional amplifier is shown in Figure 9. The amplifier is a seven port device. It has a power nozzle

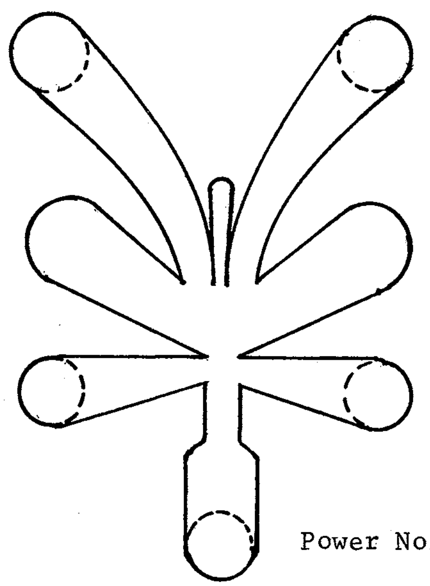


Figure 9. Silhouette of a Proportional Amplifier

port, two control ports, two vents and two receivers. The direction of flow of the high-energy power jet is controlled by two low energy control jets. The power jet is directed into the interaction region. Control jets are also directed into this region from either side of the power jet. The direction that the power jet assumes after interaction depends on the momentum flux of the power jet and the forces exerted by the control jets on the power jet. At some distance downstream the power jet is collected by receivers. When the control pressures and the loading are equal, the receivers collect the same quantity of fluid and hence the output pressures will be equal. A small change in either of the control pressures will deflect the main jet and a differential output pressure will result. Since the output difference normally is greater than the input difference, the device is termed an amplifier.

The dynamics of a proportional amplifier can be considered to be made up of the dynamics of four individual processes: 1) control port, 2) transport from nozzle to receiver, 3) vent, and 4) receiver. Since most amplifiers have short and wide vents, vent dynamics can be considered insignificant compared with the dynamics of the control and receiver ports. Figure 10 shows an equivalent circuit of the proportional amplifier.

Control Dynamics

The lumped parameter model for the control ports is shown in Figure 10. L_c is the inertance and C_c is the capacitance of the control passage. The block S_c represents the experimentally measured static characteristics of the input port. The dynamic behavior of control port 2 can be described by the equations:

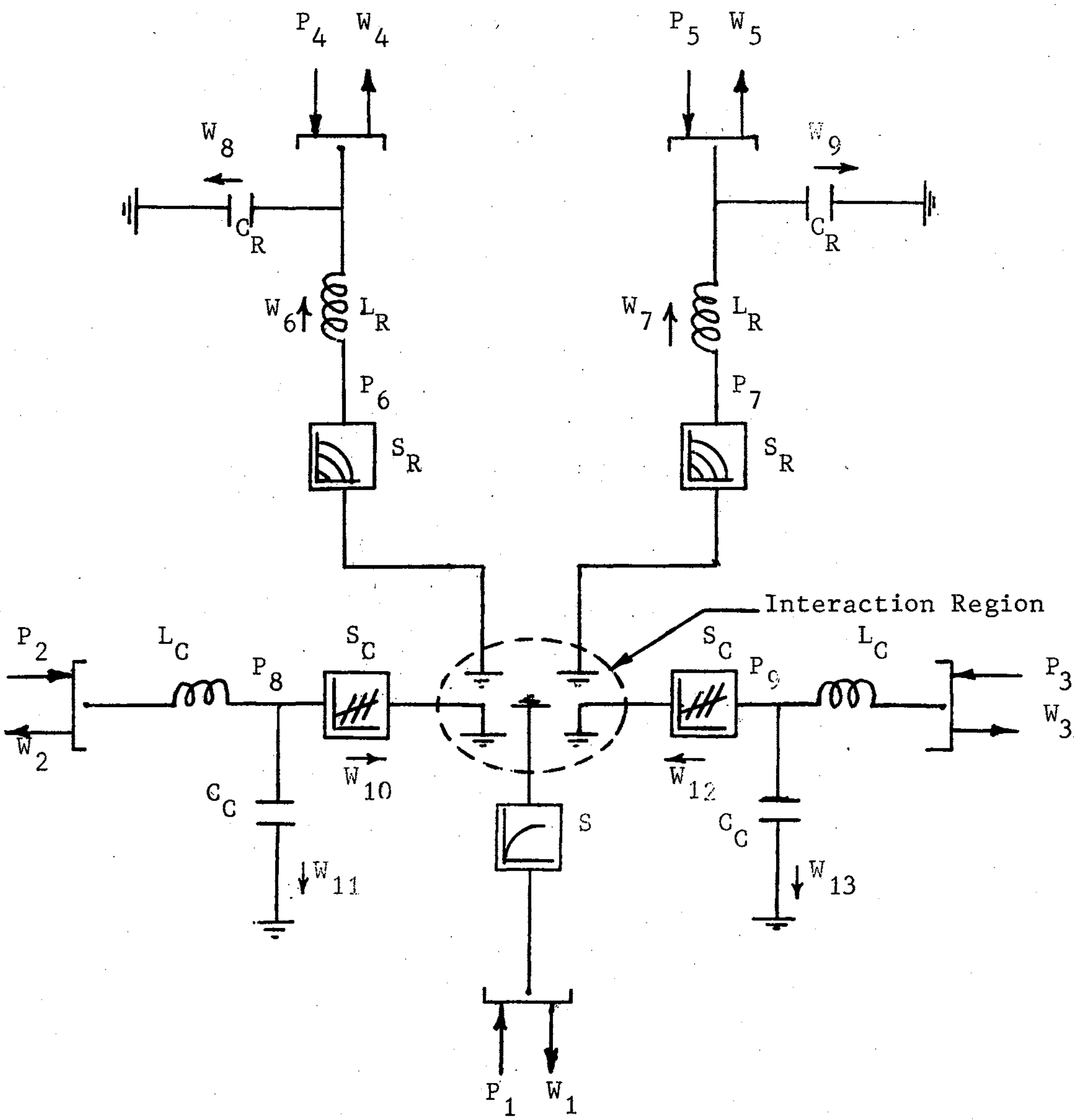


Figure 10. Equivalent Circuit for a Proportional Amplifier

$$\frac{dW_2}{dt} = \frac{1}{L_c} (P_2 - P_8) \tag{25}$$

$$\frac{dP_8}{dt} = \frac{1}{C_c} W_{11} \tag{26}$$

$$W_{10} - W_1 \sum_{j=1}^{NY+1} \sum_{i=1}^{NX+1} C_{ij} \left(\frac{P_8}{P_1}\right)^{(i-1)} \left(\frac{P_8 + P_9}{P_1}\right)^{(j-1)} = 0 \tag{27}$$

$$W_2 - W_{11} - W_{10} = 0 \tag{28}$$

$$W_1 - \sum_{i=1}^{NP+1} C_i \, P_1^{(i-1)} = 0 \tag{29}$$

Similarly, equations for port 3 can be written as:

$$\frac{dW_3}{dt} = \frac{1}{L_c} (P_3 - P_9) \tag{30}$$

$$\frac{dP_9}{dt} = \frac{1}{C_c} W_{13} \tag{31}$$

$$W_{12} - W_1 \sum_{j=1}^{NY+1} \sum_{i=1}^{NX+1} C_{ij} \left(\frac{P_9}{P_1}\right)^{(i-1)} \left(\frac{P_8 + P_9}{P_1}\right)^{(j-1)} = 0 \tag{32}$$

$$W_3 - W_{13} - W_{12} = 0 \tag{33}$$

$$W_1 - \sum_{i=1}^{NP+1} C_i \, P_1^{(i-1)} \tag{34}$$

It should be noted that the above equations for the control ports take into account the bias level of the inputs.

Transport From Nozzle to Receiver

Pure time delays result from wave propagation and fluid transport effects. It is assumed that the delay due to pressure wave propagation is negligible compared to the delay due to fluid transport [21]. It is also assumed that the dynamics of the jet due to its inertia and capacitance are small. Hence, the jet model becomes:

$$(P_8 - P_9)_R = e^{-TD}(P_8 - P_9) \tag{35}$$

where T is the time delay due to fluid transport from the nozzle to the receiver, and the subscript R refers to the receiver. That is, the effect of change in the differential pressure (P_8 - P_9) is felt at the receivers after a delay time T. It should be pointed out here that experimental results have shown that the phase lag caused by this delay is very samll compared with those caused due to the dynamics of the control and receiver passages, and hence can be neglected for low frequency models [21]. The proportional amplifier model used in the computer program, FCAP, neglects all dynamic effects associated with the power jet.

Receiver Dynamics

The lumped parameter model for the receivers is shown in Figure 10. L_R is the inertance and C_R is the capacitance of the output port. The block S_R represents the experimentally measured static characteristics of the output port. The receiver dynamics for port 4 can be represented by the following equations:

$$P_6 - P_1 \sum_{j=1}^{NY+1} \sum_{i=1}^{NX+1} C_{ij} \left(\frac{W_6}{W_1}\right)^{(i-1)} \left(\frac{P_8 - P_9}{P_1}\right)^{(j-1)} = 0 \tag{36}$$

$$W_6 - W_4 - W_8 = 0 \tag{37}$$

$$\frac{dW_6}{dt} = \frac{1}{L_R} \left(P_6 - P_4 \right) \tag{38}$$

$$\frac{dP_4}{dt} = \frac{1}{C_R} W_8 \tag{39}$$

$$W_1 - \sum_{i=1}^{NP+1} C_i \, P_1^{(i-1)} = 0 \tag{40}$$

Similarly, the equations for output port 5 can be written as

$$P_7 - P_1 \sum_{j=1}^{NY+1} \sum_{i=1}^{NX+1} C_{ij} \left(\frac{W_7}{W_1}\right)^{(i-1)} \left(\frac{P_8 - P_9}{P_1}\right)^{(j-1)} = 0 \tag{41}$$

$$W_7 - W_5 - W_9 = 0 \tag{42}$$

$$\frac{dW_7}{dt} = \frac{1}{L_R} \left(P_7 - P_5 \right) \tag{43}$$

$$\frac{dP_5}{dt} = \frac{1}{C_R} W_9 \tag{44}$$

$$W_1 - \sum_{i=1}^{NP+1} c_i P_1^{(i-1)} = 0 \tag{45}$$

Equations (25) through (45) describe the nonlinear dynamic behavior of the proportional amplifier. Since some of the algebraic equations are nonlinear, it is not possible to eliminate the intermediate variables explicitly by hand and write the differential equations in terms of only the independent and dependent port variables. It will be shown in later sections that the computational algorithm first solves the nonlinear set of algebraic equations and then integrates the nonlinear set of differential equations.

Equations similar to (25) through (45) may also be derived for other active devices using the above approach.

Example

A nonlinear dynamic model was derived based on the geometrical data and measured static characteristics (see Figures 11 and 12) for a typical proportional amplifier [21]. Step responses computed with the nonlinear model are shown in Figure 13 along with results obtained for the same amplifier by Boothe. Boothe's model was linear and neglected the capacitive effects of both the control and the receiver ports. Hence, for comparison, these effects were also neglected in the nonlinear model. For differential signal inputs, the step responses predicted by the two models agree well. As the amplitude of the differential input is increased, the steady state value predicted by the nonlinear model increases, but the rise time seems to be less affected.

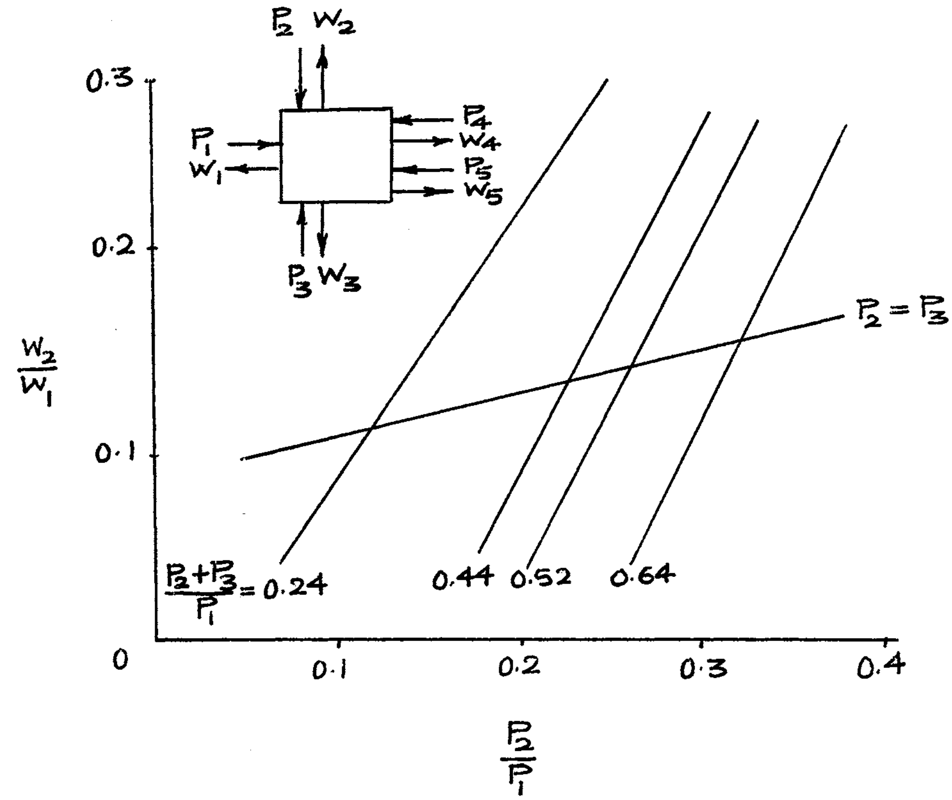


Figure 11. Normalized Input Characteristics for the Proportional Amplifier Presented by Boothe [21]

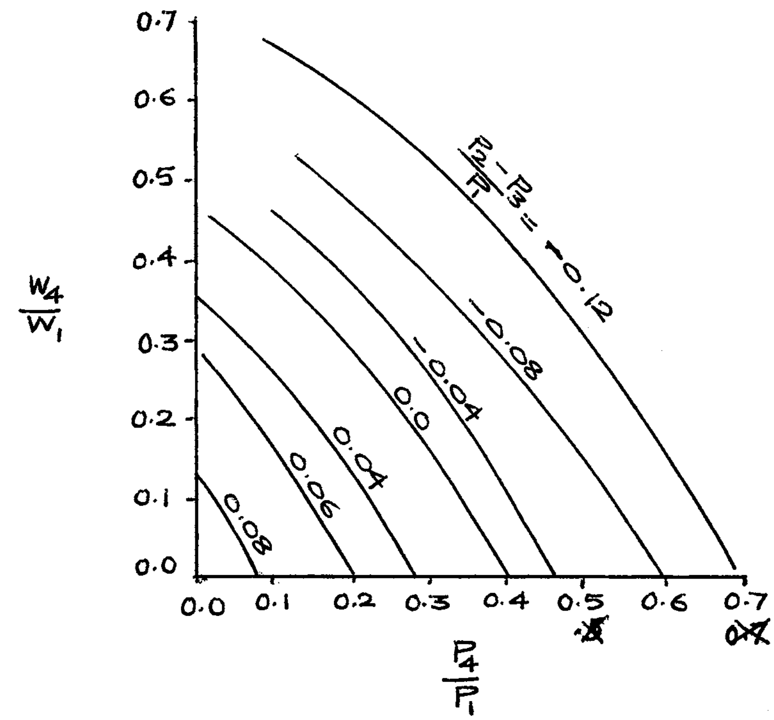


Figure 12. Normalized Output Characteristics for the Proportional Amplifier Presented by Boothe [21]

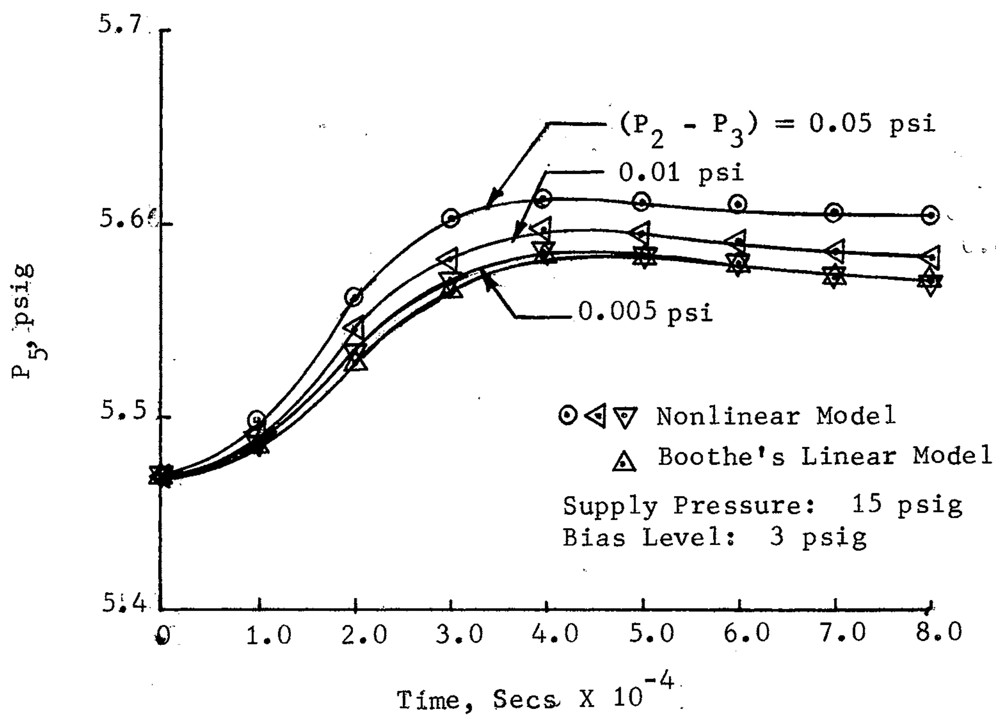


Figure 13. Comparison of Nonlinear and Linear Models for Proportional Fluid Amplifier [21]

Solution Process

As in the static analysis a modular approach is adopted for dynamic simulation. In the FCAP program, each discrete component is described by a model incorporated within a separate FORTRAN subprogram.

Component models fall into three classes: 1) algebraic equations only (if the component is purely resistive), 2) differential equations only, and 3) both algebraic and differential equations. In general, the dynamic model for a component can be represented by the equations:

$$\dot{\underline{X}} = \underline{F}\,(\underline{X}, \underline{U}, \underline{Y}, t) \tag{46}$$

$$\underline{Z} = \underline{Z}(\underline{X}, \underline{Y}, \underline{U}, t) \tag{47}$$

$$\underline{V} = \underline{V}\,(Y) \tag{48}$$

or

$$\underline{V} = \underline{V}(x, t) \tag{49}$$

where

X is the state variable,

U is the independent port variable,

Y is the algebraic variable,

V is the dependent port variable,

F is the functional form of the differential equations,

Z is the functional form of algebraic equations.

A component representation of this type has been presented by Smith [25]. In the general form of dynamic model described above, the dependent port variable V can be either equal to an algebraic variable or be a function of states and/or time.

The FCAP transient analysis program first solves for the steady state operation at the initial time and then initializes all the states in each dynamic component to the steady state values. The steady state operation is determined by using the static models of the components. The general form of static model for a component is

$$\underline{G} = \underline{G}\ (\underline{U}, \underline{V}) = 0 \tag{50}$$

where,

G is the functional form of equations describing the static models,

U is the independent variable at a port,

V is the dependent variable at a port.

The solution process for determining the steady state operation at the initial time is the same as the one described in Chapter III, Static Analysis.

The dynamic model for a component as described by Equations (46) through (50) may involve algebraic equations. The derivatives of states are in general, functions of states, independent port variables, algebraic variables, and time. It is therefore necessary to solve for the algebraic variables in the dynamic model before evaluating the derivatives. A set of algebraic equations in one component may be coupled with another set of algebraic equations in a component connected to it. Consider, for example, the system in Figure 14. Port 1 of component 'A' is connected to Port 2 of component 'B'. If V_{A1} is a function of an algebraic variable in component A and if the functional form of algebraic equations Z in component B involve U_{B2}, then the algebraic equations in component A will be coupled with those in component B. The same is true for V_{B2} and U_{A1}.

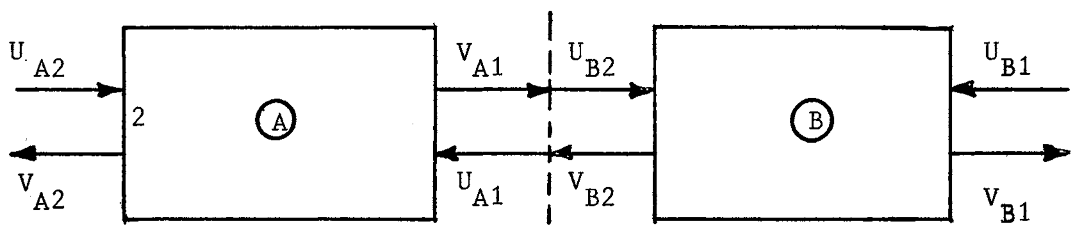


Figure 14. Port Connection Between Two Components

In a practical circuit, there could be many such coupled algebraic loops which are isolated from each other. The FCAP program isolates these algebraic loops and solves each loop separately. The method used in FCAP for algebraic loop isolation is similar to the one described by Smith [25].

Figure 15 is a simplified flow diagram which shows the computational procedure used in FCAP. The standard Newton-Raphson technique is used to solve the algebraic equations in each loop. The differential equations are integrated using the Runge-Kutta method.

Example Problem 2

Consider the simple fluidic circuit shown in Figure 16. The circuit is made up of a General Electric AW 32 proportional amplifier, two linear resistors, a capacitor, and two function generators. The data used for simulation were the following.

