DEVELOPMENT OF A UNIFIED APPROACH TO THE SIMULATION OF STATIC AND DYNAMIC BEHAVIOR OF LARGE MOBILE HYDRAULIC SYSTEMS

By

SESHADRI KALYAN RAM IYENGAR

Bachelor of Engineering
University of Jabalpur
Jabalpur, India
1963

Master of Science
Oklahoma State University
Stillwater, Oklahoma
1972

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
May, 1977

DEVELOPMENT OF A UNIFIED APPROACH TO THE

SIMULATION OF STATIC AND DYNAMIC

BEHAVIOR OF LARGE MOBILE

HYDRAULIC SYSTEMS

Thesis Approved:

E C Fitch
Thesis Adviser

K. N. Reid

Kotlarski

James R. Rowland

Norman N Durham
Dean of the Graduate College

PREFACE

This report is part of a continuing effort to mathematically model, simulate, and qualitatively appraise fluid power systems. The study was aimed at developing techniques for time domain simulation and eigenanalysis of large mobile hydraulic systems. By starting from the premise that mathematical models for components are available in the form of suitable equations, attention has been focused on the problem of synthesizing large system models in the time domain, from subsystem models, in a form suitable for digital simulation and eigenanalysis.

I wish to express my sincere appreciation and thanks to the following people and organizations who have encouraged and supported my educational endeavors:

Dr. E. C. Fitch, who served as my major advisor and graduate committee chairman. His counsel, encouragement, patience, and friendship have been major contributing factors not only towards completion of this work but also my professional advancement in the field of engineering.

Dr. K. N. Reid, for his constructive criticism and advice at various stages of the research effort.

The other members of my graduate committee, Dr. I. I. Kotlarski and Dr. J. R. Rowland, who were generous with their time, and whose guidance is gratefully acknowledged.

The sponsors of the Basic Fluid Power Research Program and the U. S. Army Mobility Equipment Research and Development Command, Ft. Belvoir, Virginia, whose sponsored research has furnished financial support during my graduate studies.

My colleagues at the Fluid Power Research Center, especially Dr. G. E. Maroney, for their encouragement, counsel, and criticism at various stages of the research; Joel Moore, David Scott, and Robert Sloan whose professional drafting assistance is acknowledged.

Dr. C. W. Gear of the University of Illinois, Champaign-Urbana, for furnishing the documentation on his computer program for implicit integration and answering sundry questions as to its use.

Miss Velda Davis for her excellent editing and preparation of the final manuscript of this thesis.

Lastly, my parents and brother who deserve special thanks for their encouragement and support.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

Mathematical Variables

x	State Vector*
y	Output vector*
u	Input Vector*
w	Aggregate subsystem input-output vector
X	Aggregate of subsystem state vectors (large system pseudo-state vector)
$\underline{X}$	Differential-Algebraic state vector
Y	Aggregate of subsystem output vectors (large system output vector)
U	Aggregate of subsystem input vectors
W	Aggregate of subsystem input-output vectors
n	Dimension of state vector*
m	Dimension of output vector*
r	Dimension of input vector*
V	External input to large system
f	Functional representation of differential equation*
g	Functional representation of algebraic equation*
H	Functional representation of interconnection between subsystems
N	Number of subsystems in large system
T	Matrix transpose operator

---

*Subscripted when they refer to a subsystem. Individual elements identified by a second subscript.

t	Time
σ	Stiffness ratio
λ	Eigenvalue

Physical Variables*

X	Spool displacement
a	Metering orifice area
P	Pressure
Q	Flow
C	Capacitance
I	Inertance
B	Resistance
W	Actuator load
Γ	Through variable*
Δ	Across variable**

Subscripts

First Subscript

S	Supply port
A	Work Port 'A'
B	Work Port 'B'
T	Tank Port

---

*Subscripted for identification.