1. Components:

a) Proportional amplifier

Inertance of each control port = 0.2292 sec^2/in^2

Inertance of each output port = 0.2050 sec^2/in^2

The static characteristics are the same as shown in Figures 5, 6, and 7.

b) Capacitor 2:

Capacitance = 0.2 in^2

c) Resistor 3:

Resistance = 0.01 psi/lbm/sec

d) Resistor 4:

Resistance = 0.02 psi/lbm/sec

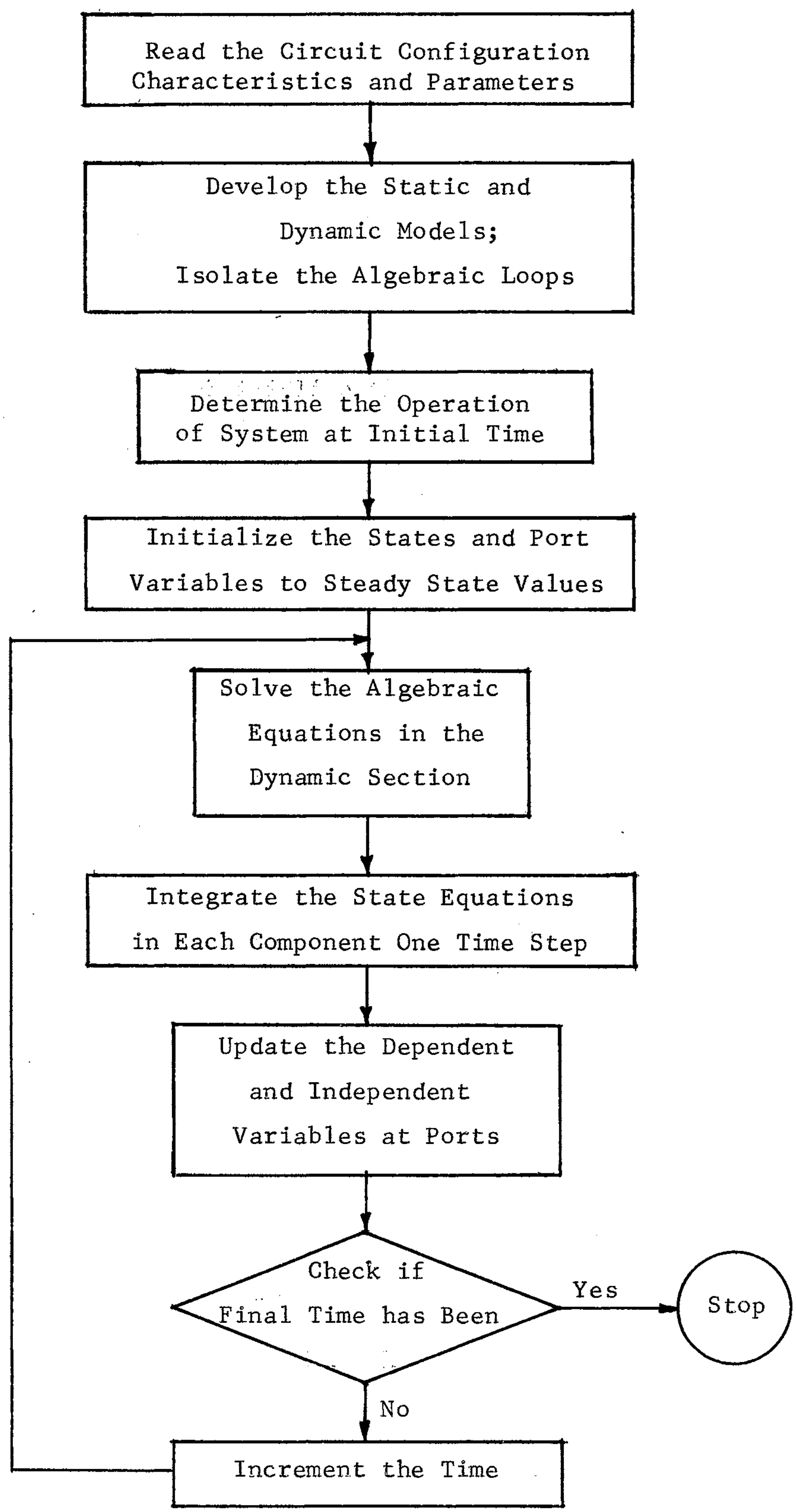
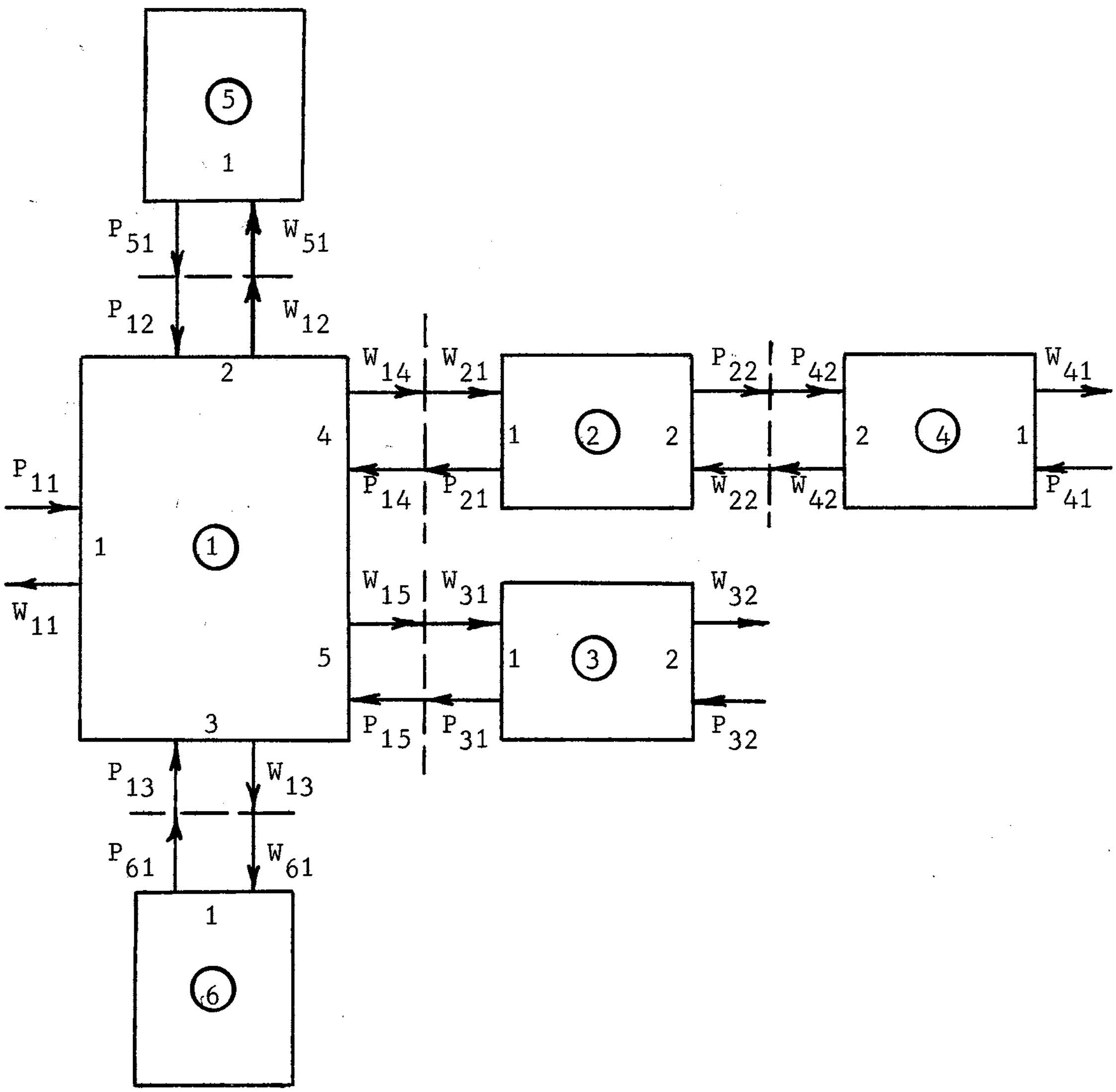


Figure 15. Computer Algorithm for Transient Analysis



Components: (1) GE AW 32 Proportional Fluid Amplifier

(2) Capacitor

(3) Resistor

(4) Resistor

(5) Forcing Function

(6) Forcing Function

Figure 16. Diagram of a Simple Fluidic Circuit

e) Forcing function 5

$$P_{51} = 0.15 \text{ psig for } t \leq 0$$
$$= 0.10 \text{ psig for } t > 0$$

f) Forcing function 6

$$P_{61} = 0.15 \text{ psig for } t \leq 0$$
$$= 0.20 \text{ psig for } t > 0$$

The independent variables at the "free ports" were assumed to be:

$$P_{11} = 1.5 \text{ psig}, \quad P_{32} = 0.0 \text{ psig, and } P_{41} = 0.0 \text{ psig.}$$

The system was simulated for a total elapse time of 0.3 sec. The time step used for the integration was 0.005 sec. The FCAP program solved the algebraic equations at the initial time (t = 0) and initialized the states and algebraic variables in the proportional amplifier and the capacitor to the following values:

a) Proportional amplifier:

$$W_{12} = Y_2 = 7.381 \ \mu \text{ lbm/sec}, \quad W_{14} = X_3 = 19.13 \ \mu \text{ lbm/sec}$$
$$P_{12} = X_1 = 0.15 \text{ Psi}, \quad P_{14} = U_4 = 0.382 \text{ Psi}$$
$$W_{13} = Y_3 = 7.381 \ \mu \text{ lbm/sec}, \quad W_{15} = X_3 = 24.985 \ \mu \text{ lbm/sec}$$
$$P_{13} = X_2 = 0.15 \text{ Psi}, \quad P_{15} = U_5 = 0.25 \text{ Psi.}$$

b) Capacitor:

$$P_{21} = X_1 = 0.382 \text{ Psi}, \quad P_{22} = Y_1 = 0.382 \text{ Psi.}$$

The pressure and flow at port 4 of the proportional amplifier are plotted as a function of time in Figures 17 and 18 respectively. The steady state values of the pressure and flow at the initial and final times agree within 5 percent of the results obtained by graphical analysis. This error is within the limits of the accuracy of the

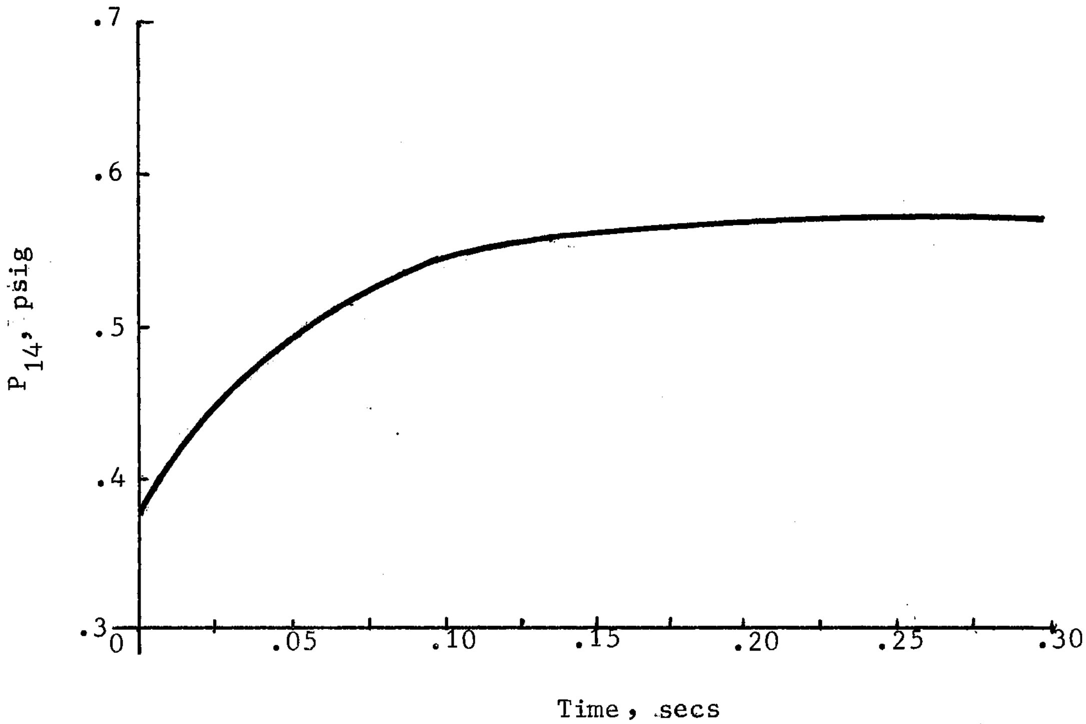


Figure 17. Pressure P_{14} Versus Time

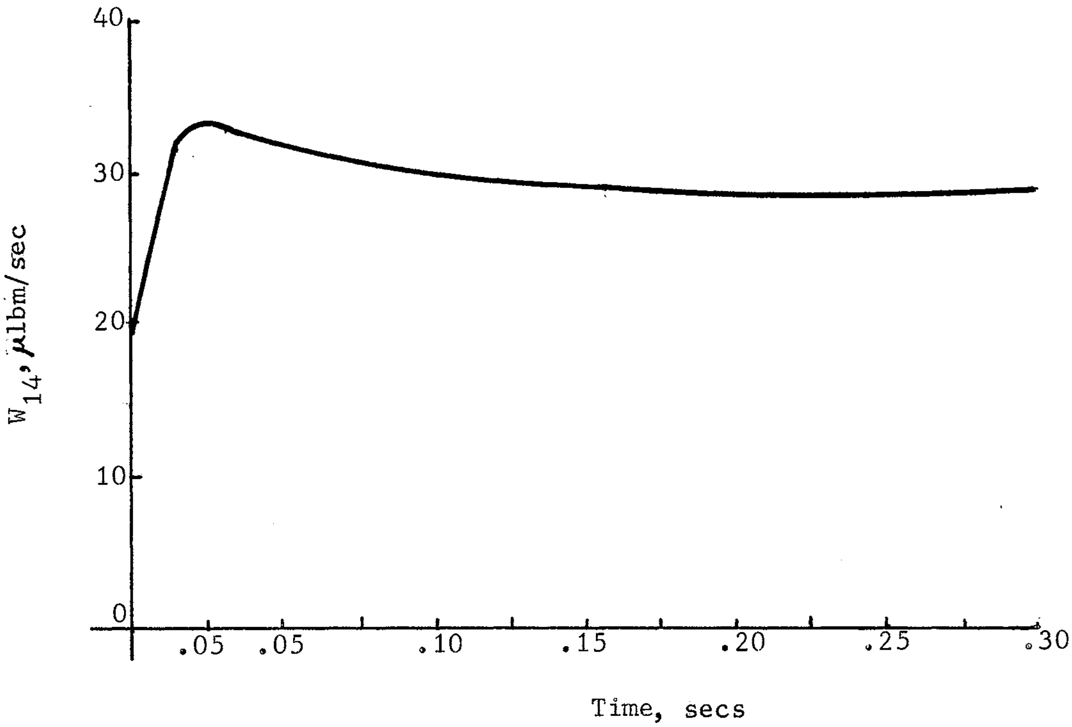


Figure 18. Flow W_{14} Versus Time

graphical technique. No attempt was made to validate the results obtained from the FCAP program with experimental measurements. The accuracy of the results predicted by the computer program, FCAP, depends upon the refinement of the models used for the components.

CHAPTER V

NETWORK OPTIMIZATION

Real fluidic components are nonlinear in behavior. As a result, a circuit consisting of real fluidic components may have a totally different type of behavior for each different set of parameter values. There may be many combinations of the parameters which result in essentially identical operation of the circuit. However, only one of these combinations is "best" according to some criterion. It is difficult and time consuming to determine all feasible combinations of the parameters by hand or even by using a computerized analysis program. An optimization procedure is needed by which the parameters of a circuit may be selected to extremize a certain criterion. This chapter discusses a NETwork OPTimization porgram, NETOPT, which can be used to determine an optimum set of parameters for a fluidic circuit in accordance with a specified static performance criterion. The circuit components may be linear or nonlinear, and active or passive. A NETOPT user's manual is presented in Appendix B.

Criterion to be Optimized

Typical performance criteria of interest in the design of fluidic circuits are: 1) minimize power consumption of the active components, 2) maximize pressure or flow gain, 3) generate a specific input-output function, 4) maintain linear operation, or 5) combination of these. The selection of a particular performance criterion depends upon the specific design problem. Active fluidic devices are open centered and therefore inherently wasteful of power. Moreover, as the circuit complexity increases, the concern for total power consumption increases as well. It is in the design of relatively complex circuits that the computer-aided static optimization procedure of this thesis becomes particularly important. Consequently, minimum power consumption seems the logical choice as a single most important performance criterion to incorporate into the procedure.

The NETwork OPTimization program, NETOPT, allows the determination of the optimum values of certain component parameters such that the power consumed by the active fluidic amplifiers is a minimum, while the static performance of the circuit meets certain parameter and output constraints. The program can be modified easily to include other performance criteria.

Parameter and Output Constraints

In order to obtain physically realizable results, some component parameters and certain circuit outputs must be constrainted. The constraints may be either regional or functional constraints which place restrictions on parameters and output functions. The regional constraints specify the upper and lower limits of the parameters and outputs, while the functional constraints express certain functional relationships that have to be satisfied by the parameters and outputs. The component parameters considered for optimization are: 1) supply pressures to active devices, 2) resistance values of passive linear resistors, and 3) areas of certain types of passive nonlinear resistors.

These particular parameters have been selected since they are normally the ones at the control of the designer. It is assumed that the geometry of each active device is fixed.

The system output constraints may be general functions of the port variables. One output constraint which has been programmed in NETOPT is the gain between any two specified points in the circuit. Other output constraints can be specified through a user supplied subprogram, CNSTNT, described in Appendix B.

Mathematical Statement of the Problem

The mathematical problem solved by the optimization program, NETOPT, may be stated as follows: Determine a parameter vector $\underline{K}$ which minimizes a performance criterion

$$f(\underline{U}, \underline{V}, \underline{K}) = \sum_{i=1}^{IACT} (U_{N(I),I})(V_{N(I),I}) \tag{51}$$

subject to the constraints,

$$a \leq g_K(\underline{U}, \underline{V}, \underline{K}) \leq b \quad , \quad k = 1, 2, \ldots, NG \tag{52}$$

$$g_i(\underline{U}, \underline{V}, \underline{K}) \geq 0 \quad , \quad i = NG+1, \ldots, M \tag{53}$$

$$h_j(\underline{U}, \underline{V}, \underline{K}) = 0 \quad , \quad j = M+1, \ldots, M+MZ \tag{54}$$

where U is the independent variable at a port,

V is the dependent variable at a port,

K is the parameter to be optimized,

g_K is the functional relations for the gains to be constrained, a is the lower limit of gain allowed,

b is the upper limit of gain allowed,

g_i is the functional form of the user-supplied inequality constraints,

h_j is the functional form of user-supplied equality constraints,

IACT is the total number of active components,

N is the vector containing component numbers of active components,

NG is the number of gain constraints,

M is the total number of inequality constraints,

MZ is the number of equality constraints.

Optimization Strategy

NETOPT uses the sequential unconstrained minimization technique presented by Mylander, et. al. [26] to solve the constrained optimization problem stated in Equations (51) through (54). The solution to the constrained optimization problem is obtained by solving a sequence of unconstrained problems whose solutions approach that of Equations (51) through (54). A penalty function of the form:

$$\eta(\underline{U}, \underline{V}, \underline{K}, r) = f(\underline{U}, \underline{V}, \underline{K}) - r\sum_{i=1}^{M} \ln(g_i(\underline{U}, \underline{V}, \underline{K})) + \frac{1}{r}\sum_{j=M+1}^{M+MZ} h_j^2(\underline{U}, \underline{V}\underline{K}) \quad (55)$$

is defined. r is a slack variable whose value is arbitrarily set (normally equal to 1) at the beginning of the problem. The solution to the constrained minimization problem stated in Equations (51) through (54) may be obtained by sequentially reducing the value of r and minimizing the unconstrained penalty function $\eta(\underline{U}, \underline{V}, \underline{K}, r)$. A decrease in the value of $\eta(\underline{U}, \underline{V}, \underline{K}, r)$ is due to a combination of three factors: 1) a decrease in the value of $f(\underline{U}, \underline{V}, \underline{K})$, 2) an increase in the value of $\eta[g_i(\underline{U}, \underline{V}, \underline{K})]$, that is, an increase in $g_i(\underline{U}, \underline{V}, \underline{K})$, and 3) a decrease in the value of $h_j(\underline{U}, \underline{V}, \underline{K})$. The minimization of $\eta(\underline{U}, \underline{V}, \underline{K}, r)$ for a given value of r is called a subproblem. Solving a sequence of subproblems by reducing the value of r drives the inequality constraints to greater than zero and the equality constraints to near zero. The necessary conditions for the existance of the minima of $\eta(\underline{U}, \underline{V}, \underline{K}, r)$ are disucssed by Fiacco, et. al., in reference [27].

The NETOPT program incorporates four methods to minimize a subproblem: 1) Newton Raphson technique, 2) modified Newton-Raphson technique, 3) first-order gradient technique, and 4) modified Fletcher-Powell technique. One of these methods is specified by the user through the input data (see Appendix B). The first two methods require the first and second partial derivatives of the performance criterion and constraints with respect to the parameters, while the third and fourth methods require only the first partial derivatives. These derivatives are evaluated by numerical differencing.

The NETOPT Program

A simplified flow chart of the computation procedure used in NETOPT is shown in Figures 19 and 20. Like the FCAP program, NETOPT also has a library in which there exists a subroutine for each type of fluidic component. Presently, the NETOPT component library includes the following components:

1. Proportional amplifier,

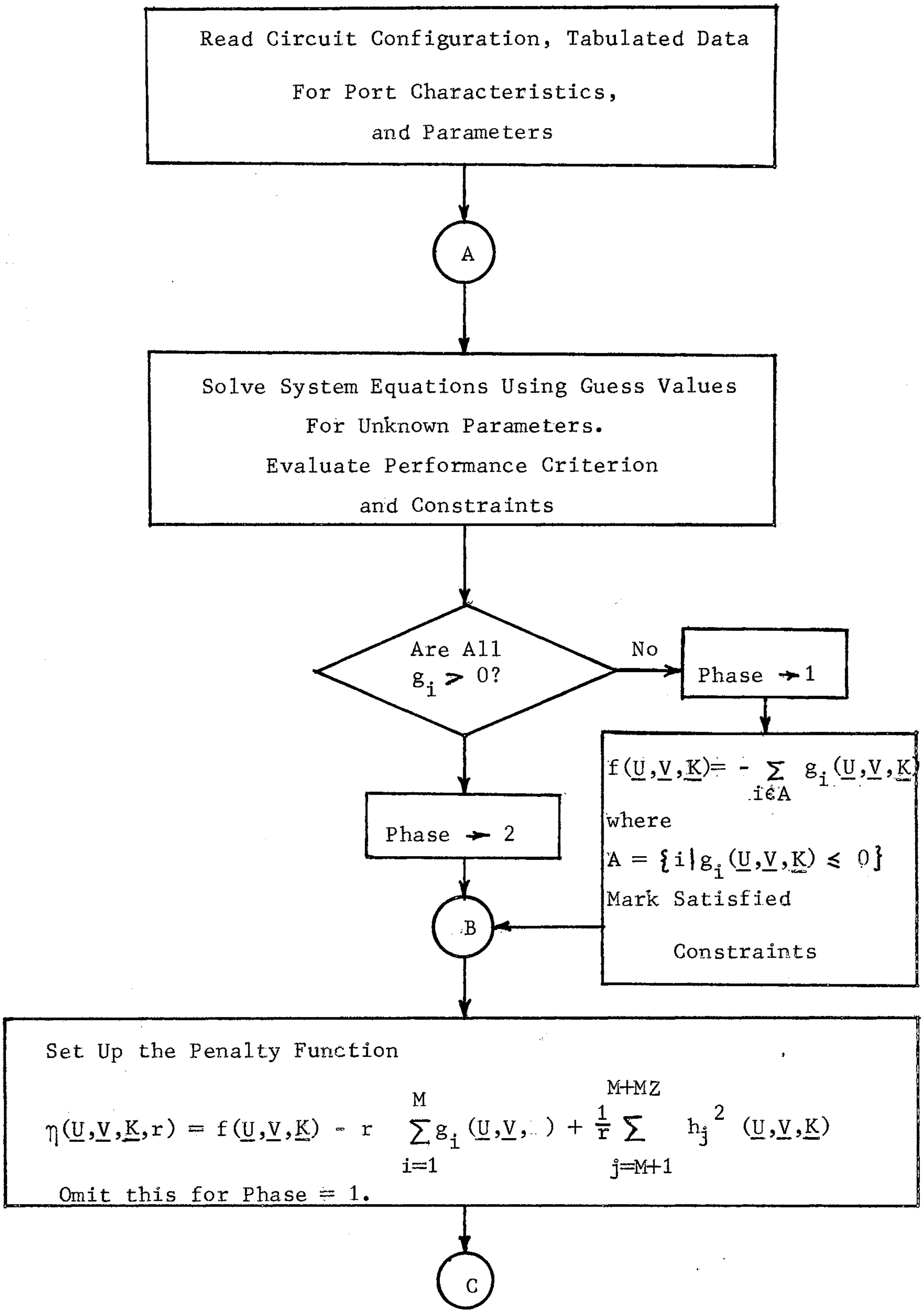


Figure 19. Simplified Flow Chart of NETOPT

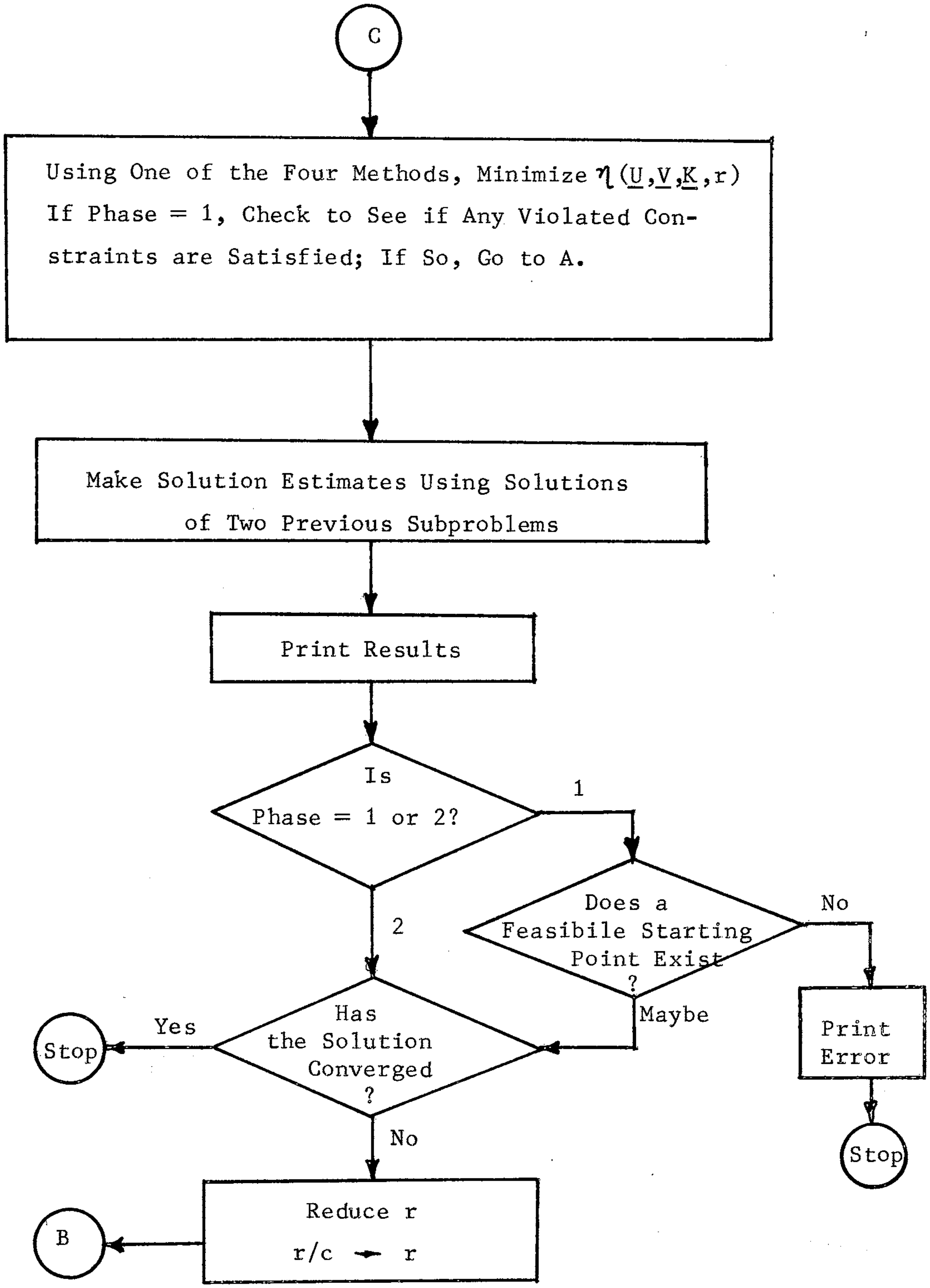


Figure 19 (Continued)

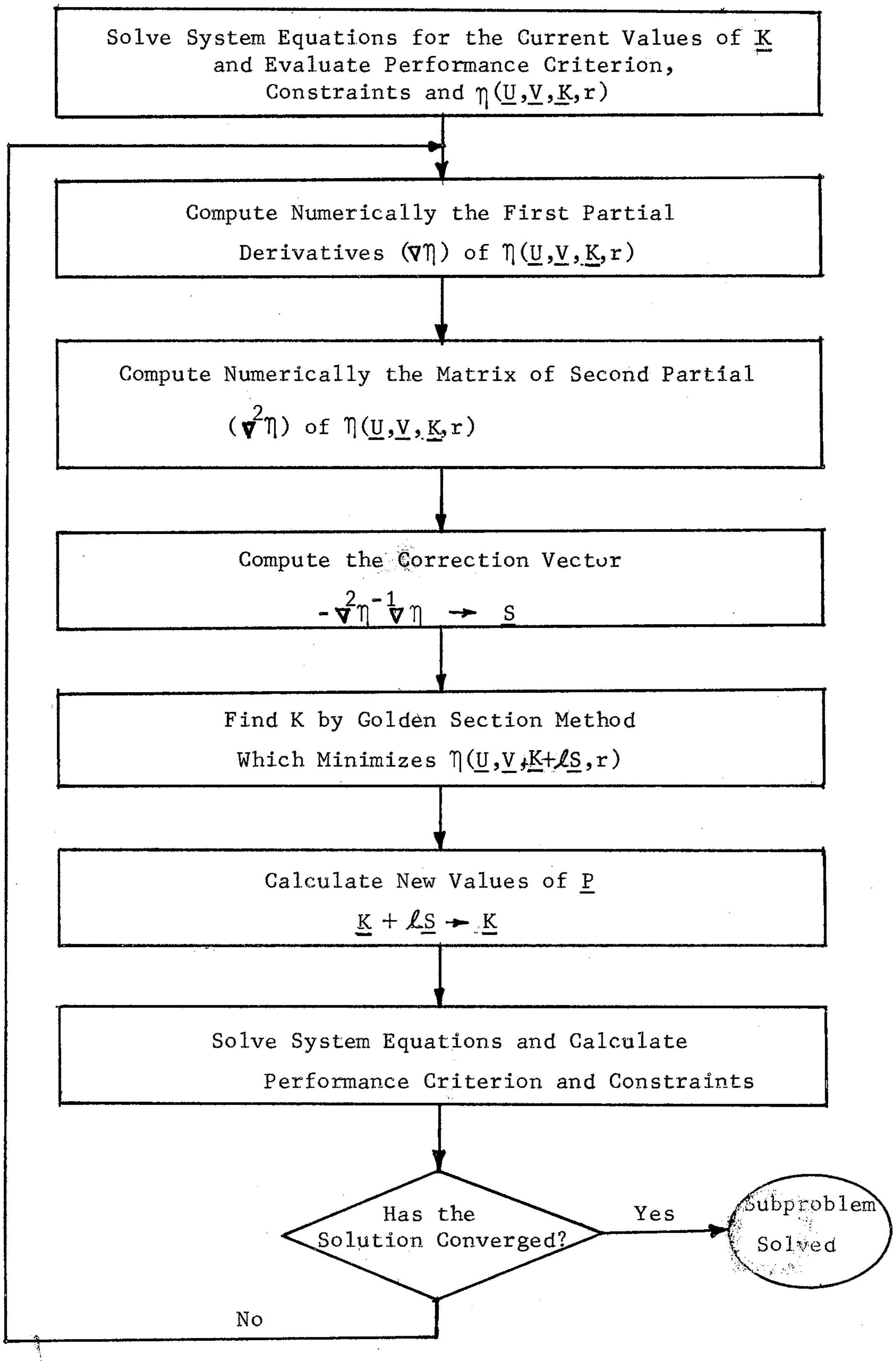


Figure 20. Simplified Flow Chart for Solving a Subproblem Using Newton-Raphson Technique

2. Rectifier,

3. Linear resistor,

4. Tee,

5. Transmission line,

6. Forcing function.

Other types of components may be added easily on a permanent basis (see Appendix B). The static operation of any fluidic circuit, made up of any number (presently limited to 20) of components from the component library, may be optimized using NETOPT. A number of organizational and computational subroutines are also included in the NETOPT program. The organizational subroutines check the input data for errors and set up the control vectors required for the specific problem at hand.

The basic steps a user is required to follow in order to initiate the NETOPT program are:

1. Write the block diagram of the circuit to be optimized,

2. Assign unique numbers between 1 and 20 to each component,

3. Fill a first table which designates interconnections of the various components,

4. Fill a second table which specifies the tabulated data for the components which are represented by phenomenological models,

5. Fill a third table which indicates the ports of different components which have identical characteristics,

6. Fill a fourth table which specifies values of the independent variables at the ports which are not connected to any other components (free ports),

7. Fill a fifth table which specifies the data for the optimization problem, e.g., the number of parameters to be optimized,

the number of inequality and equality constraints, and which parameters are to be optimized,

8. Fill the sixth table which specifies the parameter values for the components,

9. Write the subroutine CNSTNT which supplies the user specified constraints.

The above mentioned tables are transferred to computer cards which form the input data for solving a problem using NETOPT.

Example Problem 3

In the design of control systems, it is often necessary to adjust the gains of cascaded or individual amplifiers to achieve some particular output levels for a given input level. The gain of a proportional amplifier may be adjusted by varying parameters such as the supply pressure or the load resistance attached to the output ports, or a combination of these.

Consider as an example the circuit shown in Figure 21. Suppose it is required to determine the supply pressure (P_{11}) to the proportional amplifier, and the values of the linear resistances R_2 and R_3 which will minimize the power consumption while the circuit operation satisfies the following constraints:

1. The differential pressure gain of the proportional amplifier should be 2.5 ± 0.5,
2. The supply pressure to the proportional amplifier should be between 0.25 to 10.0 psig,
3. Resistance R_2 should be between 0.0 and 5.0 psi/lb/sec,
4. Resistance R_3 should be between 0.0 and 5.0 psi/lb/sec.

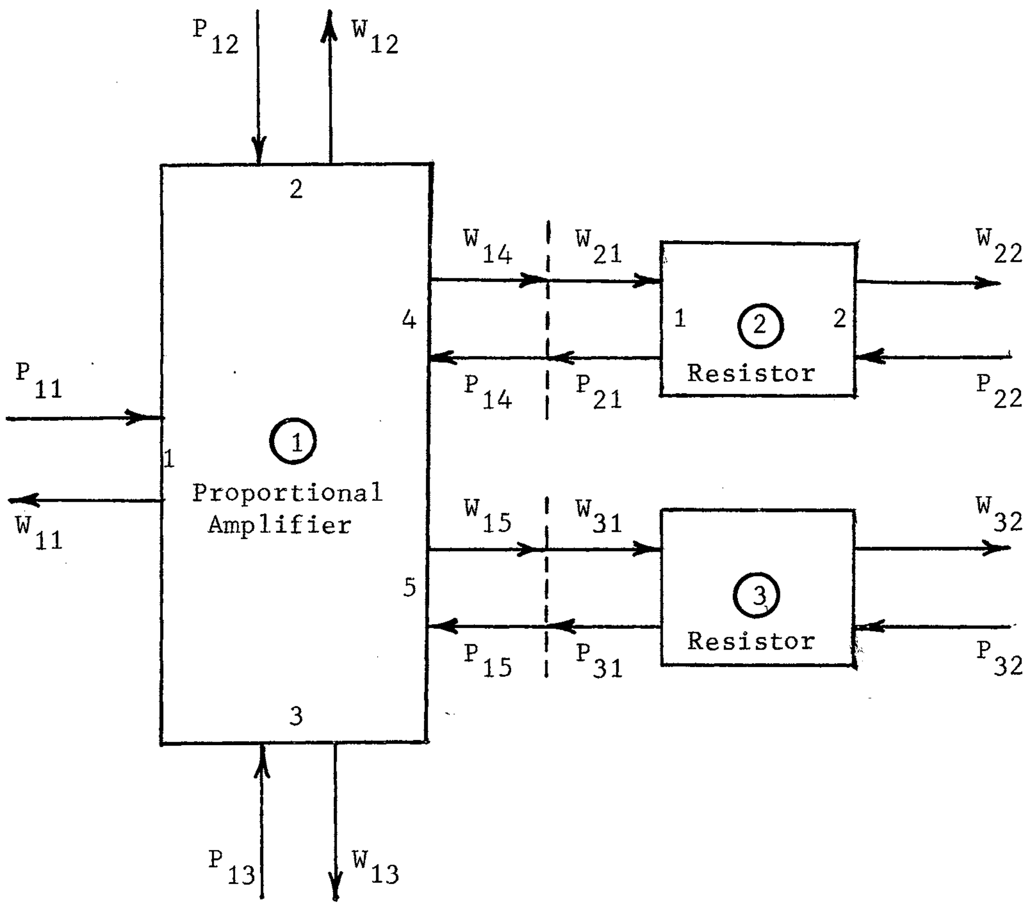


Figure 21. A Fluidic Circuit With a Proportional Amplifier Loaded by Linear Resistors

Mathematically, the problem may be stated as follows:

Minimize

$$f(\underline{U}, \underline{V}) = U_{11}V_{11} \tag{56}$$

Subject to the constraints:

$$2.0 \leq G \leq 3.0 \tag{57}$$

$$P_{11} - 0.25 \geq 0.0 \tag{58}$$

$$10.0 - P_{11} \geq 0.0 \tag{59}$$

$$5.0 - R_2 \geq 0.0 \tag{60}$$

$$5.0 - R_3 \geq 0.0 \tag{61}$$

where G is the differential pressure gain of the amplifier.

The above problem was solved using the NETOPT program described in Appendix B. The input data to NETOPT consisted of: 1) the description of the network configuration, 2) the tabulated data for the static characteristics of the proportional amplifier (GE AW 32) and 3) the data for the inputs, the starting values for the parameters, and the upper and lower limits of the gain. The constraints given in Equations (58) through (61) were supplied through the program CNSTNT.

The inputs at free ports and the starting values of the parameters were assumed to be:

$$U_{12} = 0.05 \text{ Psi}$$
$$U_{13} = 0.05 \text{ Psi}$$
$$U_{22} = 0.0 \text{ Psi}$$
$$U_{32} = 0.0 \text{ Psi}$$

$P_{11} = 1.5$ Psi

$R_2 = 0.1285$ Psi / lbm / Sec

$R_3 = 0.0133$ Psi / lbm / Sec

The power consumption of the circuit (power jet of proportional amplifier) corresponding to the starting values was 104.2 units; and the differential gain of the proportional amplifier was 4.7.

The NETOPT solution converged after solving two subproblems. The first subproblem involved five penalty function evaluations while the second subproblem involved ten penalty function evaluations. The final results were:

Power consumption = 4.3 units

Gain = 2.53

Supply pressure P_{11} = 0.25 psig

Resistance R_2 = 0.00562 psi/lbm/sec

Resistance R_3 = 0.0153 psi/lbm/sec

As expected, the results show that the supply pressure has been reduced to its lower limit. This is because the lower limit on the load resistors was zero. Hence, the program has determined the values of the load resistors such that for a supply pressure of 0.25 psig, the differential gain is 2.53.

CHAPTER VI

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

Summary

Computer-aided techniques have been developed for the static analysis, the transient analysis, and the static optimization of proportional fluidic control systems. In the static analysis, methods have been developed for determining 1) the operating point of the circuit for large signal inputs, and 2) the linearized gain for small signal variations around the operating point. The static models for the components may be characterized either by analytic equations or by tabulated data derived from experimental measurements or manufacturer's catalog sheets. A phenomenological model has been derived for describing the nonlinear dynamic behavior of a proportional fluidic amplifier. This model is based on the geometry and the measured static characteristics of the device.

A completely automatic user-oriented program, FCAP, has been developed for the static and dynamic analysis of fluidic systems. The program produces engineering solutions without requiring the user to write any type of equations or do any real programming. Some principal features of FCAP are:

1. The static models of components may be described by tabulated data or analytic functions,
2. The program automatically initializes all the state variables

to the steady state values at the starting time.

3. FCAP has a component library which contains programs for most of the commonly used fluidic devices. Hence, complex fluidic systems can be "built" as an assemblage of components from the component library.

4. Models for new components and refined models for existing components may be added permanently to the FCAP library by the user.

A synthesis technique has been developed for optimizing the static operation for fluidic systems. The technique determines the optimum design parameters which minimize the power consumption while the circuit operation satisfies certain parameter and output constraints. A user oriented NETwork OPTimization program, NETOPT, has been developed. Some special features of NETOPT are:

1. Static models for components may be described by tabulated data or analytic functions.

2. NETOPT has a permanent component library of static models for most of the commonly used proportional fluidic devices.

3. Functional and regional constraints which are general functions of output variables and parameters may be specified.

4. Gains between some pairs of points in the circuit may be constrained. This constraint has been programmed and is internal to NETOPT.

5. Models for new components and refined models for existing components may be added permanently to the NETOPT library.

Example problems have been worked using FCAP and NETOPT. Results of some simple problems have been verified with graphical analysis.

Conclusions

The automatic circuit analysis computer program, FCAP, is a convenient and an efficient tool for the static and dynamic analysis of proportional fluidic circuits. It alleviates the problems associated with the presently used experimental and hand analysis techniques, and provides a rigid approach to the nonlinear analysis of complex fluidic circuits.

The computer program, NETOPT, is the first step in the computer-aided design of proportional fluidic circuits. This user-oriented circuit optimization program eliminates most of the guesswork, intuition and tedium from the presently used "cut and try" techniques.

Step responses of the proportional fluid amplifier computed using the phenomenological model presented in this thesis agree well with those obtained using the Boothe linear model for small input signals. Results shown in Figure 13 indicate that the linear assumption is valid only for very small perturbation in the input amplitude and that the nonlinear effects start dominating as the input amplitude increases.

Recommendations for Further Study

It is recommended that further developments of the analysis programs FCAP and NETOPT be centered around the following areas:

1. An input language should be developed so that the input data can be provided through a free-format, easy-to-learn, engineer-oriented language.
2. Provision should be made for adding components to the FCAP and NETOPT libraries on a temporary basis.

3. Other methods of representing the tabulated data should be explored with storage and accuracy in view.
4. NETOPT should be extended to include other performance criteria, especially dynamic criteria. The interplay between static and dynamic optimization should be explored.
5. Step response experiments should be conducted on active proportional devices of known geometry in order to validate the nonlinear dynamic modeling approach presented.

BIBLIOGRAPHY

1. Dawson, D. F., F. F. Kuo, and W. E. Magnuson. "Computer-Aided Design of Electronic Circuits--A User's Viewpoint." Proc. IEEE, Vol. 55, No. 11 (November, 1967), 1946-1954.

2. Lindgren, A. G. "Computer-Aided Circuit Design: A User's Viewpoint." Simulation, Vol. 16, No. 5 (May, 1971), 195-216.

3. Malmberg, A. F., F. L. Cronwell, and F. N. Hofer. "NET-1 Network Analysis Program." Los Alamos Scientific Laboratory, Los Alamos, New Mexico, Rept. LA-319, 7090/94 Version, August, 1964.

4. Mathers, H. W., S. R. Sedore, and J. R. Sents. "Automated Digital Computer Program for Determining Responses of Electronic Circuits to Nuclear Radiation (SCEPTRE)." Vol. 1, IBM Space Guidance Center, Owego, New York, IBM File 66-928-611, February, 1967.

5. Milliman, L. D., W. A. Massena, and R. H. Dickhaut. "CIRCUS--A Digital Computer Program for Transient Analysis of Electronic Circuits--User's Guide." The Boeing Company, Seattle, Washington, Harry Diamond Labs., Rept. AD-346-1, January, 1967.

6. Calahan, D. A. "Linear Network Analysis and Realization Digital Computer Programs: An Instruction Manual." University of Illinois Bulletin, Vol. 62, No. 58 (February, 1965).

7. "1620 Electronic Circuit Analysis Program [ECAP] [1620-EE-02X] User's Manual." IBM Application Programs File 420-0170-1, 1965.

8. Dertouzous, M. L. "CIRCAL: On-Line Circuit Design." Proc., IEEE, Vol. 55 (May, 1967), 637-654.

9. Katzenelson, J. "AEDNET: A Simulator for Nonlinear Networks." Proc., IEEE, Vol. 54 (November, 1966), 1536-1552.

10. Chua, L. O. Introduction to Nonlinear Network Theory. New York: McGraw-Hill Book Company, 1969.

11. Walden, J. M. "Digital Computer Analysis of Systems With Nonlinear Elements." (Unpub. Ph.D. dissertation, Oklahoma State University, May, 1966).

12. Kirshner, J. M. "Some Topics in Fluid Circuit Theory." First International Conference on Fluid Logic and Amplification, Cranfield, England, September, 1965.

13. Lechner, T. J., and M. W. Wambsganns. "Proportional Power Stages for Impedance Mathcing." Proc. of the Fluid Amplifications Symposium, October, 1962.

14. Dexter, E. M. "A Technique for Matching Pure Fluid Components Applied to the Design of a Shift Register." Proc. of the Fluid Amplifications Symposium, October, 1962.

15. Brown, F. T. "A Combined Analytical and Experimental Appraoch to the Development of Fluid Jet Amplifiers." ASME Symposium on Fluid Jet Devices, November, 1962.

16. Katz, S., and R. J. Dockery. "Staging of Proportional and Bistable Fluid Amplifiers." Fluid Amplification 11, Rept. Tr-1165, Washington, D. C., Harry Diamond Labs., August, 1963.

17. Shinn, J. N., and W. A. Boothe. "Connecting Elements Into Circuits and Systems." Control Engineering, September, 1964.

18. Parker, G. A., and M. K. Addy. "A Linear Static Matching Technique for Beam-Deflection Amplifier Circuits." Fourth Cranfield Fluidics Conference, Coventry, England, March, 1970.

19. Belsterling, C. A., and K. C. Tsui. "Applications Techniques for Proportional Pure Fluid Amplifiers." Proc. of the Fluid Amplification Symposium, May, 1964.

20. Brown, F. T. "On the Stability of Fluid Systems." Proc. of the Fluid Amplifications Symposium, May, 1964.

21. Boothe, W. A. "A Lumped Parameter Technique for Fluid Jet Amplifiers." JACC, June, 1964.

22. Healey, A. J. "Vent Effects on the Response of a Proportional Amplifier." ASME Paper No. 67-WA/FE-12.

23. Manion, F. M. "Proportional Amplifier Simulation." Advances in Fluidics, May, 1967.

24. Reid, K. N. "Static Characteristics of Fluid Amplifiers." Proc. Fluid Power Research Conference, Oklahoma State University, Stillwater, Oklahoma, July, 1967.

25. Smith, C. K. "HYDSIM User's Manual." School of Mechanical Engineering, Oklahoma State University, Stillwater, Oklahoma, August, 1973.

26. Mylander, W. C., R. L. Holmes, and G. P. McCormick. "A Guide to SUMT-Version 4." Research Analysis Corporation, McLean, Virginia, October, 1971.

27. Fiacco, A. V., and G. P. McCormick. Nonlinear Programming: Sequential Unconstrained Minimization Techniques. New York: John Wiley and Sons, Inc., 1968.

APPENDIX A

A USER'S GUIDE TO THE FCAP PROGRAM

A.1. Introduction

The traditional hand calculation and experimental methods employed in the analysis of proportional fluidic circuits are very tedious, and become increasingly complex as the size of the circuit increases. The fluidic circuit analysis program, FCAP, provides an efficient, computerized approach to the determination of the static and dynamic performance of fluidic circuits.

This appendix presents a guide to the use of the FCAP program. Section A.2 describes the main features of FCAP, Section A.3 explains the details of preparation of input data for a FCAP run, Sections A.4 through A.8 present the output and diagnostic information, Section A.9 describes the component models stored in the FCAP library, and Section A.10 describes the procedure for adding new component models or changing the existing ones in the FCAP library.

A.2. Capabilities and Features of FCAP

FCAP is a user-oriented, circuit analysis program. In general, the user is not required to do any programming. Tables are provided for the user to enter data such as circuit topology, component types, component steady state characteristics, component parameters, and other information relating to the particular type of analysis to be conducted and the constraints to be employed. These data are transferred from the tables to standard punched cards for use as input data to the program.