**Superscript indicates subsystem and subscript identifies port.

Second Subscript

1	Actuator Number 1
2	Actuator Number 2

Variables in Gear's Algorithm

$\underline{y}$	Differential algebraic state vector
y	Scalar variable, element of $\underline{y}$
P	Dimension of $\underline{y}$
k	Order of integration
f	Functional representation of differential algebraic system
F_n	Discretized version of f, at the n'th time step
t	Time
n	Current step number
t_o	Initial time
t_f	Final time
h	Step size
$\left.\begin{matrix} a_i \\ b_i \end{matrix}\right\}$	Coefficients in multistep formula
$\left.\begin{matrix} \alpha_i \\ \beta_i \end{matrix}\right\}$	Coefficients in Gear's algorithm
J	Jacobian of F_n
PW	Computer representation of J.

CHAPTER I

INTRODUCTION

Fluid power systems in machinery can be classified broadly as "power" and "control" applications. The former category includes the majority of material handling equipment where hydraulics is used for "muscle" power--presses, compactors, hoists, cranes, and earthmoving equipment. The latter category includes mostly positioning and tracking equipment, where power levels, though high, are usually much less than in the first one. Examples of the second category abound in the aerospace and machine-tool industries. Hydraulics finds use in control applications because of the advantageously large power/weight and power/volume ratios offered by it in comparison to other implementations. If the key word for the first category of applications is efficiency, that for the second is precision. The line of demarcation between the two is hazy and it is entirely possible that in the near future it will become artificial.

In spite of the commonality of the basic mechanism of energy transfer in the two categories, the above mentioned difference of emphasis has resulted in two different methodologies of design. Apart from meeting force and velocity requirements, power systems are expected to exhibit good overall energy conversion efficiency. They are usually allowed substantial latitude in transient behavior, provided no premature failure of parts or operator-incompatibility is experienced. In the case of control systems, appraisal usually takes the form of tracking/positioning accuracy in the face of changing inputs, and disturbances. Historically, they have been treated as single-input single/multiple output linear systems (with the conventional extension to linearized nonlinear systems). Specifications for their appraisal almost always involve dynamic behavior first, with power/weight and power/volume ratios receiving secondary considerations, and efficiency, tertiary, at best. The variety of operations performed by control systems is somewhat less than power applications, so much so that a standard design procedure to cover a variety of applications can be laid out, as attested by the number of tutorial papers on the topic (1, 2, 3, 4). Similar procedures for power applications appear to be nonexistent, one possible reason being the latitude in circuit design and component selection allowed to the system designer.

Consequently, it is difficult to establish to what extent power systems, designed with the current state-of-the-art, deviate from the optimal. It would appear, however, that a systematization of the design procedure, based upon a thorough mathematical analysis of the operational tasks of a given machine would result, to some extent, in filling this void. Such a mathematical analysis would necessarily involve the development of a mathematical model which could be used to examine the behavior of relevant physical variables, as the machine is subject to specified operational tasks. One of the reasons why such analysis has not been widely used in the power systems area is that it involves the solving of a large set of coupled algebraic or differential-algebraic equations which usually exhibit pronounced nonlinearities. However, due to rapid advances in computer technology, the solving of algebraic and differential-algebraic equations is no longer the severe hurdle it once was, and improvements in both hardware and software promise to make accurate mathematical analysis an economically feasible tool for the design and appraisal of large classes of systems.

Power hydraulic system designers have, generally speaking, lagged behind control system designers in using machine computation facilities, partly due to a lack of incentive for accurate analysis, and partly due to a lack of appreciation of computer capabilities. As a result of wider dissemination of state-space theory, the transient analysis of various components, e.g., relief valves, pressure reducing valves, etc., has been attempted by component designers with varying degrees of success (5, 6, 7, 8). Entire systems, notably hydrostatic drives have been simulated (9). (Such systems straddle the line of demarcation between power and control applications.)

To contrast with systems described by differential-algebraic equations, those described by purely algebraic equations are called static systems, and simulations using such models as static simulation. The simulation of complete duty cycles of power systems, in which transients occupy only a small fraction of the cycle time and are to be ignored, requires static simulation. Since the equations describing the system behavior are usually nonlinear and often implicit, their solution is less straightforward than that of differential-algebraic equations. Only a few examples of static simulation are documented in the literature (10, 11, 12, 13, 14).