Models for most common proportional circuit components are permanently stored in an FCAP library. Two types of models are included: 1) analytical--requiring parameter values only to be supplied as input data, and 2) phenomenological--requiring measured static characteristics or data from manufacturer's catalogs to be supplied as input data. A provision is made for the user to supply new or refined models on a permanent basis when desired. The program may be used to analyze any proportional circuit, including electronic, mechanical, pneumatic or hydraulic, providing suitable models are added to the component library.

FCAP automatically "assembles" the circuit on the basis of input circuit topology data, and checks for inconsistencies in the data set. There is virtually no limitation on the circuit that can be analyzed. The output data normally includes a listing of the user-supplied input data, and a printout of the user-requested port variables and/or state variables.

FCAP may be used to determine the steady-state operating point of a circuit, the small signal pressure or flow gain between specified points in a circuit, and the dynamic behavior of a circuit when subjected to a specified disturbance input. Nonlinear algebraic equation sets are solved within the program using a Newton-Raphson technique and nonlinear differential equations are solved using a Runge-Kutta technique.

A.3. Preparation for a FCAP Run

Eight basic steps are required in the preparation of the data for a FCAP run.

1. Sketch a FCAP block diagram,
2. Fill a circuit data table,

3. Fill a port data table,
4. Fill an equality table,
5. Fill a free port table,
6. Fill a gain data table,
7. Fill a parameters table,
8. Transfer the tables onto computer cards and arrange them in the proper order.

These steps are explained in detail below.

A.3.1. Sketching the Block Diagram

A conventional circuit diagram with standard fluidic symbols is transferred into block diagram form using the FCAP component symbols shown in Table VIII. The FCAP block diagram aids the user in filling the circuit data table and checking the port connections. The following rules must be observed when preparing the block diagram:

1. The symbol for each component must be identical to the FCAP component symbols shown in Table VIII,
2. The connections between the ports must match as shown in Table IX,
3. The port and component type numbers of each FCAP component symbol must be recorded exactly as shown in Table VIII,
4. A unique number (component number) between 1 and 20 must be assigned to each component. This number may be written below the component type number.

The component type numbers and the port numbers on the component symbols in Table VIII _must never be changed_. The user-selected component numbers are used by the computer to identify the particular

TABLE VIII

FCAP AND NETOPT COMPONENT SYMBOLS

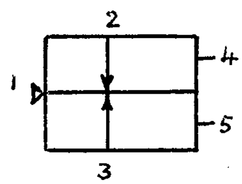
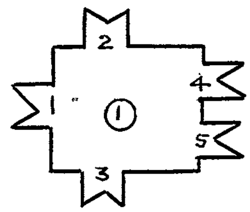
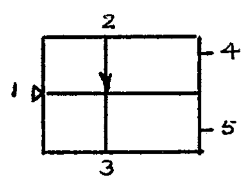
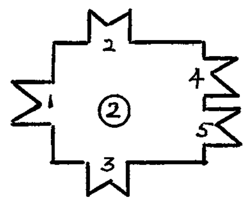
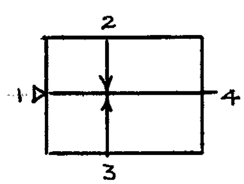
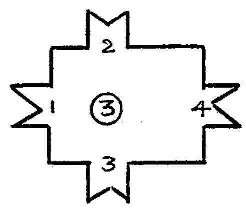
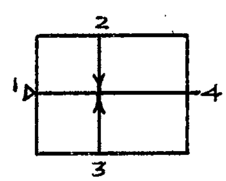
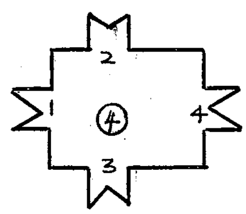
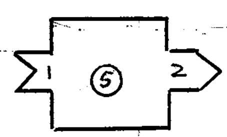
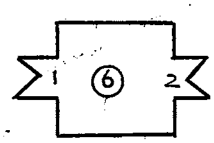
Component Name	Component Type Number	Fluidic Symbol	FCAP/NETOPT Component Symbol	Ports
Proportional Amplifier (Algebraic)	1			1-Supply 2,3-Control 4,5-Output
Proportional Amplifier (Dynamic)	2			1-Supply 2,3-Control 4,5-Output
Rectifier (Algebraic)	3			1-Supply 2,3-Input 4-Output
Rectifier (Dynamic)	4			1-Supply 2,3-Input 4-Output
Resistor 1	5			
Resistor 2	6			

TABLE VIII (Continued)

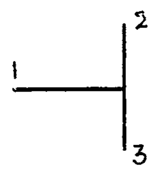
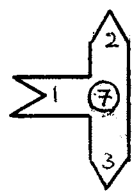
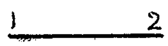
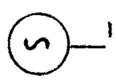
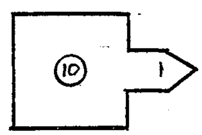
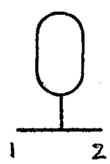
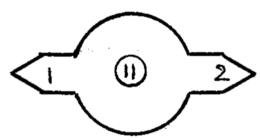
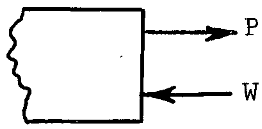
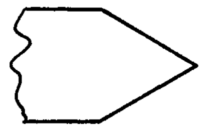
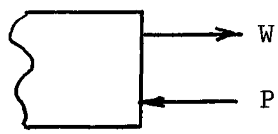
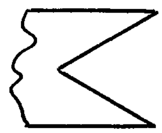
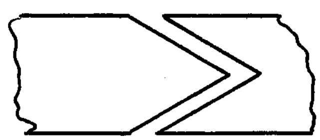
Component Name	Component Type Number	Fluidic Symbol	FCAP/NETOPT Component Symbol	Ports
Tee	7			
Line	8			
Forcing Function	10			
Capacitor	11			

TABLE IX

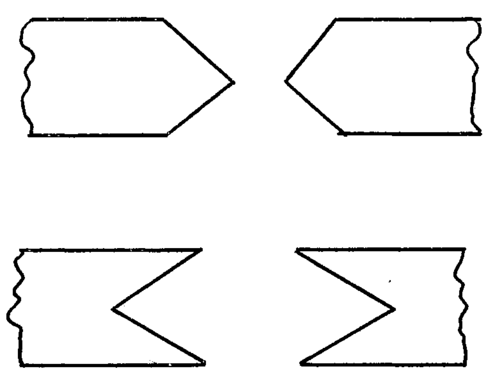
PORT CONNECTIONS

Multi-Port Symbol	FCAP/NETOPT Symbol	Port Type
		1
		2

Valid Port Connection



Invalid Port Connection



component.

A.3.2. Circuit Data Table

The circuit data table shown in Figure 22 consists of the Option Card and the Circuit Connection Table. The Option Card specifies the type of analysis to be conducted, and some associated information. The entries which are not applicable may be left blank on the Option Card. A problem title also may be specified through the Option Card.

The entries in the Option Card are:

1. IOPT: Analysis option,

   = 1 for operating point analysis

   = 2 for gain analysis

   = 3 for dynamic analysis,

2. IGAIN: Type of gain (for gain analysis only),

   = P or blank for pressure gain

   = F for flow gain,

3. TSTART: Initial time for the dynamic simulation,
4. TFINAL: Final time for the dynamic simulation,
5. DT: Step size for dynamic simulation
6. NWRITE: Number of integration time steps between printouts,
7. TITLE: A comment to be printed at the beginning of the problem.

The Circuit Connection Table specifies the interconnections between components. The FCAP block diagram will aid in the filling of this table. The fields for the various entries in the Circuit Connection Data Table are shown in Figure 22. All the integer numbers must be right justified.

The entries in the Circuit Connection Table are:

FCAP/NETOPT CIRCUIT DATA TABLE	IOPT		IGAIN		TSTART	TFINAL	DT	NWRITE	TITLE
CARD COL. #	1		6		11 – 20	21 – 30	31 – 40	41 – 45	46 – 78
OPTION CARD									

CIRCUIT CONNECTION TABLE				PORT 1					PORT 2					PORT 3					PORT 4					PORT 5				STATE VARIABLES TO BE PRINTED				
	COMP. #	TYPE #		FREE ?	COMP. #	PORT #	PRINT ?		FREE ?	COMP. #	PORT #	PRINT ?		FREE ?	COMP. #	PORT #	PRINT ?		FREE ?	COMP. #	PORT #	PRINT ?		FREE ?	COMP. #	PORT #	PRINT ?	1	2	3	4	5
CARD COL. # / COMP. NAME	10 11	12 13		16 17	18 19	20 21	22 23		26 27	28 29	30 31	32 33		36 37	38 39	40 41	42 43		46 47	48 49	50 51	52 53		56 57	58 59	60 61	62 63	64 65	66 67	68 69	70 71	72 73

Figure 22. FCAP/NETOPT Circuit Data Table

1. Comp. # (Col. 10-11): User assigned number for the component. The rest of the data in this row refers to this component.

2. Type # (Col. 12-13): Type number for the component.

3. Port 1 (Col. 16-23)[1]: Data about the component connected to port 1.

   Free? (Col. 16, 17): Is port 1 free or connected to another port?
   = 1 if free
   = 0 if connected;

   Comp. # (Col. 18, 19): Number of the component connected to port 1, if port 1 is connected,
   = 0 if port 1 is free;

   Port # (Col. 20, 21): Port number of the component connected to port 1, if port 1 is connected,
   = 0 if port 1 is free;

   Print? (Col. 22, 23): Should the dependent and independent variables at port 1 be printed?
   = 1 if yes, 0 otherwise.

4. Port 2 (Col. 26-33)[2]: Data about component connected to port 2. Entries have similar meanings as in port 1.

---

[1] It is sufficient to furnish the data at only one of the two ports for the connection between two ports. The data for the other port may be ignored.

[2] If a component has less than 5 ports, then the data for the ports not applicable are ignored.

5. Port 3 (Col. 36-43): Data about component connected to port 3. Entries have similar meanings as in port 1.

6. Port 4 (Col. 46-53): Data about component connected to port 4. Entries have similar meanings as in port 1.

7. Port 5 (Col. 56-63): Data about component connected to port 5. Entries have similar meanings as in port 1.

8. State variables to be printed (Col. 64-73): Five state variables of the component (Col. 10-11) may be printed. These may be specified in Columns 64-73 in I_2 format.

NOTE: Columns 1-9 should be left blank.

A.3.3. Port Data Table

The tabulated data for the static characteristics of the ports of components are supplied in this section. The maximum number of points per characteristic is limited to 50. The first card in the data for each port must contain only the component and port numbers; the portion under the variables must be left blank (see Figure 23). The component number and port number may be left blank on the subsequent cards. The values of the variables must be entered on the remaining cards for this port. The variable values are furnished in the floating point form. Variable 1, Variable 2, and Variable 3 for the various ports of components are specified in the model specification for each component in

FCAP/NETOPT PORT DATA TABLE	COMP. #			PORT #			VARIABLE 1	VARIABLE 2	VARIABLE 3
CARD COL. #	11	12		18	19		31 - 40	41 - 50	51 - 60
							LEAVE THIS PORTION BLANK ON THE FIRST CARD OF EACH PORT		

Figure 23. FCAP/NETOPT Port Data Table

Section A.9 on the component library.

The input characteristics may be specified for only one of the two input ports of a symmetrical proportional amplifier or rectifier. The characteristics for the other port are automatically transferred. Similarly, the characteristics of only one output port may be specified for a symmetrical proportional amplifier; the data for the other output port are transferred automatically.

A.3.4. Equality Table

If the same type of component is used in more than one place in a circuit, then tabulated data (if needed) may be furnished only for one of the components in the Port Data Table. The same data may be transferred to other components through specifications in the Equality Table (Figure 24). The characteristics may be equivalenced one port at a time. This is done to enable the user to transfer the data for any port of a component to any other port of a different component with minimum effort. The equivalences in the Equality Table override the implicit transfer of port characteristics of symmetrical amplifiers in the Port Data Section. The data in the Equality Table need not be ordered. The entries in the Equality Table are:

1. Comp. # (Col. 11-12): Number of the component whose port requires the data transfer,
2. Port # (Col. 18-19): Number of the port requiring data transfer,
3. Comp. # (Col. 25-26): Number of the component whose port characteristics are known,
4. Port # (Col. 32-33): Number of the port whose characteristics

FCAP/ NETOPT EQUALITY TABLE	COMP. #			PORT #			COMP. #			PORT #	
CARD COL.#	11	12		18	19		25	26		32	33

Figure 24. FCAP/NETOPT Equality Table

are known. (The characteristics for this port may have been given either in the Port Data Table or in the Equality Table. If it is given in the Equality Table, then it need not appear prior to the data presently being furnished.)

A.3.5. Free Port Table

Data for the independent variables at ports which are not connected to any other component are furnished in this table (Figure 25). Only the numerical value of an independent variable must be supplied. If a component has more than one free port, then data may be supplied in any order. The various entries in the Free Port Table are:

1. Comp. # (Col. 9-10): Number of the component whose free port data are being furnished,
2. Port # (Col. 11-12): Number of the port which is free,
3. Value of Ind. Variable (Col. 15-24): Numerical value of the independent variable at port stated in Col. 11-12.
4. Port # (Col. 25-26): Number of the port which is free,
5. Value of Ind. Variable (Col. 29-38): Numerical value of the independent variable at port stated in Col. 25-26,
6. Port # (Col. 39-40): Number of the port which is free,
7. Value of Ind. Variable (Cols. 43-52): Numerical value of the independent variable of port stated in Col. 39-40,

FCAP/ NETOPT FREEPORT TABLE	COMP #		1				2				3				4			
			PORT #			VALUE OF IND. VAR.	PORT #			VALUE OF IND. VAR.	PORT #			VALUE OF IND. VAR.	PORT #			VALUE OF IND. VAR.
CARD COL. #	9	10	11	12		15-24	25	26		29-38	39	40		43-52	53	54		57-66

Figure 25. FCAP/NETOPT Free Port Table

8. Port # (Col. 53-54): Number of the port which is free,

9. Value of Ind. Variable (Col. 57-66):
Numerical value of the independent variable at port stated in Col. 53-54.

A.3.6. Gain Data Table

This table must be filled only if the gain analysis option is specified. In one run, gains may be calculated from one point in the circuit to 10 different points in the circuit. Reruns may be made and gains may be calculated from a different point by having a card for each run. The ports where the input signals are applied must necessarily be free. The entries in the Gain Data Table are (Figure 26):

1. Input at: Data regarding the place where the inputs are applied are furnished here,

   Comp. # (Col. 11-12): Number of the component to which the inputs are applied,

   Port # (Col. 15): Number of the port at which the input is applied,
   = 0 if the input is differential, (The input ports must be free.)

2. Input signal change (Col. 16-25):
Numerical value of the perturbation in the input signal. (Default value = 0.005),

3. Gain From: Data for the component from where the gain has to be computed. This column may be ignored if the entries in this

FCAP

GAIN DATA

TABLE

	INPUT AT				INPUT SIGNAL CHANGE	GAIN FROM				GAIN TO																																							
										1				2				3				4				5				6				7				8				9				10			
	COMP.#			PORT #		COMP.#			PORT #	COMP.#			PORT #	COMP.#			PORT #	COMP.#			PORT #	COMP.#			PORT #	COMP.#			PORT #	COMP.#			PORT #	COMP.#			PORT #	COMP.#			PORT #	COMP.#			PORT #	COMP.#			PORT #
CARD COLUMNS	11	12		15	16-25	26	27		30	31	32		35	36	37		40	41	42		45	46	47		50	51	52		55	56	57		60	61	62		65	66	67		70	71	72		75	76	77		80

Figure 26. FCAP Gain Data Table

column are the same as those in "Input At" column,

Comp. # (Col. 26-27): Number of the component from where the gain has to be calculated,

Port # (Col. 30): Port number of the component in Col. 26-27 from where the gain has to be calculated;
= 0 for differential control signal,

4. Gain Up To: Data for the component up to which gain has to be computed. If the component is either a proportional amplifier or rectifier, then the gains are computed up to its output ports,

Comp. #: Number of the component up to which gain has to be computed,

Port #: Port number of the component up to which gain is to be computed,
= 0 for differential gain.

A.3.7. Parameter Table

The parameters for components with analytic models are furnished in this table (Figure 27). Seven parameters may be furnished per card. If the component has more than seven components, data may be continued on the next card. No identification is necessary to indicate that it is a continuation card. The component number may be omitted on the continuation card. The Parameter Table has the following entries:

1. Comp. #: Number of the component whose parameters

FCAP/ NETOPT PARAMETER TABLE	COMP. #		PARAMETERS FOR COMPONENTS						
CARD COL. →	9	10	11-20	21-30	31-40	41-50	51-60	61-70	71-80

Figure 27. FCAP/NETOPT Parameter Table

are being furnished,

2. Parameters for component:

The physical quantities that the parameters represent are given in model specifications in Section A.9 on the Component Library. These parameters must be furnished consecutively. Seven parameters may be furnished per card. If there are more than seven parameters, they may be continued on the next card. No indication is necessary for continuation.

A.3.8. Arrangement of Data Cards

Figure 28 shows the arrangement of data cards for a FCAP run. The cards with names PORTDATA, EQUALITY, FREEPORT, GAINDATA, and PARAMTRS are placed at the beginning of the Port Data, Equality Data, Free Port Data, Gain Data and Parameters Data cards respectively. These names are keywords which transfer the control to various sections in the program, and hence should never be changed. The keywords must be punched in the first eight columns of the card. The data section on gain data may be omitted if gain analysis is not required. An "END" card with END punched anywhere in Columns 1-8 must be placed at the end of the data as shown in Figure 28.

A.4. FCAP Output

The FCAP program prints out the user-supplied input data, and the

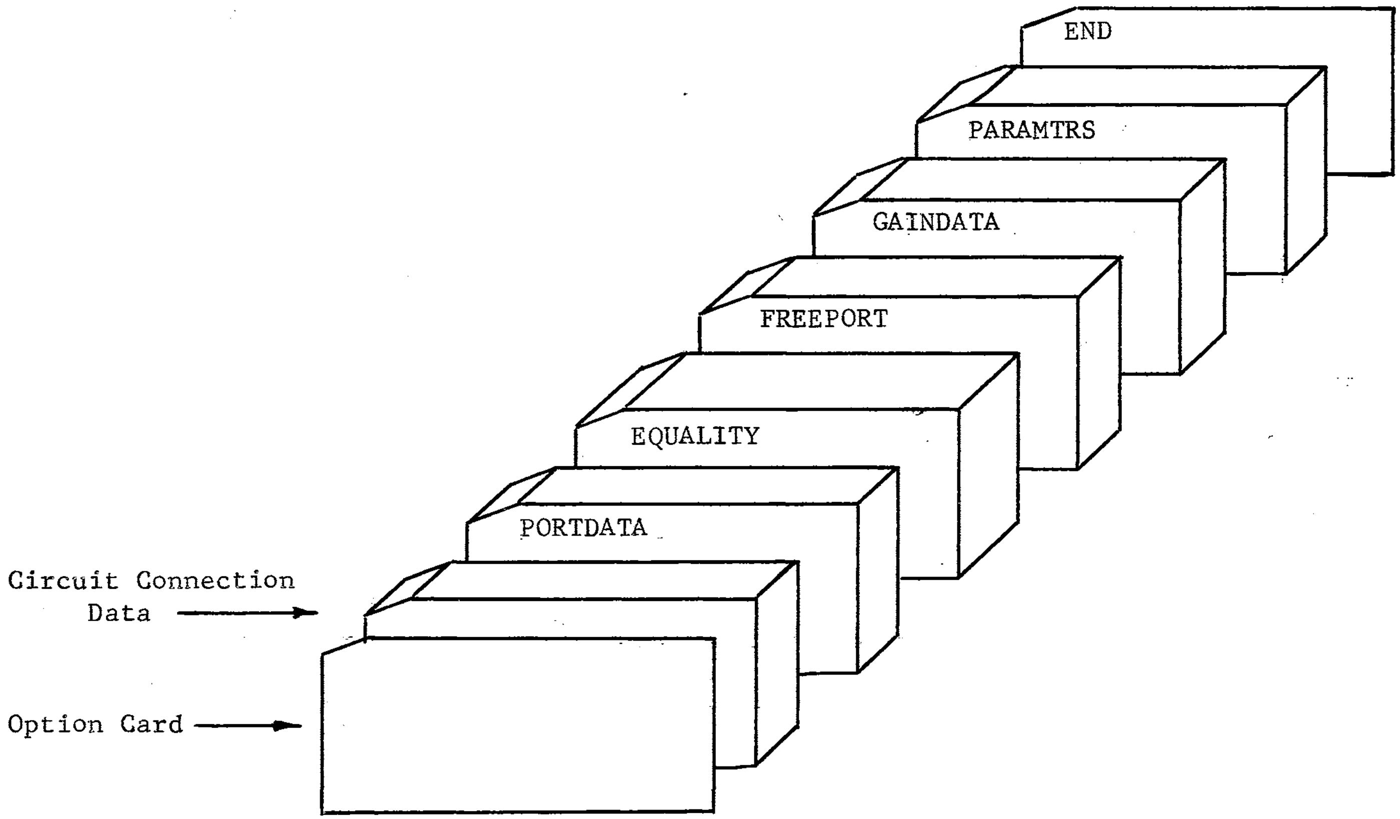


Figure 28. Arrangement of Data Cards for FCAP

results of the analysis. The input data are printed for user verification and permanent record. If errors are detected in the data, they are printed immediately below the appropriate data item.

There is no limitation on the number of port variables that can be printed. However, the programmer is discouraged from printing redundant and unnecessary port variables. The number of state variables that may be printed per component is limited to five.

The symbols and units of the port variables are:

Pressure - psig - P

Flow - lbm/sec - W.

These units are printed along with the output. The units of the state variables are not printed. These may be obtained from the component equations.

A.5. Step Sizes for Dynamic Simulation

It is often difficult to estimate accurately the step size required for the simulation of a nonlinear dynamic system. The selection of the largest step size for a given problem depends upon the following factors:

1. The digital computer can store only a finite number of significant digits. Hence, if time steps below a certain limit are chosen, truncation errors result. This will offset the advantage gained in accuracy due to a reduced time step.
2. Too large a step size makes the numerical solution unstable. This is primarily due to the erroneous approximation of the solution function by the difference equation.

As a general rule, the step size used in the dynamic simulation of linear systems is equal to 1/10 to 1/100 of the smallest time constant or shortest period associated with a natural frequency appearing in the system. The same approach may be used in estimating a reasonable step size for nonlinear systems, providing approximate time constants and natural frequencies are determined intuitively or by using linearized analysis.

A.6. Newton-Raphson Convergence

The Newton-Raphson method is used to solve the algebraic equations in the static and dynamic models. The maximum number of Newton-Raphson iterations is presently set at fifty. If the solution to the algebraic equations does not converge within fifty iterations, an error message is printed and the execution of the program is terminated. The probable reasons for the nonconvergence of the Newton-Raphson method are:

1. The number of iterations allowed may not be sufficient. The limit may be increased by increasing the value of the variable NITER in the main program,
2. Using improper values of the parameters may cause this problem. For example, if components using tabulated data are connected to passive resistors, an incorrect value (too large or small) of resistance may cause the flows to be in the wrong directions and the pressures to be negative. This problem may be alleviated by choosing reasonable values for the resistances,
3. If a new component model is being tried, errors in the equations for either the functions or the partial derivatives may cause nonconvergence.
4. The initial estimates for all the algebraic variables default

to zero. Non-zero initial values for some port variables may insure convergence; these may be introduced as parameters (see Section A.10).

A.7. Matrix Singularity

FCAP uses the Gauss-Jordan elimination technique to solve simultaneous linear equations which result 1) during the least square curve or surface fitting of the tabulated data, and 2) during the solution of system equations using the Newton-Raphson method. A singular matrix may result in either case.