The main motivation for this research was that existing computer programs for dynamic systems simulation were considered inadequate for the class of systems under scrutiny, for one or more of the following reasons:

1. They use the generalized network approach, which is not geared towards accepting models of components or subsystems in the form of mathematical equations.
2. They cannot simulate systems described by purely algebraic equations.
3. They place restrictions, not based on physical considerations, on the manner in which subsystems can be interconnected.
4. They are either inefficient or incapable of simulating stiff systems, which are characterized by the presence of widely differing eigenvalues.

Objectives of Study

This dissertation addresses itself to the formulation of computerizable algorithms for analyzing mobile hydraulic systems, using lumped-parameter time domain models of their components. The interconnections between subsystems, which is called the topological structure of the system is considered describable by a set of algebraic equations. In this dissertation, the word 'topology' is intended to be construed only in the above sense and carries no overtones of meanings assigned to the word in mathematics. Even though the analysis of mobile hydraulic systems was the motivation for this effort, the mathematical treatment presented herein is general enough to be applicable to the entire class of systems whose subsystems can be described by differential-algebraic equations and whose topological structure can be described by algebraic equations.

A fundamental premise of the research effort is that the manner in which the mathematical models for subsystems are arrived at is immaterial insofar as the behavior of the total system is concerned. Starting from this premise, the development of a form of mathematical representation for the total system, which can explicitly display the subsystem models, as well as the system topological structure, was the first objective of the research. A system represented in the above manner is called a large system in the context of this thesis. The formulation of algorithms for time domain simulation and qualitative appraisal of large systems was the second objective.

Results of Study

One of the major conclusions of this research is that the order of a large system obtained by interconnecting subsystems may be less than the sum of the orders of the subsystems. An important consequence of this result is that explicit numerical integration methods are either incapable or inefficient in simulating large systems involving such order reduction.

The implicit form of representation, which is developed in this thesis, is shown to be suitable not only for representing large systems, but also for numerical integration without consideration of the order of the system. A computerized algorithm for qualitative appraisal of the dynamic behavior of large systems represented in the implicit form is also presented.

As an example, a mobile hydraulic system model is formulated in the implicit form and results of dynamic simulation as well as qualitative appraisal are presented. It is shown that digital simulation, which uses Gear's method of numerical integration of differential-algebraic equations, can be accelerated by switching models of subsystems at appropriate times determined by the values of the state and algebraic output variables.

Outline of Thesis

Chapter II gives examples to illustrate how order reduction can arise in the synthesis of large system models using subsystem models. It also discusses the limitations of explicit integration methods, which form the backbone of the vast majority of dynamic system simulation software. In Chapter III a new approach for modeling large systems is discussed in terms of a canonical representation for subsystems and the mathematical implications of physical interconnection between subsystems. In Chapter IV the implicit form of representing large systems is developed and shown to be suitable for digital simulation using Gear's algorithm, as well as for qualitative appraisal. As an example of a large system, an open center mobile hydraulic system is analyzed in Chapter V. The final chapter summarizes the important conclusions of the research and presents recommendations for further investigations.

Appendix A postulates and proves the order reduction theorem, which asserts that in the type of systems under consideration, algebraic constraints on outputs of subsystems, arising due to their interconnection, leads to order reduction. Appendix B presents an algorithm for qualitative appraisal of the dynamic behavior of large systems, based upon eigenanalysis in a prescribed operating region. Gear's method of implicit integration of differential-algebraic equations is briefly reviewed in Appendix C, while Appendix D explains the function of key subprograms in the large scale system simulation program using selected FORTRAN listings. Numerical values of parameters used in the example system simulation are documented in Appendix E.

CHAPTER II

THE LIMITATIONS OF EXISTING SOFTWARE

The establishment of systems methodology has served to decouple the modeling process from the mathematical analysis needed to obtain behavioral information of a system. Ever since the realization that the basic phenomena responsible for the dynamic behavior of many fluid power systems could be adequately described by exactly the same general set of ordinary differential and algebraic equations as are used to describe passive electrical networks and mechanical systems, some systems analysts have stressed that one need only develop the methodology for combining models of the basic elements (i.e., resistances, capacitances, inertances, gyrators, sources, etc.), in order to be able to describe the behavior of a system of any complexity whatsoever (15, 16, 17, 18). This philosophy of dissecting a system to its basic elements will be for lack of better terminology, referred to as the generalized network approach.

Even though the general applicability of the state space approach to the modeling of general lumped parameter dynamic systems is recognized, significant theorems on existence of solutions, order of systems, etc., are still formulated in terms of 'cut-sets', 'trees', and 'forests', or node analysis, concepts carried over from electrical network theory and not intuitively appealing to fluid power engineers (19, 20, 21, 22, 23). Computer programs written specifically for the analysis of fluid power systems have eschewed the generalized network approach, but have imposed restrictions on the manner in which subsystem models may be interconnected (24, 25). An explanation of these restrictions, which can be traced to a fundamental premise of the analysis will be given in this chapter.

The fundamental premise of the current research, which was summarized in the first chapter, will be elaborated upon in the following section, so as to form the background for a discussion, with appropriate examples, of 'order reduction', and an explanation of why current simulation software is incapable of handling systems involving order reduction. The last part of this chapter includes a critical review of general purpose dynamic simulation software and software written specifically for hydraulic systems analysis. It is shown that the inadequacy of all presently known software is based on its reliance on the explicit state vector formulation for the system.

Fundamental Premise and Goals

The fundamental premise of this research is that mathematical models for individual subsystems or components, which describe their behavior in terms of energy port variables are available to the analyst. Once a mathematically adequate subsystem representation is available, details of the internal constructional features or other details of the hardware are irrelevant, insofar as the analysis of the static and dynamic behavior of the large system is concerned. Since the analysis is to be restricted to lumped parameter systems, the most important implication of this premise is that components or subsystems may be described completely and unambiguously by sets of differential-algebraic equations. The interconnection of two subsystems implies the equality of one or more physical variables corresponding to the energy ports which are connected.

The following goals were laid out for the formulation of the large system model:

1. The mathematical model for the large system should explicitly exhibit not only the subsystem models but also the topological structure of the large system.
2. No restrictions, apart from those arising due to physical considerations, are to be imposed on the interconnections between models of components. Equivalently, complete freedom is to be allowed in demarcating subsystem boundaries.
3. The system model should be amenable, with only a minimum of algebraic manipulation, for digital simulation as well as qualitative appraisal.
4. The models for individual subsystems should be completely independent of each other so that changes in or substitution of a subsystem model would have no impact on other subsystem models.

Many computer simulation packages for dynamic systems, which exhibit the modularity concept outlined as goal 1, above, fail to meet goal 2. The next two sections explore the reasons for this failure by first giving examples of 'order reduction' and next outlining the inadequacies of explicit numerical integration in handling systems involving order reduction.

Order Reduction

Two large systems will be used as examples to illustrate order reduction. The first one has been chosen primarily to indicate why order reduction is of little practical significance in using the generalized network approach to problem formulation. The second example system is formulated, first using linear subsystem models, and next using a nonlinear model for one subsystem. The objective in presenting these last two examples is to demonstrate that order reduction is not as evident when subsystem models are given as sets of differential-algebraic equations, and topological constraints are described as algebraic equations, as when the generalized network approach is used. The linear model for the second system is used to demonstrate that representation of the large system in the explicit vector differential form may require derivatives of the external inputs, while the nonlinear model is developed to show that explicit state vector representation may sometimes be impossible.

Example I

Consider two RC networks as shown in Figure 1. Using the notation shown in the figure, and assuming that current sources are the inputs to the subsystems, the following models can be derived:

$$\dot{e}_{12} = [0]e_{12} + \begin{bmatrix} \frac{1}{c_1} & -\frac{1}{c_1} \end{bmatrix} \begin{bmatrix} i_{11} \\ i_{12} \end{bmatrix} \qquad (2.1a)$$

Subsystem #1

$$e_{11} = [1]e_{12} + [R_1 \quad 0] \begin{bmatrix} i_{11} \\ i_{12} \end{bmatrix} \qquad (2.1b)$$

Subsystem #2

$$\dot{e}_{21} = [0]\, e_{21} + \begin{bmatrix} \frac{1}{c_2} & -\frac{1}{c_2} \end{bmatrix} \begin{bmatrix} i_{21} \\ i_{22} \end{bmatrix} \qquad (2.2a)$$

$$e_{22} = [1]\, e_{21} + \begin{bmatrix} 0 & R_2 \end{bmatrix} \begin{bmatrix} i_{21} \\ i_{22} \end{bmatrix} \qquad (2.2b)$$

The above equations are written in the canonical form:

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

$$y = Cx + Du$$

Each of the subsystems is of the first order.