A singular matrix may result during curve or surface fitting due to the lack of sufficient number of points. At least (N + 1) points should be furnished for curve fitting, where N is the order of the polynomial; and (NX + 1) x (NY + 1) points should be furnished for surface fitting, where NX and NY are the orders of the polynomials in variable 1 and variable 2 respectively.

A singular matrix may result during the Newton-Raphson iteration due to a poor choice of initial estimates for the port variables. Non-zero initial values for some port variables may sometimes alleviate the problem.

A.8. Program Limitations

The main limitations of the FCAP program are:

1. The number of state variables that may be printed is limited to five,
2. The dimensions of the arrays are presently set up to accomodate 20 components per circuit. This may, however, be increased

easily,

3. Provision has not been made for temporary addition of new components,

4. The input data must be furnished in fixed format.

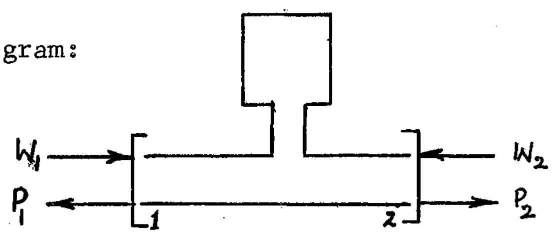
A.9. FCAP Component Library

A.9.1. Capacitor

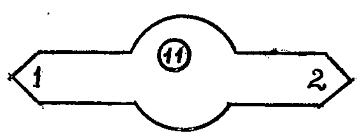
FCAP Name: CAPSTR

Type Number: 11

Component Diagram:



FCAP Component Symbol:



Component Equations:

Assumptions

Walls of the capacitor are rigid

Gas expands isothermally

Parameters

K_1 - R, Gas constant $\frac{in^2}{sec^2\ {}^oR}$

K_2 - T, Temperature (^oR)

K_3 - v, Volume (in^3)

Dynamic Model

Definitions

$U_1 = W_1 \,,\; U_2 = W_2 \,,\; Y_1 = P_2 \,,\; x_1 = P_1$

Model Equations

$$-(W_1 + W_2) = \frac{d}{dt}\left(\rho g \upsilon\right)$$

$$-(W_1 + W_2) = g \upsilon \frac{d\rho}{dt}$$ (from assumption 1)

$$P_1 = \rho R T$$

$$\therefore \frac{dP_1}{dt} = RT \frac{d\rho}{dt}$$

$$\therefore -(W_1 + W_2) = \frac{g\upsilon}{RT} \frac{dP_1}{dt}$$

$$P_2 - P_1 = 0$$

State Equations

$$\frac{dx_1}{dt} = -\frac{K_1 K_2}{g K_3}\left(U_1 + U_2\right)$$

Algebraic Equations

$$Z_1 = Y_1 - x_1 = 0$$

Port Equations

$$V_1 = X_1$$

$$V_2 = Y_1$$

Static Model

Definitions

$$U_1 = W_1 \ , \quad U_2 = W_2 \ , \quad V_1 = P_1 \ , \quad V_2 = P_2$$

Equations

$$U_1 + U_2 = 0$$

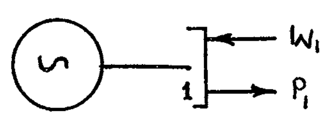
$$V_1 - V_2 = 0$$

A.9.2. <u>Forcing Function</u>

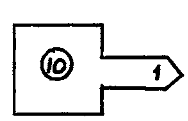
FCAP Name: FRCFUN

Type Number: 10

Component Diagram:



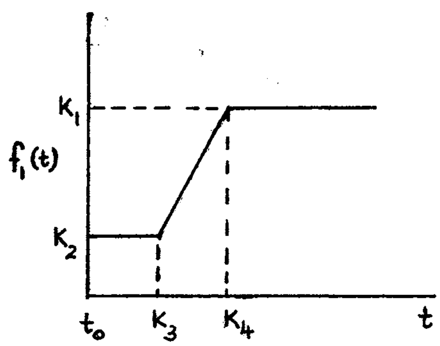
FCAP Component Symbol

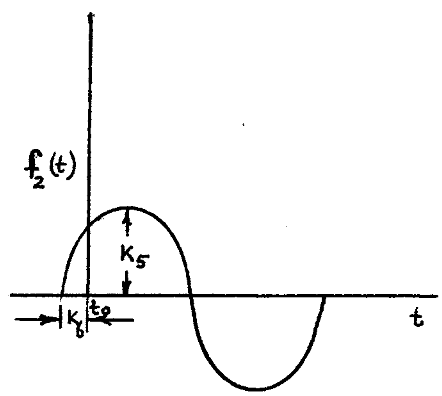


Component Equations

Assumption: None

Parameters:





K_1 - Maximum pressure $(\frac{lbf}{in^2})$

K_2 - Pressure at time t_o $(\frac{lbf}{in^2})$

K_3 - State time of ramp (secs)

K_4 - End time of ramp (secs)

K_5 - Amplitude of sine wave$(\frac{lbf}{in^2})$

K_6 - Phase lag (degrees)

K_7 - Frequency (cps)

Dynamic Model

Definitions

$U_1 = W_1$, $Y_1 = P_1$

Model Equations

$$f_1(t) = \begin{cases} K_2, & t_0 \leq t < K_3 \\ K_2 + (K_1 - K_2)(t - K_3)/(K_4 - K_3), & K_3 \leq t < K_4 \\ K_1, & K_4 = K_3 \\ K_1, & K_4 \leq t \end{cases}$$

$$f_2(t) = K_5 \sin\left(2\pi K_7 (t - t_0) + \pi K_6/180\right)$$

$$f = f_1(t) + f_2(t)$$

State Equations

None

Algebraic Equations

$$Z_1 = f - Y_1 = 0$$

Static Model

Definitions

$$U_1 = W_1, \quad V_1 = P_1$$

Equations

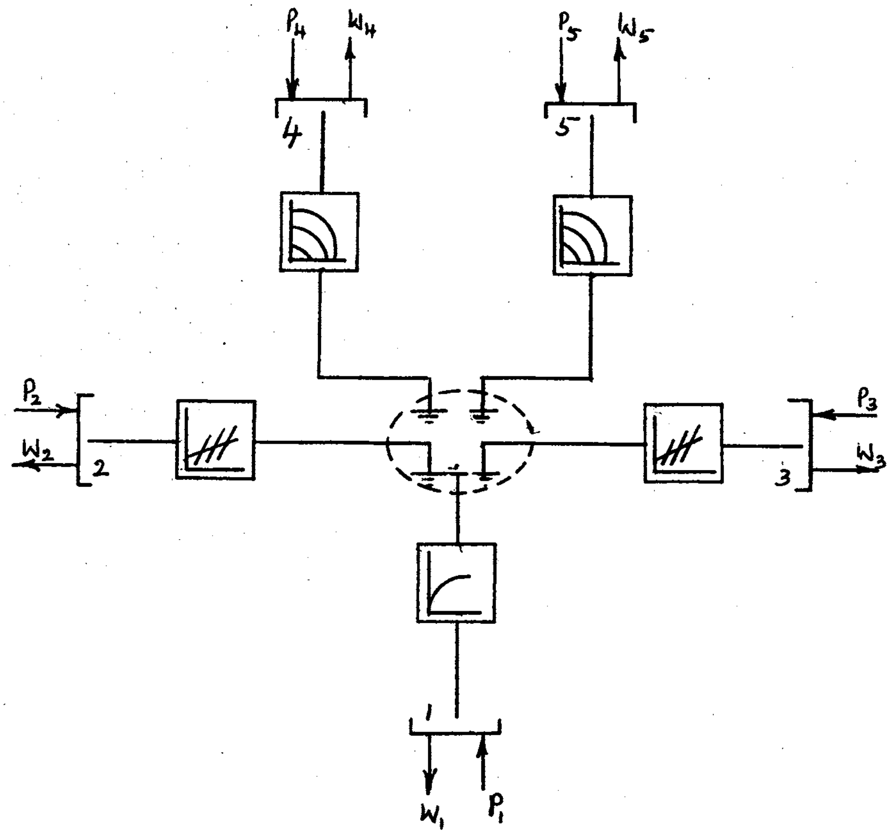
$$G_1 = f - V_1 = 0$$

A.9.3. Proportional Amplifier (Algebraic)

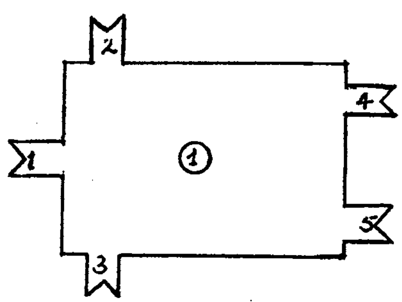
FCAP Name: PRPAMP (1)

Type Number: 1

Component Diagram



FCAP Component Symbol



Component Equations

Assumptions

The static characteristics may be normalized with respect to the supply pressure and flow conditions.

The amplifier is well vented so that the static pressure in the interaction region is atmospheric.

Dynamic effects are negligible.

Parameters

None

Dynamic Model

Definitions

$U_1 = P_1$, $U_2 = P_2$, $U_3 = P_3$, $U_4 = P_4$, $U_5 = P_5$

$Y_1 = W_1$, $Y_2 = W_2$, $Y_3 = W_3$, $Y_4 = W_4$, $Y_5 = W_5$

State Equations

None

Algebraic Equations

$$Z_1 = Y_1 + \sum_i c_{1_i} U_1^{(i-1)} = 0$$

$$Z_2 = Y_2 + Y_1 \sum_j \sum_i c_{2_{ij}} \left(\frac{U_2}{U_1}\right)^{(i-1)} \left(\frac{U_2 + U_3}{U_1}\right)^{(j-1)} = 0$$

$$Z_3 = Y_3 + Y_1 \sum_j \sum_i c_{3_{ij}} \left(\frac{U_3}{U_1}\right)^{(i-1)} \left(\frac{U_2 + U_3}{U_1}\right)^{(j-1)} = 0$$

$$Z_4 = U_4 - U_1 \sum_j \sum_i c_{4_{ij}} \left(\frac{Y_4}{Y_1}\right)^{(i-1)} \left(\frac{U_2 - U_3}{U_1}\right)^{(j-1)} = 0$$

$$Z_5 = U_5 - U_1 \sum_j \sum_i c_{5_{ij}} \left(\frac{Y_5}{Y_1}\right)^{(i-1)} \left(\frac{U_2 - U_3}{U_1}\right)^{(j-1)} = 0$$

Port Equations

$$V_1 = Y_1$$

$$V_2 = Y_2$$

$$V_3 = Y_3$$

$$V_4 = Y_4$$

$$V_5 = Y_5$$

Static Model

Definitions

$$U_1 = P_1\ ,\quad U_2 = P_2\ ,\ U_3 = P_3\ ,\ U_4 = P_4\ ,\ U_5 = P_5$$

$$V_1 = W_1\ ,\quad V_2 = W_2\ ,\ V_3 = W_3\ ,\ V_4 = W_4\ ,\ V_5 = W_5$$

Equations

$$G_1 = V_1 + \sum_i C_{1_i} U_1^{(i-1)} = 0$$

$$G_2 = V_2 + V_1 \sum_j \sum_i C_{2_{ij}} \left(\frac{U_2}{U_1}\right)^{(i-1)} \left(\frac{U_2 + U_3}{U_1}\right)^{(j-1)} = 0$$

$$G_3 = V_3 + V_1 \sum_j \sum_i C_{3_{ij}} \left(\frac{U_3}{U_1}\right)^{(i-1)} \left(\frac{U_2 + U_3}{U_1}\right)^{(j-1)} = 0$$

$$G_4 = U_4 - U_1 \sum_j \sum_i C_{4_{ij}} \left(\frac{V_4}{V_1}\right)^{(i-1)} \left(\frac{U_2 - U_3}{U_1}\right)^{(j-1)} = 0$$

$$G_5 = U_5 - U_1 \sum_j \sum_i C_{5_{ij}} \left(\frac{V_5}{V_1}\right)^{(i-1)} \left(\frac{U_2 - U_3}{U_1}\right)^{(j-1)} = 0$$

Characteristics to be read

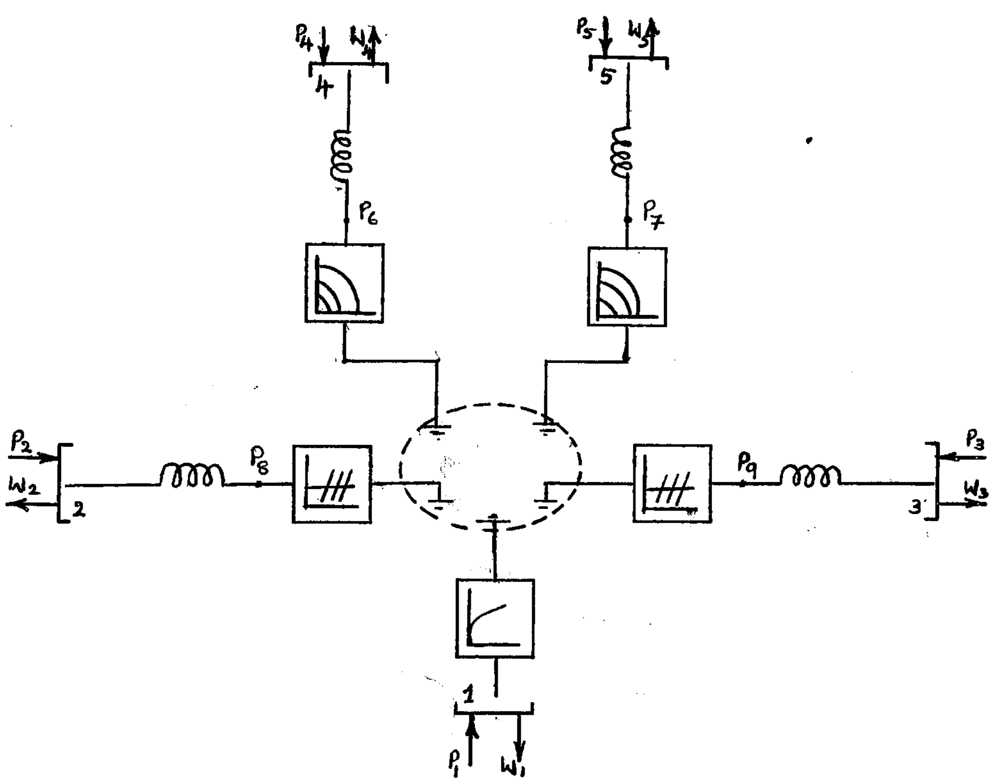
Port	Variable 1	Variable 2	Variable 3
1	P_1	W_1	–
2	P_2/P_1	$(P_2+P_3)/P_1$	W_2/W_1
3	P_3/P_1	$(P_2+P_3)/P_1$	W_3/W_1
4	W_4/W_1	$(P_2-P_3)/P_1$	P_4/P_1
5	W_5/W_1	$(P_2-P_3)/P_1$	P_5/P_1

A.9.4. Proportional Amplifier (Dynamic)

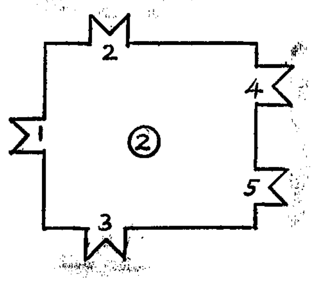
FCAP Name: PRPAMP (2)

Type Number: 2

Component Diagram:



FCAP Component Symbol



Component Equations

Assumptions

Time delay due to transportation is negligible.

Amplifier is well vented so that the static pressure in the interaction region is atmospheric.

The static characteristics may be normalized with respect to supply pressure and flow conditions.

The capacities of the control ports is negligible.

Vent dynamics are negligible.

Parameters

K_1 - Area of the control port (in^2)

K_2 - Length of the control port (in^2)

K_3 - Area of the output port (in^2)

K_4 - Length of the output port (in^2)

Dynamic Model

Definitions

$U_1 = P_1$, $U_2 = P_2$, $U_3 = P_3$, $U_4 = P_4$, $U_5 = P_5$

$Y_1 = W_1$, $Y_2 = W_2$, $Y_3 = W_3$, $X_1 = P_8$, $X_2 = P_9$, $X_3 = W_4$, $X_4 = W_5$

Model Equations

$$W_1 + \sum_i C_i P_1^{(i-1)} = 0$$

$$(P_2 - P_8) = \frac{K_2}{gK_1}\frac{dW_2}{dt}$$

$$W_2 + W_1 \sum_j \sum_i C_{2_{ij}} \left(\frac{P_8}{P_1}\right)^{(i-1)} \left(\frac{P_8 + P_9}{P_1}\right)^{(j-1)} = 0$$

$$(P_3 - P_9) = \frac{K_2}{gK_1}\frac{dW_3}{dt}$$

$$W_3 + W_1 \sum_j \sum C_{3_{ij}} \left(\frac{P_9}{P_1}\right)^{(i-1)} \left(\frac{P_8 + P_9}{P_1}\right)^{(j-1)} = 0$$

$$P_6 - P_1 \sum_j \sum_i C_{4_{ij}} \left(\frac{W_4}{W_1}\right)^{(i-1)} \left(\frac{P_8 - P_9}{P_1}\right)^{(j-1)} = 0$$

$$(P_6 - P_4) = \frac{K_4}{gK_3}\frac{dW_4}{dt}$$

$$P_7 - P_1 \sum_j \sum_i C_{5_{ij}} \left(\frac{W_5}{W_1}\right)^{(i-1)} \left(\frac{P_8 - P_9}{P_1}\right)^{(j-1)} = 0$$

$$(P_7 - P_5) = \frac{K_4}{gK_3}\frac{dW_5}{dt}$$

State Equations

$$\frac{dx_1}{dt} = \frac{gK_1U_1}{K_2Y_1}\left[(U_2 - x_1)D - (U_3 - x_2)B\right] / (AD - BC)$$

$$\frac{dx_2}{dt} = \frac{gK_1U_1}{K_2Y_1}\left[(U_3 - x_2)A - (U_2 - x_1)C\right] / (AD - BC)$$

$$\frac{dx_3}{dt} = \frac{g k_3}{K_4}\left[U_1 \sum_j \sum_i c_{4_{ij}} \left(\frac{x_3}{Y_1}\right)^{(i-1)} \left(\frac{x_1 - x_2}{U_1}\right)^{(j-1)} - U_4\right]$$

$$\frac{dx_4}{dt} = \frac{g k_3}{K_4}\left[U_1 \sum_j \sum_i c_{5_{ij}} \left(\frac{x_4}{Y_1}\right)^{(i-1)} \left(\frac{x_1 - x_2}{U_1}\right)^{(j-1)} - U_5\right]$$

where,

$$A = \sum_j \sum_i c_{2_{ij}} (i-1)\left(\frac{x_1}{U_1}\right)^{(i-2)}\left(\frac{x_1+x_2}{U_1}\right)^{(j-1)} + \sum_j \sum_i c_{2_{ij}} (j-1)\left(\frac{x_2}{U_1}\right)^{(i-1)}\left(\frac{x_1+x_2}{U_1}\right)^{(j-2)},$$

$$B = \sum_j \sum_i c_{2_{ij}} (j-1)\left(\frac{x_1}{U_1}\right)^{(i-1)}\left(\frac{x_1+x_2}{U_1}\right)^{(j-2)},$$

$$C = \sum_j \sum_i c_{3_{ij}} (j-1)\left(\frac{x_2}{U_1}\right)^{(i-1)}\left(\frac{x_1+x_2}{U_1}\right)^{(j-2)},$$

$$D = \sum_j \sum_i c_{3_{ij}} (i-1)\left(\frac{x_2}{U_1}\right)^{(i-2)}\left(\frac{x_1+x_2}{U_1}\right)^{(j-1)} + \sum_j \sum_i c_{3_{ij}} (j-1)\left(\frac{x_2}{U_1}\right)^{(i-1)}\left(\frac{x_1+x_2}{U_1}\right)^{(j-2)}.$$

Algebraic Equations

$$Z_1 = Y_1 + \sum_i c_i^* U_1^{(i-1)} = 0$$

$$Z_2 = Y_2 + Y_1 \sum_j \sum_i c_{2_{ij}} \left(\frac{x_1}{U_1}\right)^{(i-1)}\left(\frac{x_1+x_2}{U_1}\right)^{(j-1)} = 0$$

$$Z_3 = Y_3 + Y_1 \sum_j \sum_i c_{3_{ij}} \left(\frac{x_2}{U_1}\right)^{(i-1)}\left(\frac{x_1+x_2}{U_1}\right)^{(j-1)} = 0$$

Dependent Port Variables

$$V_1 = Y_1$$

$$V_2 = Y_2$$

$$V_3 = Y_3$$

Static Model

Definitions

$U_1 = P_1$, $U_2 = P_2$, $U_3 = P_3$, $U_4 = P_4$, $U_5 = P_5$

$V_1 = W_1$, $V_2 = W_2$, $V_3 = W_3$, $V_4 = W_4$, $V_5 = W_5$

Equations

$$G_1 = V_1 + \sum_j C_{1_i} U_1^{(i-1)} = 0$$

$$G_2 = V_2 + V_1 \sum_j \sum_i C_{2_{ij}} \left(\frac{U_2}{U_1}\right)^{(i-1)} \left(\frac{U_2 + U_3}{U_1}\right)^{(j-1)} = 0$$

$$G_3 = V_3 + V_1 \sum \sum C_{3_{ij}} \left(\frac{U_3}{U_1}\right)^{(i-1)} \left(\frac{U_2 + U_3}{U_1}\right)^{(j-1)} = 0$$

$$G_4 = U_4 - U_1 \sum_j \sum_i C_{4_{ij}} \left(\frac{V_4}{V_1}\right)^{(i-1)} \left(\frac{U_2 - U_3}{U_1}\right)^{(j-1)} = 0$$

$$G_5 = U_5 - U_1 \sum_j \sum_i C_{5_{ij}} \left(\frac{V_5}{V_1}\right)^{(i-1)} \left(\frac{U_2 - U_3}{U_1}\right)^{(j-1)} = 0$$

Characteristics to be read

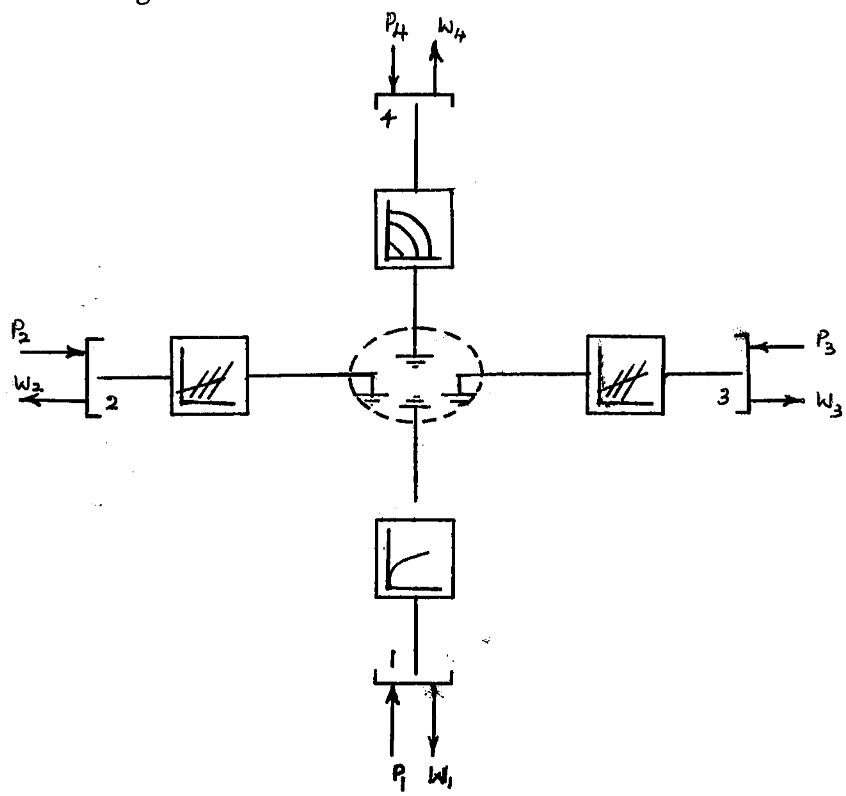
Same as proportional amplifier (algebraic)