If the subsystems are connected by the dotted lines as shown in the figure, so as to form the system, the topological constraints become as follows:

$$i_{12} = i_{21} \qquad (2.3a)$$

$$e_{12} = e_{21} \qquad (2.3b)$$

The model for the large system can be shown to be as follows:

$$\dot{e}_{12} = [0]\, e_{12} + \begin{bmatrix} \frac{1}{c_1 + c_2} & \frac{-1}{c_1 + c_2} \end{bmatrix} \begin{bmatrix} i_{11} \\ i_{22} \end{bmatrix} \qquad (2.4a)$$

$$\begin{bmatrix} e_{11} \\ e_{22} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} e_{12} + \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix} \begin{bmatrix} i_{11} \\ i_{22} \end{bmatrix} \qquad (2.4b)$$

Equation (2.4) indicates that the large system is of the first order, i.e., the order is not the sum of the orders of the two subsystems. It should be noted that the state space is reduced due to the constraints of the connection, and not due to any inherent characteristics of the subsystems. In the context of this thesis, the phenomenon whereby the order of a large system synthesized from subsystem models is

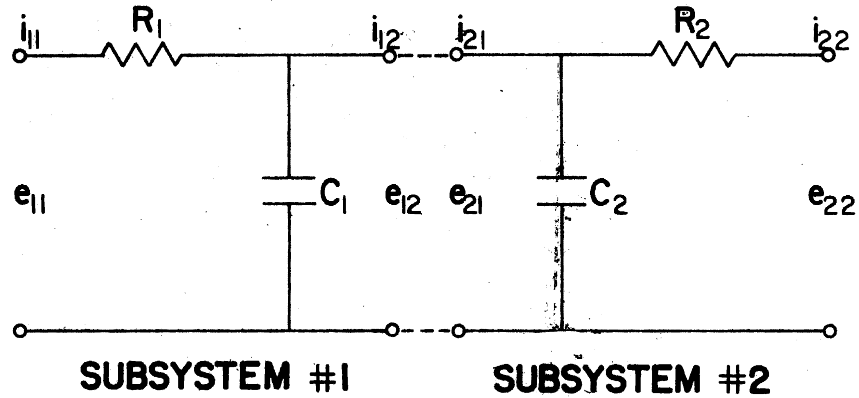


Figure 1. Example of a Large System Comprised of Two Subsystems and Involving an Algebraic State Constraint

less than the sum of the orders of the subsystems is termed order reduction.

In the above example, if order reduction were not evident from an inspection of the network, it could be inferred by algebraic manipulation of Equations (2.1), (2.2), and (2.3) so as to write the large system model in the canonical form for linear systems. Even though, in the case of linear systems, it is always possible to consolidate the subsystem models and present the model for the large system in the canonical form for linear systems, such consolidation is not always desirable, since it destroys the modularity of the large system model, and has to be repeated afresh whenever any subsystem parameters are changed.

If the model for the large system is retained in the form of Equations (2.1) through (2.3), it can be said that the large system has an algebraic state constraint due to the presence of Equation (2.3b). In that case the aggregate of the subsystem state vectors, i.e., $[e_{12} \quad e_{21}]$ can be defined as the 'pseudo' state vector for the large system (26). Thus, the terms 'order reduction' and 'algebraic state constraint' refer to the same phenomenon but have slightly different connotations.

Example 2

Figure 2 presents the circuit schematic for a hydraulic system, for which a lumped-parameter dynamic thermal model is desired. The heat exchanger and reservoir are to be modeled as first order systems, and the effects of all other components included in an equivalent heat source. Linear subsystem models, developed by Miller (27) can be used to write the model for the large system as follows:

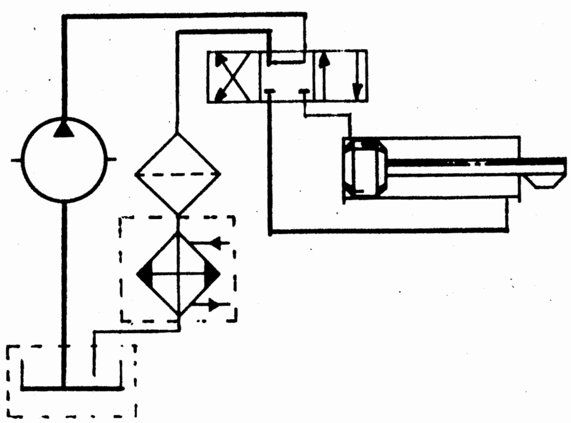


Figure 2 Hydraulic Circuit Schematic of a System Whose Thermal Model Involves Algebraic State Constraint

Heat Exchanger Subsystem

$$\dot{\Delta T}_H = a_{11}\ \Delta T_H + \begin{bmatrix} b_{11} & b_{12} \end{bmatrix} \begin{bmatrix} T_{fi_H} \\ T_{amb_H} \end{bmatrix} \tag{2.5a}$$

$$T_{fo_H} = 2\Delta T_H + 2T_{amb_H} - T_{fi_H} \tag{2.5b}$$

Reservoir Subsystem

$$\dot{\Delta T}_R = a_{21}\ \Delta T_R + \begin{bmatrix} b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} T_{fi_R} \\ T_{amb_R} \end{bmatrix} \tag{2.6a}$$

$$T_{fo_R} = 2\Delta T_R + 2T_{amb_R} - T_{fi_R} \tag{2.6b}$$

Topological Constraints

$$T_{fo_H} = T_{fi_R} \tag{2.7a}$$

$$T_{fi_H} = T_{fo_R} + \frac{\Sigma Hg}{\rho Q c_p} \tag{2.7b}$$

The notation used above, which is the same as that of Miller (27) is as follows:

T Temperature (subscripted)

ΔT Difference in temperature between bulk fluid inside a component and the relevant ambient temperature (subscripted)

ΣHg Rate of heat input to the system

ρQ Mass flow rate of fluid

c_p Specific heat of fluid

a_{11}

a_{21}

b_{11}

b_{12} Known paramters of system

b_{21}

b_{22}

fi Subscript denoting fluid inlet conditions

fo Subscript denoting fluid outlet conditions

amb Subscript denoting relevant ambient conditions

H Subscript for identifying Heat exchanger

R Subscript for identifying reservoir

Equations (2.5) and (2.6) are first order explicit state vector representations of the two subsystems, and ΔT_H and ΔT_R are their state vectors, respectively. By combining the algebraic output and topological constraint equations, it can be shown that

$$\Delta T_H = \Delta T_R + \frac{\Sigma Hg}{\rho Q c_p} + T_{amb_R} - T_{amb_H} \tag{2.8}$$

which is an algebraic constraint equation and, consequently, $[\Delta T_H \quad \Delta T_R]$ cannot be the state vector for the large system, i.e., arbitrary initial values cannot be assigned to ΔT_H and ΔT_R for purposes of digital simulation.

The consolidated explicit state vector first order model for the large system, which has been derived by Miller (27) has the form

$$\dot{T}_{fi_H} = \tau \, T_{fi_H} + f\left(T_{amb_R}, T_{amb_H}, \frac{\Sigma Hg}{\rho Q c_p}, \frac{\Sigma \dot{H}g}{\rho Q c_p}\right) \tag{2.9}$$

where τ is the effective time constant of the system, and is a function of the parameters a_{11} through b_{22}. The algebraic output equations can be written as follows:

$$T_{fo_R} = T_{fi_H} - \frac{\Sigma Hg}{\rho Q c_p} \tag{2.10a}$$

$$T_{fo_H} = g\left(T_{fi_H}, T_{amb_R}, Tamb_H, \frac{\Sigma Hg}{\rho Q c_p}\right) \tag{2.10b}$$

$$T_{fi_R} = T_{fo_H} \tag{2.10c}$$

In the above representation functions f and g are linear. Derivatives of the elements of the external input vector $\left[T_{amb_R}, T_{amb_H}, \frac{\Sigma Hg}{\rho Qc_p}\right]$ are required if the system is to be represented in the explicit state vector form.