A.9.5. Rectifier (Algebraic)

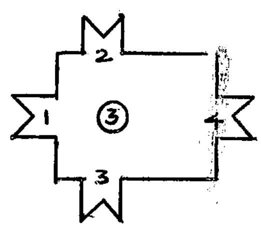
FCAP Name: RECTFR (1)

Type Number: 3

Component Diagram



FCAP Component Symbol



Component Equations

Assumptions

Same as in proportional amplifier (algebraic)

Parameters

None

Dynamic Model

Definitions

$$U_1 = P_1 ,\quad U_2 = P_2 ,\quad U_3 = P_3 ,\quad U_4 = P_4$$

$$Y_1 = W_1 ,\quad Y_2 = W_2 ,\quad Y_3 = W_3 ,\quad Y_4 = W_4$$

State Equations

None

Algebraic Equations

$$Z_1 = Y_1 + \sum_i c_{1_i} U_1^{(i-1)} = 0$$

$$Z_2 = Y_2 + Y_1 \sum_j \sum_i c_{2_{ij}} \left(\frac{U_2}{U_1}\right)^{(i-1)} \left(\frac{U_2+U_3}{U_1}\right)^{(j-1)} = 0$$

$$Z_3 = Y_3 + Y_1 \sum_j \sum_i c_{3_{ij}} \left(\frac{U_3}{U_1}\right)^{(i-1)} \left(\frac{U_2+U_3}{U_1}\right)^{(j-1)} = 0$$

$$Z_4 = U_4 - U_1 \sum_j \sum_i c_{4_{ij}} \left(\frac{V_4}{V_1}\right)^{(i-1)} \left(\frac{U_2-U_3}{U_1}\right)^{(j-1)} = 0$$

Dependent Port Variables

$$V_1 = Y_1$$

$$V_2 = Y_2$$

$$V_3 = Y_3$$

$$V_4 = Y_4$$

Static Model

Definitions

$$U_1 = P_1 , \quad U_2 = P_2 , \quad U_3 = P_3 , \quad U_4 = P_4$$

$$V_1 = W_1 , \quad V_2 = W_2 , \quad V_3 = W_3 , \quad V_4 = W_4$$

Equations

$$G_1 = V_1 + \sum_i C_{1_i} U_1^{(i-1)} = 0$$

$$G_2 = V_2 + V_1 \sum_j \sum_i C_{2_{ij}} \left(\frac{U_2}{U_1}\right)^{(i-1)} \left(\frac{U_2 + U_3}{U_1}\right)^{(j-1)} = 0$$

$$G_3 = V_3 + V_1 \sum_j \sum_i C_{3_{ij}} \left(\frac{U_3}{U_1}\right)^{(i-1)} \left(\frac{U_2 + U_3}{U_1}\right)^{(j-1)} = 0$$

$$G_4 = U_4 - U_1 \sum_j \sum_i C_{4_{ij}} \left(\frac{V_4}{V_1}\right)^{(i-1)} \left(\frac{U_2 - U_3}{U_1}\right)^{(j-1)} = 0$$

Characteristics to be Read

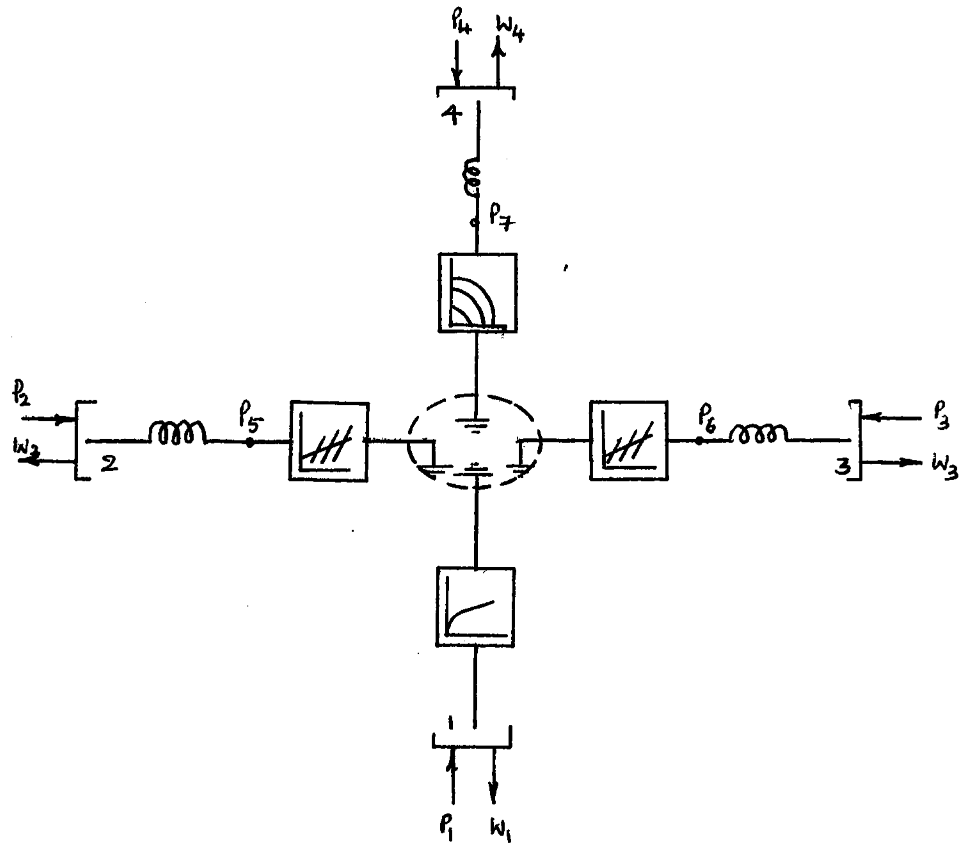
Port	Variable 1	Variable 2	Variable 3
1	P_1	W_1	—
2	P_2/P_1	$(P_2+P_3)/P_1$	W_2/W_1
3	P_3/P_1	$(P_2+P_3)/P_1$	W_3/W_1
4	W_4/W_1	$(P_2-P_3)/P_1$	P_4/P_1

A.9.6. <u>Rectifier (Dynamic)</u>

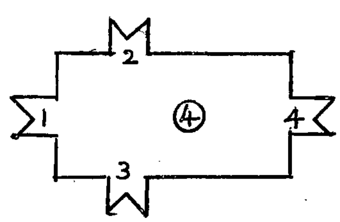
FCAP Name: RECTFR (2)

Type Number :4

Component Diagram



FCAP Component Symbol



Component Equations

Assumptions

Same as in proportional amplifier (dynamic)

Parameters

K_1 - Area of the control port (in^2)

K_2 - Length of the control port (in)

K_3 - Area of the output port (in^2)

K_4 - Length of the output port (in)

Dynamic Model

Definitions

$U_1 = P_1$, $U_2 = P_2$, $U_3 = P_3$, $U_4 = P_4$

$Y_1 = W_1$, $Y_2 = W_2$, $Y_3 = W_3$, $X_1 = P_5$, $X_2 = P_6$, $X_3 = W_3$

Model Equations

$$W_1 + \sum_i C_{1i} P_1^{(i-1)} = 0$$

$$(P_2 - P_5) = \frac{K_2}{gK_1} \frac{dW_2}{dt}$$

$$W_2 + W_1 \sum_j \sum_i C_{2_{ij}} \left(\frac{P_5}{P_1}\right)^{(i-1)} \left(\frac{P_5 + P_6}{P_1}\right)^{(j-1)} = 0$$

$$(P_3 - P_6) = \frac{K_2}{gK_1} \frac{dW_3}{dt}$$

$$W_3 + W_1 \sum_j \sum_i C_{3_{ij}} \left(\frac{P_6}{P_1}\right)^{(i-1)} \left(\frac{P_5 + P_6}{P_1}\right)^{(j-1)} = 0$$

$$P_7 - P_1 \sum_j \sum_i C_{4_{ij}} \left(\frac{W_4}{W_1}\right)^{(i-1)} \left(\frac{P_5 - P_6}{P_1}\right)^{(j-1)} = 0$$

$$(P_7 - P_4) = \frac{K_4}{gK_3} \frac{dW_4}{dt}$$

State Equations

$$\frac{dx_1}{dt} = \frac{g k_1 U_1}{k_2 Y_1}\Big[(U_2 - x_1) D - (U_3 - x_2) B\Big] \Big/ (AD - BC)$$

$$\frac{dx_2}{dt} = \frac{g k_1 U_1}{k_2 U_1}\Big[(U_3 - x_2) A - (U_2 - x_1) C\Big] \Big/ (AD - BC)$$

$$\frac{dx_3}{dt} = \frac{g k_3 U_1}{k_4}\left[\sum_j \sum_i C_{4_{ij}} \left(\frac{x_3}{Y_1}\right)^{(i-1)} \left(\frac{x_1 - x_2}{U_1}\right)^{(j-1)} - U_4\right]$$

where,

$$A = \sum_j \sum_i C_{2_{ij}} (i-1) \left(\frac{x_1}{U_1}\right)^{(i-2)} \left(\frac{x_1 + x_2}{U_1}\right)^{(j-1)} + \sum_j \sum_i C_{2_{ij}} (j-1) \left(\frac{x_1}{U_1}\right)^{(i-1)} \left(\frac{x_1 + x_2}{U_1}\right)^{(j-2)},$$

$$B = \sum_j \sum_i C_{2_{ij}} (j-1) \left(\frac{x_1}{U_1}\right)^{(i-1)} \left(\frac{x_1 + x_2}{U_1}\right)^{(j-2)},$$

$$C = \sum_j \sum_i C_{3_{ij}} (j-1) \left(\frac{x_2}{U_1}\right)^{(i-1)} \left(\frac{x_1 + x_2}{U_1}\right)^{(j-2)},$$

$$D = \sum_j \sum_i C_{3_{ij}} (i-1) \left(\frac{x_2}{U_1}\right)^{(i-2)} \left(\frac{x_1 + x_2}{U_1}\right)^{(j-1)} + \sum_j \sum_i C_{3_{ij}} (j-1) \left(\frac{x_2}{U_1}\right)^{(i-1)} \left(\frac{x_1 + x_2}{U_1}\right)^{(j-2)}$$

Algebraic Equations

$$Z_1 = Y_1 + \sum_i C_{1_i} U_1^{(i-1)} = 0$$

$$Z_2 = Y_2 + Y_1 \sum_j \sum_i C_{2_{ij}} \left(\frac{x_1}{U_1}\right)^{(i-1)} \left(\frac{x_1 + x_2}{U_1}\right)^{(j-1)} = 0$$

$$Z_3 = Y_3 + Y_1 \sum_j \sum_i C_{3_{ij}} \left(\frac{x_2}{U_1}\right)^{(i-1)} \left(\frac{x_1 + x_2}{U_1}\right)^{(j-1)} = 0$$

Dependent Port Variables

$$V_1 = Y_1$$

$$V_2 = Y_2$$

$V_3 = Y_3$

Static Model

Definitions

$U_1 = P_1$, $U_2 = P_2$, $U_3 = P_3$, $U_4 = P_4$

$V_1 = W_1$, $V_2 = W_2$, $V_3 = W_3$, $V_4 = W_4$

Equations

$$G_1 = V_1 + \sum_i C_{1_i} U_1^{(i-1)} = 0$$

$$G_2 = V_2 + V_1 \sum_j \sum_i C_{2_{ij}} \left(\frac{U_2}{U_1}\right)^{(i-1)} \left(\frac{U_2+U_3}{U_1}\right)^{(j-1)} = 0$$

$$G_3 = V_3 + V_1 \sum_j \sum_i C_{3_{ij}} \left(\frac{U_3}{U_1}\right)^{(i-1)} \left(\frac{U_2+U_3}{U_1}\right)^{(j-1)} = 0$$

$$G_4 = U_4 - U_1 \sum_j \sum_i C_{4_{ij}} \left(\frac{V_4}{V_1}\right)^{(i-1)} \left(\frac{U_2+U_3}{U_1}\right)^{(j-1)} = 0$$

Characteristics to be read

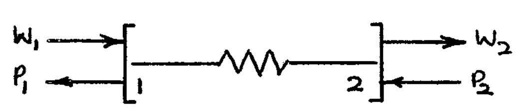
Same as in Rectifier (algebraic)

A.9.7. Resistor 1

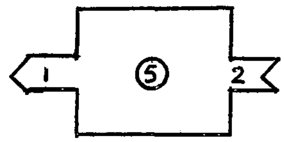
FCAP Name: RESTR1

Type Number: 5

Component Diagram:



FCAP Component Symbol:



Component Equations:

Assumptions

The fluid is incompressible.

Momentum effects are negligible

Parameters

K_1 - R, Resistance $\frac{\text{lbf}}{\text{in}^2} / \frac{\text{lbm}}{\text{sec}}$

Dynamic Model

Definitions

$U_1 = W_1$, $U_2 = P_2$, $Y_1 = P_1$, $Y_2 = W_2$

State Equations

None

Algebraic Equations

$Z_1 = Y_1 - U_2 - K_1 Y_2 = 0$

$Z_2 = U_1 + Y_2 = 0$

Dependent Port Variables

$V_1 = Y_1$

$V_2 = Y_2$

Static Model

Definitions

$U_1 = W_1$, $U_2 = P_2$, $V_1 = P_1$, $V_2 = W_2$

Equations

$G_1 = V_1 - U_2 - K_1 V_2 = 0$

$G_2 = U_1 + V_2 = 0$

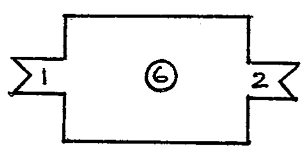
A.9.8. Resistor 2

FCAP Name: RESTR2

Type Number: 6

Component Diagram:

FCAP Symbol:



Component Equations:

Assumptions

Same as in Resistor 1

Parameters:

K_1 - R, Resistance $\frac{\text{lbf}}{\text{in}^2} / \frac{\text{lbm}}{\text{sec}}$

Dynamic Model

Definitions

$U_1 = P_1$, $U_2 = P_2$, $Y_1 = W_1$, $Y_2 = W_2$

State Equations

None

Algebraic Equations

$Z_1 = Y_1 + Y_2 = 0$

$Z_2 = U_1 + U_2 - K_1 Y_2 = 0$

Dependent Port Variables

$V_1 = Y_2$

$V_2 = Y_2$

Static Model

Definitions

$U_1 = P_1$, $U_2 = P_2$, $V_1 = W_1$, $V_2 = W_2$

Equations

$G_1 = V_1 + V_2 = 0$

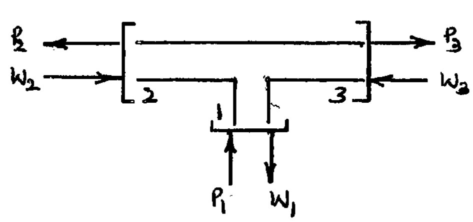
$G_2 = U_1 - U_2 - K_1 V_2 = 0$

A.9.9. Tee

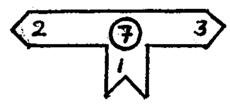
FCAP Name: TEE

Type Number: 7

Component Diagram:



FCAP Component Symbol



Component Equations:

Assumptions

There are no losses in the bends.

Parameters

None

Dynamic Model

Definitions

$U_1 = P_1, \quad U_2 = W_2, \quad U_3 = W_3, \quad Y_1 = W_1, \quad Y_2 = P_2$

State Equations

None

Algebraic Equations

$$Z_1 = Y_1 + U_2 + U_3 = 0$$

$$Z_2 = Y_2 - U_1 = 0$$

Dependent Port Variables

$$V_1 = Y_1$$

$$V_2 = Y_2$$

$$V_3 = Y_2$$

Static Model

Definitions

$U_1 = P_1, \quad U_2 = W_2, \quad U_3 = W_3$

$V_1 = W_1, \quad V_2 = P_2; \quad V_3 = P_3$

Equations

$G_1 = V_1 + U_2 + U_3 = 0$

$G_2 = V_2 - U_1 = 0$

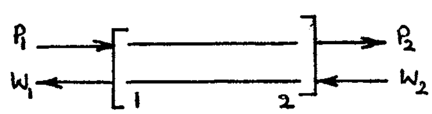
$G_3 = V_3 - U_1 = 0$

A.9.10. Transmission Line

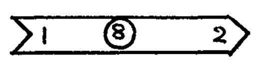
FCAP Name: LINE1

Type Number: 8

Component Diagram:



FCAP Symbol:



Component Equations

Assumptions

Fluid velocity at any point is much less than the velocity of sound in the fluid.

Perturbations in the density are negligible compared to the average density.

Temperature effects are negligible.

Fluid viscosity is spatially independent.

Flow field is axi-symmetric.

Pressure levels are small.

Parameters

K_1 - Diameter (inch)

K_2 - Length (inch)

K_3 - Speed of sound (in/sec)

K_4 - Density (lbf-sec^2/in^4

K_5 - Viscosity (lbf-sec/in^2)

Dynamic Model

Definitions

$U_1 = P_1$, $U_2 = W_2$, $V_1 = W_1$, $V_2 = P_2$, $x_1 = P_2$, $x_4 = W_1$

Model Equations

$$\left[T_1T_2D^3 + (T_1+T_2)D^2 + \left(T_3 + \frac{32T_e^2}{K\pi^2}\right)D + 1\right]P_2 =$$

$$(1+T_3D)P_1 - \frac{g\rho 8 T_e Z_0}{K}\left(1 + (T_1+T_2)D + T_1T_2D^2\right)W_2$$

$$\left[T_1T_2D^3 + (T_1+T_2)D^2 + \left(T_3 + \frac{32T_e^2}{K\pi^2}\right)D + 1\right]W_1 =$$

$$\frac{g\rho T_e}{Z_0}\left(D + T_3D^2\right)P_1 + (1+T_3D)W_2$$

where,

$K = K_1^2 K_4 / 4K_5$

$T_1 = K/5.78$

$T_2 = K/56.6$

$T_3 = K/40.9$

$T_e = K_2/K_3$

$Z_0 = 4K_4K_3/\pi K_1^2$

State Equations

$$\frac{dx_1}{dt} = x_2 - K_4 g Z_0 \pi^2 U_2 / 4T_e$$

$$\frac{dx_2}{dt} = x_3 + \frac{T_3 k \pi^2}{32 T_e^2 T_1 T_2} U_1$$

$$\frac{dx_3}{dt} = -\left(\frac{T_1+T_2}{T_1 T_2}\right) x_3 - \left(\frac{1 + T_3 k \pi^2 / 32 T_e^2}{T_1 T_2}\right) x_2 - k\pi^2/(32 T_e^2 T_1 T_2)\cdot x_1$$

$$+ \frac{\pi^2 k}{32 T_e^2 T_1 T_2} U_1 - \frac{T_3 k \pi^2 (T_1+T_2)}{32 T_e^2 T_1^2 T_2^2} U_1 + \frac{K_4 g \pi^4 z_0 k T_3}{128 T_e^3 T_1 T_2} U_2$$

$$\frac{dx_4}{dt} = x_5 + \frac{K_4 g \pi^2 k T_3}{32 T_e T_1 T_2 z_0} U_1$$

$$\frac{dx_5}{dt} = x_6 + \frac{K_4 g (T_1 T_2 - T_2 T_3 - T_3 T_1) \pi^2 k}{32 T_e T_1^2 T_2^2 z_0} U_1 + \frac{\pi^2 k T_3}{32 T_e^2 T_1 T_2} U_2$$

$$\frac{dx_6}{dt} = -\left(\frac{T_1+T_2}{T_1 T_2}\right) x_6 - \left(\frac{T_1 T_2 + \pi^2 k T_3}{32 T_e^2 T_1 T_2}\right) x_5 - \frac{\pi^2 k}{32 T_e^2 T_1 T_2} x_4$$

$$- \frac{K_4 g}{T_1 T_2}\left\{\frac{(T_1 T_2 - T_2 T_3 - T_3 T_1)\pi^2 k (T_1+T_2)}{32 T_e T_1^2 T_2^2 z_0} + \frac{\pi^2 k T_3}{32 T_e T_1 T_2 z_0}\left(1 + \frac{\pi^2 k T_3}{32 T_e^2}\right)\right\} U_1$$

$$- \frac{1}{T_1 T_2}\left[\frac{\pi^2 k T_3}{32 T_e^2 T_1 T_2}(T_1+T_2) - \frac{\pi^2 k}{32 T_e^2}\right] U_2$$

Initial Conditions

$$x_1(0) = U_1(0) - 8 T_e z_0 K_4 g\, U_2(0) / k$$

$$x_2(0) = K_4 g \pi^2 z_0 U_2(0) / 4 T_e$$

$$x_3(0) = - T_3 k \pi^2 U_1(0) / (32 T_e^2 T_1 T_2)$$

$$x_4(0) = U_2(0)$$

$$x_5(0) = K_4 g \pi^2 k T_3 U_1(0) / (32 T_e T_1 T_2 z_0)$$

$$x_6(0) = - K_4 g (T_1 T_2 - T_2 T_3 - T_3 T_1) \pi^2 k\, U_1(0) / (32 T_e T_1^2 T_2^2 z_0)$$

$$- \pi^2 k T_3 U_2(0) / (32 T_e^2 T_1 T_2)$$

Algebraic Equations

None

Dependent Port Variables

$Y_1 = X_4$

$Y_2 = X_1$

Static Model

Definitions

$U_1 = P_1$, $U_2 = W_2$, $Y_1 = W_1$, $Y_2 = P_2$

Equations

$G_1 = Y_1 + U_2 = 0$

$G_2 = Y_2 + U_1 - 8 T_e Z_o K_4 g U_2 / K$

A.10. Adding Components to the FCAP Component Library

The four step procedure described below enables new components to be added to the component library on a permanent basis.

1. Prepare the component equation for the computer model,
2. Write the component subroutine,
3. Modify the FCAP assays,
4. Modify the program statements.

A.10.1. Preparing the Component Equations for the Computer Model

The preparation of the component equations for a computer model consists of the following five major steps.

a) Assign the component a unique name which consists of six letters or less.

b) Assign the component a type number. This type number must be followed consecutively, i.e., the type number of the new component must be one higher than the highest type number in the FCAP component library.

c) Draw the standard component diagram. Assign consecutive numbers to ports starting with 1. There may be a maximum of five ports. Assign the independent and dependent variables at each port.

d) Develop a unique FCAP component symbol. Indicate each port type with the proper FCAP port symbol shown in Table IX. Write the type number of the component in a circle within the component symbol.

e) Write the component equations using basic physical principles or polynomials (for the tabulated data). Follow the six steps below in writing the component equations.

   1. State the assumptions made in deriving the component equations,

   2. List the parameters required for the component model. Denote the parameters by K_i, $i = 1, \ldots, NP$; where NP is the number of parameters. If the dynamic model of the component involves more parameters than the static model, then list the parameters of the static model first. The parameters may be used to initialize any dependent port variable. This may be done

especially when the programmer wishes the user to supply guess values of dependent variables for the Newton-Raphson method.

3. If an analytic model exists, derive the component equations from basic principles. Define flow into a port as negative and flow out of a port as positive.
4. Define the independent port variables as U's, dependent port variables as V's, algebraic variables as Y's, and static variables as X's. The subscripts of the U's and V's must be identical to their port numbers. Write the dynamic model of the component using the component equations in the following FCAP formulation:

$$\underline{\dot{X}} = \underline{F}(\underline{X}, \underline{Y}, \underline{U}, t),$$

$$\underline{Z} = \underline{Z}(\underline{X}, \underline{Y}, \underline{U}, t) = 0$$

Dependent port variables

$$V = V(Y)$$

or $V = V(X, t)$

where

X is the state variable vector,

Y is the algebraic variable vector,

U is the independent variable at the ports,

V is the dependent variable at the ports,

F is the functional form of the state derivatives,

Z is the functional form of algebraic equations in the dynamic model.

In the dynamic model, a dependent port variable V may have one of two forms: 1) equal to one of the five algebraic variables (Y), or 2) a function of states and/or time. For example,

$$V(1) = Y(2)$$

$$V(2) = Y(4)$$

$$V(3) = X(1) + X(2) + \mathrm{Sin}(t)$$

If the dependent variable is equal to an algebraic variable, then the port to which it belongs is termed an algebraic port.

There can be only a maximum of five algebraic variables (Y's) in a dynamic model, and corresponding to each variable there should be a Z equation. The algebraic equations must be separated into coupled sets of simultaneous equations, depending on the coupling of Y variables. Each set must be numbered consecutively starting from 1. For example, let

$$Z(1) = Z_1(\ldots, Y(1), Y(3), \ldots),$$

$$Z(2) = Z_2(\ldots, Y(2), Y(4), \ldots),$$

$$Z(3) = Z_3(\ldots, Y(3), Y(5), \ldots),$$

$$Z(4) = Z_4(\ldots, Y(4), \ldots),$$

$$Z(5) = Z_5(\ldots, Y(5), \ldots)$$

In the above example, Z(1), Z(3), and Z(5) form one set, and Z(2) and Z(4) form another set. Hence, Z(1), Z(3), and Z(5) are assigned Set 1 and Z(2) and Z(4) are assigned Set 2.

5. Define the independent port variables as U's and the dependent port variables as V's. The subscript of the U and V must be identical to that of the corresponding

port number. The static model of the component may be either an analytic model or it may be in the form of tabulated data.

a. Analytic model: Equate all the state derivatives in the component dynamic model equations to zero.

b. Tabulated data: Express the tabulated data in the form of polynomials in as many variables (one or two) as is appropriate for the particular port. The polynomial representation must be of the form

$$Q = \sum_{i=1}^{NX+1} C_i R^{(i-1)} \qquad \text{, for one variable,}$$

or

$$Q = \sum_{j=1}^{NY+1} \sum_{i=1}^{NX+1} C_{ij} R^{(i-1)} S^{(j-1)}$$

for two variables. In the first equation, designate R as variable 1, and Q as variable 2; in the second equation, designate R as variable 1, S as variable 2, and Q as variable 3. In the above equations, Q, R, and S may be either the independent or dependent port variables, or a function of both. For example, in the case of the static characteristic of the control port of a proportional amplifier,

$$Q = \frac{U_2}{U_1} \quad , \quad R = \left(\frac{U_2 + U_3}{U_1}\right) \quad , \quad S = \frac{V_2}{V_1}$$

The earlier mentioned sign convention must be used for flow.

Write the static model of the component in the form:

$$G = G(U, V) = 0$$

where, G is the functional form of the static model (a maximum of 5 equations), the U's are independent port variables, and the V's are dependent port variables.

6. If the static model uses tabulated data, write a table showing the port number, number of variables required and what the variables 1, 2 and 3 are for that particular port.

A.10.2. <u>Writing the Component Subroutine</u>

Instruction for writing the component subroutine are given in the computer listing shown in Figure 29.

A.10.3. <u>Modifying the FCAP Arrays</u>

There are 11 arrays in FCAP which store permanently the information about each type of component. The first ten arrays are:

1. ITAB
2. NPARS
3. NPARED
4. ISET
5. INDIFF
6. IDFOUT
7. IMPLCT
8. ISTPRT

```
C      EXAMPLE COMPONENT SUBROUTINE FOR THE COMPONENT XXXXXX
C
C      SUBROUTINE XXXXXX
C
C      IMPLICIT INTEGER*2(I-N)
C      INTEGER*4 NR,NW
C      COMMON/ALL/ NITER,KCLASS,IER,INDEXB,ISECTN,IPOINT,NORDER(20),ICLAS
C     1S(20),ISORT1(100),ISORT2(100),NPOLX(20,5),NPOLY(20,5),NPN(20,5),IN
C     2DCOM(20,5),IFNP(20,5),NPORT(20,5),IB(20,5),ITAB(11,12),DVDX(20,5),
C     3NCALL,KSET,IUVX(20,5),PARAM(20,10),ALGVAR(20,5),      B(100),VARDE
C     4P(20,5),VARIND(20,5),CONST(20,5,36),A(10000)
C      COMMON/DYNAM/NWRITE,NDYN,TSTART,TFINAL,DT,T,LDC(20),NSTATE(20),
C     1STATE(20,10),DSTATE(20,10)
C      COMMON/MNWTON/LLSET,NDIMB,ISAVE,KSTART,KEND,ICNTRL,NNSET(25),NEQN(
C     125),LORDNM(25),LORDST(25),IALG1(125),IALG2(125),IBALG(20,5),IFALG(
C     220,5)
C      COMMON/IO/ NR,NW
C
C
C      GO TO (100,200,300,400,500,500,500),ISECTN
C
C
C      ISECTN = 1.. INITIALIZE CONSTANTS AND EXPECTED VALUE FOR PORT VAR.
C      ISECTN = 2.. TRANSFER THE VALUES OF ALGEBRAIC VARIABLES (Y'S) TO
C                   DEPENDENT PORT VARIABLES (V'S).
C      ISECTN = 3.. TRANFER THE INITIAL CONDITIONS OF THE STATES
C      ISECTN = 4.. EVALUATE THE DEPENDENT PORT VARIABLES WHICH ARE
C                   FUNCTIONS OF STATES AND / OR TIME
C      ISECTN = 5.. EVALUATE A MATRIX (DZ/DY), AND B VECTOR (-Z) FOR
C                   ALGEBRAIC EQUATIONS
C      ISECTN = 6.. CALCULATE THE DERIVATIVES OF THE STATES
C      ISECTN = 7.. EVALUATE A MATRIX (DG/DV), AND B VECTOR (-G) FOR
C                   STATIC MODELS.
C
C 100  CONTINUE
C
C      PLACE ISECTN = 1 CARDS HERE.
C
C      RETURN
C
C 200  CONTINUE
C
C      PLACE ISECTN = 2 CARDS HERE
C
C      RETURN
C
C 300  CONTINUE
C
C      PLACE ISECTN = 3 CARDS HERE
C
C      RETURN
C
C
```

FIGURE 29 INSTRUCTIONS FOR WRITING A COMPONENT SUBROUTINE FOR FCAP

```
C
C 400 CONTINUE
C
C     PLACE ISECTN = 4 CARDS HERE
C
C     RETURN
C
C 500 CONTINUE
C
C     PLACE THE CARDS FOR UPDATING THE INDEPENDENT VARIABLES HERE
C
C     IF(ISECTN.EQ.6) GO TO 600
C     IF(ISECTN.EQ.7) GO TO 700
C
C     GO TO (510,520,530,540,550),KSET
C
C 510 CONTINUE
C
C     PLACE THE CARDS FOR ALGEBRAIC SET 1 HERE
C
C     RETURN
C
C 520 CONTINUE
C
C     PLACE THE CARDS FOR ALGEBRAIC SET 2 HERE
C
C     RETURN
C
C 530 CONTINUE
C
C     PLACE THE CARDS FOR ALGEBRAIC SET 3 HERE
C     RETURN
C
C 540 CONTINUE
C
C     PLACE THE CARDS FOR ALGEBRAIC SET 4 HERE
C
C     RETURN
C
C 550 CONTINUE
C
C     PLACE THE CARDS FOR ALGEBRAIC SET 5 HERE
C
C     RETURN
C
C 600 CONTINUE
C
C     PLACE ISECTN = 6 CARDS HERE
C
C     RETURN
C
C
      FIGURE 29 (CONTINUED)
C
```

```
C
C 700 CONTINUE
C
C      PLACE ISECTN = 7 CARDS HERE
C
C      RETURN
C      END
       MAIN                          1
C      COMPONENT MODEL
C
C
C      A STEP BY STEP PROCEDURE IS DESCRIBED IN THIS SECTION TO CONVERT
C      THE COMPONENT EQUATIONS INTO A COMPONENT SUBROUTINE
C
C      DEFINITION OF VARIABLES
C          VARDEP(NCALL,I) = V(I)
C          VARIND(NCALL,I) = J(I)
C          PARAM(NCALL,I) = K(I)
C          STATE(NCALL,I) = X(I)
C          DSTATE(NCALL,I) = DX(I)/DT
C          ALGVAR(NCALL,I) = Y(I)
C          A(IBALG(NCALL,I)+IFALG(NCALL,J)) = DZ(I)/DY(J)
C          A(IBALG(NCALL,I)+IFALG(NPORT(NCALL,J),IUVX(NCALL,J))) =
C          DZ(I)/DU(J)
C          B(IBALG(NCALL,I)) = -Z(I)
C          A(IB(NCALL,I)+IFNP(NCALL,J)) = DG(I)/DV(J)
C          A(IB(NCALL,I)+IFNP(NPORT(NCALL,J),NPN(NCALL,J))) = DG(I)/DJ(J)
C          B(IB(NCALL,I) = -G(I)
C
C      THERE ARE SEVEN SECTIONS IN EACH COMPONENT SUBROUTINE. THE ENTRY
C      INTO EACH SECTION IS CONTROLLED BY AN INTEGER "ISECTN" WHICH IS
C      THE ARGUMENT OF A COMPUTED GO TO.
C
C      GO TO (100,200,300,400,500,500,500),ISECTN
C
C      ISECTN = 1
C      THIS SECTION IS EXECUTED ONLY ONCE PER RUN,AND MAY BE USED FOR:
C      (1) INITIAL CALCULATION OF CONSTANTS,LIKE AREAS,VOLUMES,ETC.
C      (2) THE ALGEBRAIC EQUATIONS IN THE DYNAMIC AND STATIC MODELS ARE
C          SOLVED BY NEWTON-RAPHSON METHOD. IF INITIAL ESTIMATES ARE
C          NECESSARY TO INSURE CONVERGENCE,THI ESE VALUES MAY BE INCLUDED
C          AS PARAMETERS AND TRANSFERRED TO PROPER VARIABLES. FOR EXAMPLE
C          IF K(I) IS THE ESTIMATED VALUE FOR Y(J), AND K(L) IS THE
C          ESTIMATED VALUE FOR V(M),THEN STORE,
C              ALGVAR(NCALL,J) = PARAM(NCALL,I)
C              VARDEP(NCALL,M) = PARAM(NCALL,L)
C          THE DEFAULT VALUES ARE ZEROS.
C          NOTE : NEVER CHANGE THE VALUES OF ANY CONTROL VECTORS IN THIS
C          SECTION
C
C
C
       FIGURE 29 (CONTINUED)
C
```

```
C       ISECTN = 2
C
C       IF A DEPENDENT VARIABLE AT A PORT IS ALGEBRAIC, THEN IT MUST BE
C       EQUAL TO AN ALGEBRAIC VARIABLE.  TRANSFER THE VALUES OF ALL THE
C       ALGEBRAIC VARIABLES (Y'S) TO THE CORRESPONDING DEPENDENT PORT
C       VARIBLES (V'S) IN THIS SECTION. FOR EXAMPLE, IF V(I) = Y(J), THEN
C       STORE,
C               VARDEP(NCALL,I) = ALGVAR(NCALL,J)
C
C
C       ISECTN = 3
C
C       ENTRY INTO THIS SECTION IS MADE AFTER THE STEADY STATE OPERATION
C       OF THE CIRCUIT AT STARTING TIME HAS BEEN DETERMINED. INITIALIZE
C       THE STATE VARIABLE WHOSE INITIAL CONDITIONS ARE NONZERO BY USING
C       THE STEADY STATE VALUES OF THE PORT VARIABLES. FOR EXAMPLE,
C       IF X(1) = V(2), AND X(3) = U(1), THEN STORE,
C               STATE(NCALL,1) = VARDEP(NCALL,2)
C               STATE(NCALL,3) = VARIND(NCALL,1)
C
C       ISECTN = 4
C
C       CALCULATE THE DEPENDENT PORT VARIABLES WHICH ARE FUNCTIONS OF THE
C       STATE VARIABLES AND/OR TIME ALONE. FOR EXAMPLE, IF V(I) = X(J),
C       THEN STORE,
C               VARDEP(NCALL,I) = STATE(NCALL,J)
C
C
C       ISECTN = 5
C
C           UPDATE THE INDEPENDENT VARIABLES
C
C       FIGURE 27 (CONTINUED)
C       (1) FOR EACH U(I), IF NPORT(NCALL,I) IS NONZERO, THEN STORE,
C               VARIND(NCALL,<I>) = <+OR->VARDEP(NPORT(NCALL,<I>),
C               NPN(NCALL,<I>))
C           USE - SIGN IF U(I) IS A FLOW VARIABLE.
C           NOTE : INFORMATION TO BE FURNISHED BY THE PROGRAMMER IS WITHIN
C           THE SYMBOLS < AND >. OMIT THESE SYMBOLS FROM CODING.
C
C           FOR EACH ALGEBRAIC EQUATIONS SET IN THE DYNAMIC MODEL PERFORM
C           THE FOLLOWING TWO COMPUTATIONS
C
C       (2) FOR EACH INDEPENDENT VARIABLE, U(I), IN Z(J), CHECK AND STORE,
C       DZ(J)/DU(I)
C               IF(INDCOM(NCALL,<I>).NE.1) A(IBALG(NCALL,<J>) +
C                IFALG(NPORT(NCALL,<I>),IUVX(NCALL,<I>))) =
C                <+ OR -><DZ(J)/DU(I)>
C           NOTE : ORDINARY AND PARTIAL DERIVATIVES HAVE THE SAME NOTATION
C
C
C
        FIGURE 29 (CONTINUED)
C
```

```
C      (3) FOR EACH ALGEBRAIC VARIABLE, Y(I), IN Z(J), STORE,
C              A(IBALG(NCALL,<J>)+ IFALG(NCALL,<I>)) = <DZ(J)/DY(I)>
C          FOR EACH ALGEBRAIC VARIABLE, Y(I), STORE,
C              B(IBALG(NCALL,<I>)) = -<Z(I)>
C
C
C      ISECTN = 6
C
C      STORE EACH STATE DERIVATIVE DX(I)/DT AS
C              DSTATE(NCALL,<I>) = <DX(I)/DT>
C
C
C      ISECTN = 7
C
C      (1) FOR EACH INDEPENDENT VARIABLE U(I), EXPLICITLY IN G(J),
C          CHECK AND STORE,
C              IF(NPORT(NCALL,<I>).NE.0) A(IB(NCALL,<J>) +
C               IFNP(NPORT(NCALL,<I>),NPN(NCALL,<I>))) =
C               <+ OR -><DG(J)/DU(I)>
C          USE - SIGN IF U(I) IS A FLOW VARIABLE.
C
C          FOR EACH DEPENDENT VARIABLE V(I), EXPLICITLY IN G(J), STORE,
C              A(IB(NCALL,<J>) + IFNP(NCALL,<I>)) = <DG(J)/DV(I)>
C         FOR EACH DEPENDENT VARIABLE, V(I) STORE,
C              B(IB(NCALL,<I>)) = -<G(I)>
C
C      END
```

FIGURE 29 (CONTINUED)

9. LVXTAB

10. LUSET

The eleventh array, named IOPTYP, stores information about port types. The first ten arrays are at present dimensioned for eleven types of components. If more component types are present, this dimension has to be increased. The contents of each array are arranged according to the type number of the component. That is, row 1 of each array contains information about the component whose type number is 1, and row 2 contains information about component whose type number is 2, and so on. The entries in each array are explained below.

1. ITAB (11, 12): This table contains 12 columns. Each row contains information about the component whose type number is equal to the number of the particular row. The entries in the various columns are:

Column	Entry
1	Number of ports the component has
2	Type number of port 1
3	Type number of port 2
4	Type number of port 3
5	Type number of port 4
6	Type number of port 5
7	Number of variables to be read for port 1
8	Number of variables to be read for port 2
9	Number of variables to be read for port 3
10	Number of variables to be read for port 4
11	Number of variables to be read for port 5
12	Number of static variables the component has.

Enter a zero in any column which is not applicable. For example, if the component does not require tabulated data, columns 7 through 11 will have zeros.

2. <u>NPARS</u> (11): Enter the number of parameters required for the static model of the component in this table.

3. <u>NPARED</u> (11): Enter the number of parameters required for the dynamic model of the component in this table.

4. <u>ISET</u> (11): Enter the number of algebraic sets in the dynamic model of the component in this array.

5. <u>INDIFF</u> (11, 2): This array has two columns. An entry other than zero must be made in this table only if the component has two input ports, and if these two ports are either of type 1 or type 2. Enter the port number of one of the input ports in column 1, and the number of the other port in column 2.

6. <u>IDFOUT</u> (11, 2): An entry other than zero must be made in this table only if the component has two output ports, and if these ports are either of type 1 or 2. Enter the number of one of the output ports in column 1 and the number of the other in column 2.

7. <u>IMPLCT</u> (11, 5): This table contains five columns. Entries are made in this table for transferring the tabulated data of one port of the component to other similar ports. The entry in the i^{th} column, $i = 1, \ldots, 5$ is $\pm \ell_i$, where

$$\ell_i = 0 \text{ if the } i^{th} \text{ port does not require any transfer of data,}$$
$$= \text{port number whose tabulated data are to be transferred to the } i^{th} \text{ port.}$$

Use the + sign if the characteristics for the two ports are identical (as for an input port of a proportional amplifier), and the - sign if the characteristics are complementary (as for the output characteristics of a proportional amplifier).

8. ISTPRT (11, 5): This table has five columns: If the dependent variable at a port is algebraic, then it has to be equal to a particular Y, an algebraic variable in the dynamic model. The entries in this table indicate the algebraic equations set number. Specifically, the entry in each column is:

Column 1: If V_1 is a function a Y_i, i = 1, 2, . . ., 5, enter in column 1 the algebraic equation set number to which Y_i belongs.

Column 2: Similar information for V_2.

Column 3: Similar information for V_3.

Column 4: Similar information for V_4.

Column 5: Similar information for V_5.

9. LVXTAB (11, 5): This table has 5 columns. The contents of the table indicate which Y each V is a function of. The entry in each column is:

Column 1: Subscript of the Y which is equal to V_1.

Column 2: Subscript of the Y which is equal to V_2.

Column 3: Subscript of the Y which is equal to V_3.

Column 4: Subscript of the Y which is equal to V_4.

Column 5: Subscript of the Y which is equal to V_5.

If the port is not algebraic, enter a zero in that column.

10. LUSET (10, 5, 11): This table contains a 5 x 10 matrix for each type of component. The rows correspond to 5 sets of algebraic equations (if there are 5 algebraic equations and each is independent of the other, then there are 5 separate sets of equations) the component may have, and the first 5 columns contain information about U's and the last five

columns contain information about Y's. The entries in the columns of first row are:

Column	Entry
1-5	Enter a 1 in the i^{th} column, i, = 1, . . ., 5 if the first algebraic set is a function of U_i.
6-10	Enter a 1 in the j^{th} column, j = 6, . . ., 10 if the first algebraic set is a function of Y_i

The other rows contain similar entries for the corresponding sets, i.e., row 2 corresponds to set 2, etc. Enter a zero if the set does not contain the particular U_i or Y_i.

11. IOPTYP (4, 2): This table contains two columns. The number of rows depends upon the number of port types. Two rows have to be added for every new type of port. There are at present two basic types of ports (pressure-flow, force-displacement). For every pair of port variables, there are two combinations of port types. The two types are complementary to each other: One of them is arbitrarily called the "basic type" of port and the other, the "opposite type" port. Assign the two combinations of the new type of ports consecutive numbers starting one higher than the highest port type in the table. The entries in the i^{th} row correspond to the information about the i^{th} type port. Enter in column 1 the type number of the port complementary to the i^{th} type port. Enter in column 2 a 1 if the i^{th} port is a basic type port; otherwise, enter a 2. Assign a negative sign to the number in column 2 if the independent variable of the i^{th} type port has sign convention associated with it. Make entries for both the basic type port and its opposite type port in the table.

A.10.4 Modifying the Program Statements

For every new type of component added, some existing FORTRAN statements have to be changed and some new ones added in subroutines NESBN and CNTROL (ICNAME).

a) Changes in NESBN: The statement,

GO TO (40, 50, . . ., 140), JCLASS

should be changed so that when JCLASS is equal to the type number of the new component, the control branches to the call statement for the new component subroutine. A statement, GO TO 400, should follow the call statement to the component subroutine. Both the call statement and the GO TO 400 statement should be within the FORTRAN statement:

400 CONTINUE

b) Changes in CONTRL (ICMANE): The statement,

GO TO (10, 20, . . ., 110), JCLASS

should be changed so that when JCLASS is equal to the type number of the new component, the control branches to the call statement for the new component subroutine. A statement GO TO 1000 should follow the call statement to the component subroutine. Both the call statement and the GO TO 1000 statement should be before the FORTRAN statement

1000 CONTINUE.

APPENDIX B

A USER'S GUIDE TO THE NETOPT PROGRAM

B.1. Introduction

Fluidic circuits are generally nonlinear in operation. The performance of a circuit depends upon such circuit parameters as the supply pressures to active devices and the values of passive resistors. Normally, there are many combinations of these parameters which result in a given performance of the circuit and also satisfy the parameter and output constants. However, there is generally a unique combination of these parameters which extremizes the performance criterion and also meets the constraints. The determination of this unique set of parameters from all possible feasible solutions is dictated primarily by the cost and time available. Hand analysis and experimental methods are normally highly inefficient and costly.

This appendix contains user information for a NETwork OPTimization computer program which alleviates these problems. NETOPT is an efficient design tool which provides a computer-aided design approach to the static design of proportional fluidic circuits.

This appendix presents the user's guide to the NETOPT program. Section B.2 discusses the capabilities of the NETOPT program; Section B.3 presents the preparations necessary for a NETOPT run; Sections B.4 through B.7 contain the output and diagnostic information; Section B.8 presents the models for the components in the NETOPT library; and Section B.9 explains the method for adding component models to the library.

B.2. Capabilities and Features of NETOPT

NETOPT is a user-oriented computer program for the design of proportional fluidic circuits. NETOPT determines the optimum parameters of a fluidic circuit, which minimize the power consumed by the active amplifiers, while the static operation of the circuit meets some user specified constraints. Specifically, NETOPT determines a parameter vector $\underline{K}$ which minimizes the performance criterion.

$$f(\underline{U},\underline{V},\underline{K}) = \sum_{i=1}^{IACT} \left(U_{NACT(i)1}\right)\left(V_{NACT(i)1}\right)$$

subject to constraints,

$$a \leq g_k(\underline{U},\underline{V},\underline{K}) \leq b \quad , \quad k = 1, 2, \ldots, NG$$

$$g_i(\underline{U},\underline{V},\underline{K}) \geq 0 \quad , \quad i = NG+1, \ldots, M$$

$$h_j(\underline{U},\underline{V},\underline{K}) = 0 \quad , \quad j = M+1, \ldots, M+MZ$$

where,

U is the independent variable at a port,

V is the dependent variable at a port,

K is the parameter to be optimized,

g_k is the functional relations for the gains to be constrained (programmed in NETOPT),

a is the user-specified lower limit for the gain,

b is the user-specified upper limit for the gain,

g_i is the functional form of the user-specified inequality constraints,

h_j is the functional form of the user-supplied equality constraints, IACT is the number of active components in the circuit,

NACT is the vector containing the component numbers of the active components,

NG is the number of gain constraints,

M is the total number of inequality constraints,

MZ is the number of equality constraints.

The constraints g_i, $i = NG + 1, \ldots, M$ and h_j, $j = M + 1, \ldots, M + MZ$ are supplied by the user through a subroutine CNSTNT discussed in Section B.3.

Tables are provided for the user to enter data such as the circuit topology, component types, component static characteristics, parameters and other related information. These data are transferred easily from the tables to computer cards for use as input data to the program.

The static models for most common proportional fluidic components are permanently stored in the NETOPT Library. Two types of models are included: 1) analytical models which require only parameter values to be supplied as input data, and 2) phenomenological models which require measured static characteristics or data from manufacturer's catalogs as input data. A provision is made for the user to supply new or refined models on a permanent basis.

NETOPT checks the input data set for inconsistancies and automatically "assembles" the circuit based on the input circuit topology information. There is virtually no limitation on the circuit that can be analyzed.

NETOPT includes four minimization techniques: 1) the Newton-Raphson technique, 2) the modified Newton-Raphson technique, 3) the first order gradient technique, and 4) the modified Fletcher-Powell technique. The user can specify one of these methods through the input data.

B.3. Preparation for a NETOPT Run

Nine basic steps are required in the preparation of the input data for a NETOPT run:

1. Sketch a NETOPT block diagram,
2. Fill a Circuit Connection Table,
3. Fill a Port Data Table,
4. Fill an Equality Table,
5. Fill a Free Port Table,
6. Fill an Optimization Data Table,
7. Fill a Parameters Table,
8. Transfer the tables onto computer cards and arrange them in proper order,
9. Write a subroutine CNSTNT to introduce the constraints.

These steps are discussed below.

B.3.1. Sketching the Block Diagram

The sketching of the block diagram using NETOPT component symbols is identical to that discussed for the FCAP program in Section A.3 of Appendix A.

B.3.2. Circuit Connection Table

The Circuit Connection Table specifies interconnections between components in the circuit. The entries in this table are similar to those in the FCAP Circuit Connection Table (see Section A.3, Appendix A), with the exception that in this case the entries corresponding to state variables printing (columns 64-73) are ignored. It should also be noted that there is no Option Card at the beginning of the NETOPT Circuit Connection Table.

B.3.3. Port Data Table

The entries in this table are identical to those in the Port Data Table discussed in Section A.3.3, Appendix A.

B.3.4. Equality Table

The entries in this table are identical to those in the Equality Table discussed in Section A.3.4, Appendix A.

B.3.5. Free Port Table

The entries in this Table are identical to those in the Free Port Table discussed in Section A.3.5, Appendix A.

B.3.6. Optimization Data Table

The required entries in the Optimization Data Table (shown in Figure 30) are:

Row 1:

1. No. of Parameters: Total number of parameters to be optimized,
2. No. of Inequality Constraints: Number of inequality constraints introduced by the user,
3. No. of Equality Constraints: Number of equality constraints introduced by the user,
4. Method of Minimization: = 1 for Newton-Raphson method

NETOPT OPTIMIZATION DATA TABLE	No. OF OPTIMIZATION PARAMETERS	No. OF INEQUALITY CONSTRAINTS	No. OF EQUALITY CONSTRAINTS	METHOD OF MINIMIZATION
CARD COLUMN →	10-15	16-20	21-25	26-30
ROW 1				

PARAMETERS	1		2		3		4		5		6		7		8		9		10	
	COMP.	PARAM.	COMP.	PARAM.	COMP.	PARAM.	COMP.	PARAM.	COMP.	PARAM.	COMP.	PARAM.	COMP.	PARAM.	COMP.	PARAM.	COMP.	PARAM.	COMP.	PARAM.
CARD COL. →	9-10	11-12	13-14	15-16	17-18	19-20	21-22	23-24	25-26	27-28	29-30	31-32	33-34	35-36	37-38	39-40	41-42	43-44	45-46	47-48
ROW 2																				

GAIN CONSTRAINT	TYPE OF GAIN	INPUT AT		GAIN FROM		GAIN TO		UPPER LIMIT	LOWER LIMIT	INPUT VARIATION
		COMP.	PORT	COMP.	PORT	COMP.	PORT			
CARD COL. →	9	11-20	21-25	26-30	31-35	36-40	41-45	46-55	56-65	66-75
ROW 3										

Figure 30. Optimization Data Table for NETOPT

= 2 for modified Newton-Raphson technique

= 3 for first order gradient technique

= 4 for modified first order graident technique.

Row 2:

Data for the parameters to be optimized are specified here.

Component: Number of the component to which the parameter belongs.

Parameter: Number of the parameter that is to be optimized.

Row 3:

Furnished only if gain constraint is to be specified.

1. Type: = P or blank for pressure gain

   = F for flow gain

2. Input: Data for the component where the input signal is applied is furnished in these columns.

   Component: Number of the component at which the input signal is applied.

   Port: = 0 for differential input

   = Number of the port at which the input signal is applied

3. Gain From: Data for the component from whose input ports the gain is to be constrained are furnished in these columns. This data may be ignored if they are the same as the input data.

   Component: Number of the component from whose input ports the gain has to be calculated.

   Port: = 0 if gain is to be calculated from differential input ports

= Number of the port from which the gain is to be constrained.

4. Gain To: Data for the component up to whose output ports the gain is to be calculated are furnished in these columns.

Component: Number of the Component up to whose output ports the gain has to be calculated.

Port: = 0 if the gain is to be constrained for differential output ports

= Number of the port up to which the gain is to be constrained.

5. Gain High: Upper limit of gain

6. Gain Low: Lower limit of gain

7. Input Perturbation: The amount by which the input signal has to be varied when calculating the gain.

B.3.7. Parameter Table

The entries in this table are identical to those in the Parameter Table discussed in Section A.3.7, Appendix A.

B.3.8. Arrangement of Data Cards

Figure 31 shows the arrangement of data cards for a NETOPT run. The cards with names PORTDATA, EQUALITY, FREEPORT, OPTMDATA and PARAMTRS are placed at the beginning of the Port Data, the Equality Data, the Free Port Data, the Optimization Data and the Parameter data cards respectively. These eight letter names are keywords which transfer the control to various sections in the program, and hence should

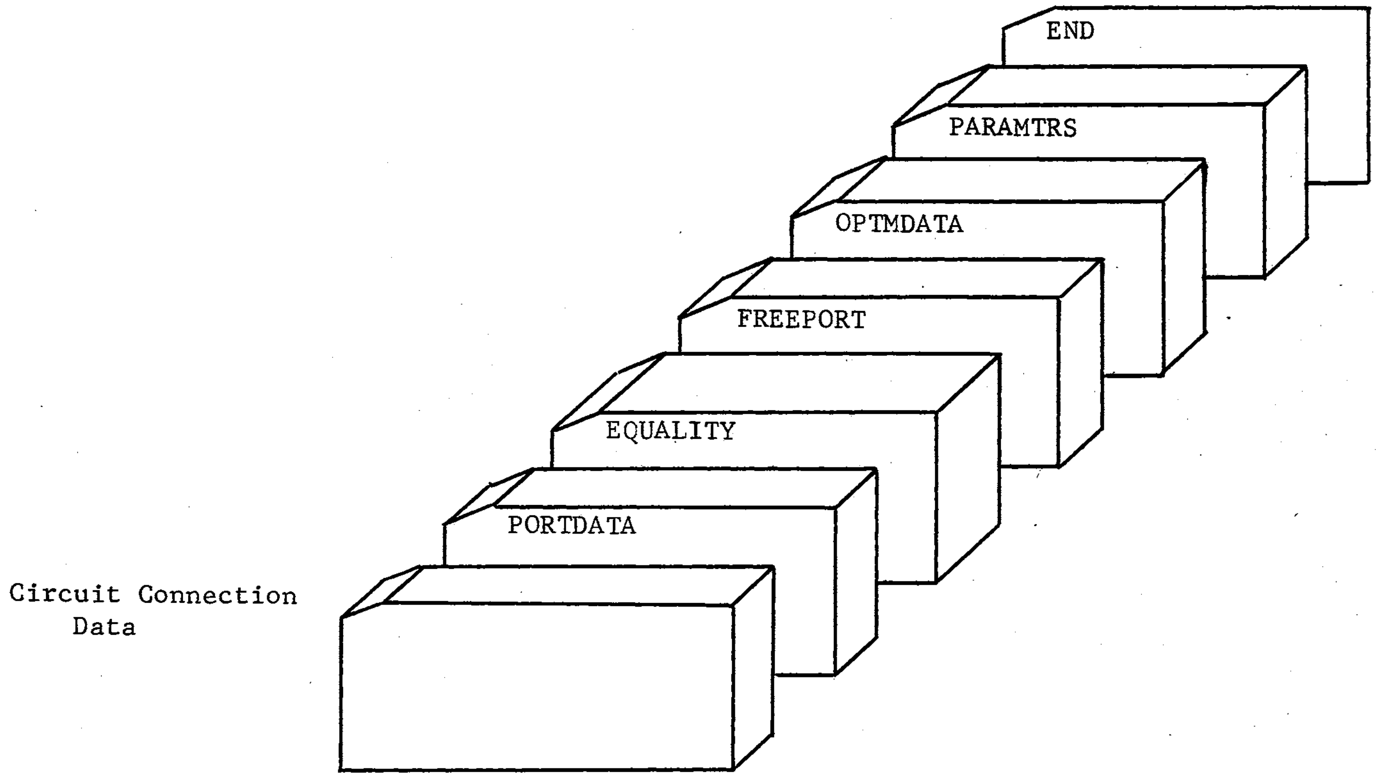


Figure 31. Arrangement of Data Cards for NETOPT

never be misspelled or changed. The keywords must be punched in the first eight columns of the card. An "END" card must be placed at the end of the data set (see Figure 29). The word END may be punched anywhere in columns 1 through 8.

B.3.9. Subroutine CNSTNT (I, VAL)

This subroutine must be supplied by the user to specify any additional constraints. The NETOPT program sets the value of I (1 to M + MZ - NG) and calls the CNSTNT subroutine for the value of the I^{th} user-supplied constraint. Hence, for a given value of I, this program must set VAL equal to the I^{th} constraint. The inequality constraints must be placed first followed by the equality constraints.

The following variable definitions must be used in writing the constraint equations:

K(L): PARAM (N, L),

U(L): VARIND (N, L),

V(L): VARDEP (N, L).

These variables are available in the COMMON block named ALL (given below) which must be included in this subroutine

```
   COMMON/ALL/ NITER, KCLASS, IER, INDEXB, IGYES, IPOINT, NORDER(20),
1  ICLASS (20), ISORT1 (100), ISORT2(100), NPOLX (20, 5), NPOLY (20, 5),
2  NPN(20, 5), INDCOM (20, 5), IFNP (20, 5), NPORT (20, 5), IB(20, 5),
3  ITAB(11, 12), INDP(20), NCALL, ICNTRL, PARAM(20, 10), B(100),
4  VARDEP(20, 5), VARIND(20, 5), CONST(20, 5, 36), A(10000).
```

B.4. NETOPT Output

The NETOPT program prints out the user-supplied input data and the results of the optimization problem. The input data are printed for user verification and permanent record. If errors are detected in the data, they are printed immediately below the appropriate data item.

NETOPT prints the values of the performance criterion, the penalty function, the constraints, and the parameters at the end of each subproblem. Port variables may be printed for debugging purposes by setting IDEBUG = 1 in the optimization data section. The port variables to be printed must be specified in the circuit connection data section. The debug option produces voluminous output. The user is discouraged from using this option unless it is absolutely essential.

B.5. Newton-Raphson Convergence

NETOPT uses the Newton-Raphson method to solve the simultaneous algebraic equations that represent the static behavior of the system. The maximum number of iterations of the Newton-Raphson method is presently limited to fifty.

If the solution to the algebraic equations does not converge within fifty iterations, an error message is printed, and the execution of the problem is continued. The probable reasons for the nonconvergence of the Newton-Raphson iteration are:

1. The number of iterations allowed may not be sufficient. The limit may be increased by increasing the value of the variable NITER in the main program,
2. If a new component model is being tried, errors in the equations for either the functions or the partial derivatives may cause

nonconvergence,

3. The initial estimates for all the algebraic variables default to zero. This may be a poor choice. Nonzero initial values for some port variables may insure convergence. These initial estimates may be introduced as parameters.

B.6. Matrix Singularity

NETOPT program uses the Gauss-Jordan elimination technique to solve simultaneous linear equations which result 1) during the least square method of curve or surface fitting for tabulated data, and 2) during the solution of system equations using the Newton-Raphson method. A singular matrix may result in either case.

The probable reason for a singular matrix to result during the curve or surface fitting may be due to the lack of sufficient number of points. At least (N + 1) points should be furnished for curve fitting, where N is the order of the polynomial; and (NX + 1) x (NY + 1) points should be furnished for surface fitting where NX and NY are the orders of the polynomials in variable 1 and variable 2 respectively.

A singular Jacobian matrix may result during the Newton-Raphson iteration due to a poor choice of initial estimates for the port variables. Nonzero initial values for some port variables may sometimes alleviate the problem.

B.7. Program Limitations

The main limitations of the NETOPT program are:

1. The control arrays are presently dimensioned to accomodate only 20 components in the circuit. The dimensions of the arrays can,

however, be increased to include more components,

2. The maximum number of design parameters is limited to ten per circuit. This can also be increased by increasing the dimensions of the proper arrays,
3. Provision has not been made for the temporary addition of components,
4. Input data must be furnished in fixed format.

B.8. NETOPT Component Library

NETOPT component library contains models for the same components as in the FCAP library. The model specifications are identical to those discussed in Section A.8, Appendix A. In the model specifications in Section A.8, only the static models are applicable since NETOPT optimizes the static operation of the circuit. The structure of the component subroutines in the NETOPT library is different from the structure of the component subroutines in the FCAP library due to the differences in the nature of the problems solved by the two programs. The structure of a NETOPT component subroutine is discussed in the next section.

B.9. Adding Components Models to the NETOPT Component Library

This section presents the procedure that enables new component models to be added to the NETOPT component library on a permanent basis. The addition procedure involves the following main steps:

1. Preparing the component equations for the computer model,
2. Writing the component subroutine,

3. Modifying the NETOPT arrays,

4. Modifying the program statements.

B.9.1. Preparing the Component Equations for the Computer Model

The preparation of the component equations for a computer model consists of the following five major steps:

1. Assign a unique name (six letters or less) to the component,
2. Assign the component a type number. This type number must be followed consecutively, i.e., the type number of the new component must be one higher than the highest type number in the NETOPT component library.
3. Draw the standard component diagram. Assign consecutive numbers to ports starting with 1. There may be a maximum of five ports. Assign the independent and dependent variables at each port.
4. Develop a unique NETOPT component symbol. Indicate each port type with the proper NETOPT port symbol shown in Table IX. Write the type number of the component in a circle within the component symbol.
5. Write the component equations using basic principles or polynomials (for tabulated data). Follow the five steps described below in writing the component equations.

   a) State the assumptions made in deriving the component equations.

   b) List the parameters required for the component model. Denote the parameters by K_i, i = 1. . ., NP; where NP is the number of parameters. If the component is an active amplifier, make the supply pressure the last parameter.

c) If an analytic model exists, write the component equations using the sign convention for flow. Flow into a port is negative and flow out of a port is positive.

d) Define the independent port variables as U's and the dependent port variables as V's. The subscripts of the U's and V's must be consistent with the corresponding port numbers. The static model of a component may be in one of the two forms. It may be either an analytic model, or it may be in the form of tabulated data. If the model is in the form of tabulated data, express it mathematically in the form of polynomials. (See the model of the proportional amplifier for illustration.) Write the static model of the component in the form:

$$G(U, V) = 0$$

where G is the functional form of the static model (a maximum of five equations),

U's are the independent port variables,

V's are the dependent port variables.

5. If the static model is in the form of tabulated data, write a table showing the port number, number of variables required and what the variables 1, 2 and 3 are for the particular port.

B.9.2. Writing the Component Subroutine

Instructions for writing the component subroutine are given in the computer listing shown in Figure 32.

B.9.3. Modifying the NETOPT Arrays

There are eight arrays in NETOPT which store permanently the information about each type of component. The first seven arrays are:

1. ITAB
2. NPARED
3. INDIFF
4. IDFOUT
5. IMPLCT
6. NPOPT
7. IACTAB

The eighth array, IOPTYP, stores information about port types. The first seven arrays are presently dimensional for eleven types of components. The contents of each array are arranged according to the type number of the component. That is, row 1 of each array contains information about the component whose type number is 1, and row 2 contains information about the component whose type number is 2, and so on. The contents of the arrays are explained below.

1. ITAB (11, 11): The ITAB table has 11 rows and 11 columns. The entries in the 11 columns of a row contain eleven or more pieces of information about the component whose type number is equal to the number of that particular row. The entries in the various columns are:

Column	Entry
1	The number of ports the component has
2	Type number of port 1
3	Type number of port 2

```
C      A STEP BY STEP PROCEDURE IS DESCRIBED IN THIS SECTION TO CONVERT
C      THE COMPONENT EQUATIONS INTO A COMPONENT SUBROUTINE
C
C
C      DEFINITION OF VARIABLES
C           VARDEP(NCALL,I) = V(I)
C           VARIND(NCALL,I) = U(I)
C           PARAM(NCALL,I) = K(I)
C           A(IB(NCALL,I)+IFNP(NCALL,J)) = DG(I)/DV(J)
C           A(IB(NCALL,I)+IFNP(NPORT(NCALL,J),NPN(NCALL,J))) = DG(I)/DJ(J)
C           B(IB(NCALL,I) = -G(I)
C
C
C      SUBROUTINE XXXXXX
C
C
C      COMMON/ALL/ NITER,KCLASS,IER,INDEXB,IGYES ,IPOINT,NORDER(20),ICLAS
C     1S(20),ISORT1(100),ISORT2(100),NPOLX(20,5),NPOLY(20,5),NPN(20,5),IN
C     2DCOM(20,5),IFNP(20,5),NPORT(20,5),IB(20,5),ITAB(11,12),INDP(20)   ,
C     3NCALL,ICNTRL,PARAM(20,10),B(100),VARDEP(20,5)
C     4        ,VARIND(20,5),CONST(20,5,36),A(10000)
C      COMMON/IO/ NR,NW
C
C
C      1. IF THIS IS AN ACTIVE COMPONENT, CHECK AND TRANSFER,
C
C                IF(INDP(NCALL) .EQ.1) VARIND(NCALL,<N>) = PARAM(NCALL,<L>)
C
C                WHERE, N IS THE SUPPLY PORT NUMBER,
C                        L IS THE SUBSCRIPT OF THE PARAMETER K DEFINED AS
C                         THE SUPPLY PRESSURE.
C
C           NOTE : INFORMATION TO BE FURNISHED BY THE PROGRAMMER IS WITHIN
C           THE SYMBOLS < AND >. OMIT THESE SYMBOLS FROM CODING.
C
C
C      2. UPDATE THE INDEPENDENT VARIABLES.
C
C         FOR EACH U(I), CHECK AND TRANSFER,
C
C                IF(NPORT(NCALL,<I>).NE.0) VARIND(NCALL,<I>) =
C                 <+ OR -> VARDEP(NPORT(NCALL,<I>),NPN(NCALL,<I>))
C
C          USE - SIGN IF U(I) IS A FLOW VARIABLE.
C
C      3.  FOR EACH INDEPENDENT VARIABLE U(I), EXPLICITLY IN G(J),
C          CHECK AND STORE,
C                IF(NPORT(NCALL,<I>).NE.0) A(IB(NCALL,<J>) +
C                 IFNP(NPORT(NCALL,<I>),NPN(NCALL,<I>))) =
C                 <+ OR -><DG(J)/DU(I)>
C
C          USE - SIGN IF U(I) IS A FLOW VARIABLE.
C          FOR EACH DEPENDENT VARIABLE V(I), EXPLICITLY IN G(J), STORE,
C                A(IB(NCALL,<J>) + IFNP(NCALL,<I>)) = <DG(J)/DV(I)>
C
C         FOR EACH DEPENDENT VARIABLE, V(I) STORE,
C                B(IB(NCALL,<I>)) = -<G(I)>
C
C      END
C
C
```

FIGURE 32 INSTRUCTIONS FOR WRITING A COMPONENT SUBROUTINE FOR NETOPT

Column	Entry
4	Type number of port 3
5	Type number of port 4
6	Type number of port 5
7	Number of variables to be read for port 1
8	Number of variables to be read for port 2
9	Number of variables to be read for port 3
10	Number of variables to be read for port 4
11	Number of variables to be read for port 5

Enter a zero in any column which is not applicable. For example, if the component does not require tabulated data, columns 7 through 11 will have zeros.

2. NPARED (11): Enter the number of parameters required for the static model of the component. If the component is an active device, enter one less than the number of parameters.

3. INDIFF (11, 2): This table has two columns. An entry other than zero must be made in this table only if the component has two input ports, and if these two ports are of either type 1 or 2. Enter the port number of one of the input ports in column 1, and the number of the other port in column 2.

4. IDFOUT (11, 2): An entry other than zero must be made in this table only if the component has two output ports, and if these ports are of either type 1 or 2. Enter the number of one of the output ports in column 1 and the number of the other in column 2.

5. IMPLCT (11, 5): This table contains five columns. Entries are made in this table for transferring the tabulated data of one port of a component to other similar ports. The entry in the i^{th} column is

$\pm \ell_i$, where i = 1, 2, . . ., 5 and

ℓ_i = 0 if the i^{th} port does not require any transfer of data,

= Port number whose tabulated data are to be transferred to the i^{th} port.

Use the + sign if the characteristics for the two ports are identical (as in the input ports of the proportional amplifier), and the - sign if the characteristics are complementary (as in the output characteristics of the proportional amplifier).

6. NPOPT (11): Enter the maximum number of parameters that may be considered for optimization for the component in question.

7. IACTAB(11): Enter the port number of the supply port, if the component is active; if not, enter a zero.

8. IOPTYP(4, 2): This table contains two columns. The number of rows depends upon the number of port types. Two rows have to be added for every new type of port. There are at present two basic types of ports (pressure-flow, force-displacement). For every pair of port variables, there are two combinations of port types. The two types are complementary to each other. One of them is arbitrarily called the "basic type" of port, and the other the "opposite type" port. Assign the two combinations of the new type of ports consecutive numbers starting one higher than the highest port type in the table. The entries in the i^{th} row correspond to the information about the i^{th} type port. Enter in column 1 the type number of the port complementary to the i^{th} type port. Enter in column 2 a 1 if the i^{th} port is a basic type port; otherwise, enter a 2. Assign a negative sign to the number in column 2 if the independent variable of the i^{th} type port has a sign convention associated with it (e.g., flow).

B.9.4. Modifying the Program Statements

The FORTRAN statement

GO TO (40, 50, . . ., 140), JCLASS

in the subroutine NESBN should be changed so that when JCLASS is equal to the type number of the new component, the control branches to the call statement of the new component. A FORTRAN statement, GO TO 400, should follow the call statement. Both the call statement and the "GO TO 400" statement should be before the statement: 400 CONTINUE.

VITA 9

Beegamudre Nagabhushana Rao Murali

Candidate for the Degree of

Doctor of Philosophy

Thesis: STATIC AND DYNAMIC ANALYSIS OF PROPORTIONAL FLUIDIC CIRCUITS

Major Field: Mechanical Engineering

Biographical:

Personal Data: Born in Tumkur, Mysore State, India, March 26, 1946, the son of Mr. and Mrs. B. Nagabhushana Rao.

Education: Graduated from St. Philomena's High School, Mysore, India, in 1961; received Bachelor of Engineering degree in Mechanical Engineering from University of Mysore, Mysore, India, in 1967; received the Master of Engineering degree from Indian Institute of Science, Bangalore, India, in 1969; completed the requirements for the Doctor of Philosophy degree at Oklahoma State University in December, 1973.

Professional Experience: Technical Assistant, Department of Mechanical Engineering, Indian Institute of Science, Bangalore, 1969-1970; graduate research assistant, Oklahoma State University, 1970-73.