By using the logarithmic mean temperature difference, instead of the arithmetic mean temperature difference, to define the bulk fluid temperature in the heat exchanger, Equation (2.5b) can be written as follows:

$$0 = -T_{fo_H} - \Delta T_H \ln \frac{T_{fi_H} - T_{amb_H}}{T_{fo_H} - T_{amb_H}} + T_{fi_H} \tag{2.11}$$

Since Equation (2.11) is algebraic, it does not change the order of the subsystem. However, it is no longer possible to establish the algebraic constraint equation in a form analogous to Equation (2.8), i.e., involving only the state vector elements and inputs.

The system can now be represented as follows:

$$\dot{\Delta T}_H = a_{11}\,\Delta T_H + [b_{11} \quad b_{12}] \begin{bmatrix} 2\,\Delta T_R + 2\,T_{amb_R} - T_{fo_H} + \frac{\Sigma Hg}{\rho Qc_p} \\ T_{amb_H} \end{bmatrix} \tag{2.12a}$$

$$0 = -T_{fo_H} - \Delta T_H \ln \frac{2\Delta T_R + 2T_{amb_R} - T_{fo_H} + \frac{\Sigma Hg}{\rho Qc_p} - T_{amb_H}}{T_{fo_H} - T_{amb_H}}$$

$$+ \left(2\Delta T_R + 2T_{amb_R} - T_{fo_H} + \frac{\Sigma Hg}{\rho Qc_p}\right) \tag{2.12b}$$

$$\dot{\Delta T}_R = a_{21}\,\Delta T_R + [b_{21} \quad b_{22}] \begin{bmatrix} T_{fo_H} \\ T_{amb_R} \end{bmatrix} \tag{2.12c}$$

The significant feature of the above model is that it cannot be reduced to explicit vector differential equation form:

$$\dot{\underline{x}} = \underline{f}\ (\underline{x},\ \underline{u},\ t) \tag{2.13a}$$

$$0 = \underline{g}\ (\underline{y},\ \underline{x},\ \underline{u},\ t) \tag{2.13b}$$

where

$$\underline{x} \ \underline{\Delta}\ [\Delta T_H,\ \Delta T_R]$$

$$\underline{u} \ \underline{\Delta} \left[T_{amb_H},\ T_{amb_R},\ \frac{\Sigma Hg}{\rho Q c_p}\right]$$

and

$$\underline{y} \ \underline{\Delta} \left[T_{fi_H},\ T_{fo_H},\ T_{fi_R},\ T_{fo_R}\right]$$

The examples above demonstrate that order reduction arising as a result of interconnection of subsystem models is not always apparent from inspection of the system equations, and that it is not always possible to obtain the explicit state vector representation for the large system. Also, the process of consolidating the subsystem models to arrive at a state vector of the minimal order generally destroys the modularity of the model.

Digital Simulation Considerations

The time domain simulation of differential algebraic equation sets relies on numerical integration to propagate trajectories of the state vectors, starting from known initial conditions. All conventional numerical integration methods, e.g., Runge-Kutta, Adams-Bashforth, Adams-Moulton (28, 29, 30), etc., require that the system differential equation be written in the form

$$\dot{\underline{x}} = \underline{f}(\underline{x},\ \underline{u},\ t) \tag{2.14}$$

where $\underline{x}$ is the state vector.

Single step methods, e.g. Runge-Kutta, use the state vector at time t_{n-1}, to establish the value of the state at time t_n. Functionally,

$$\underline{x}_n = \underline{f}^* (\underline{x}_{n-1}, \underline{u}_n, t_{n-1}, t_n) \tag{2.5}$$

where * is used to denote discretization. Multi-step methods, e.g., Adams-Bashforth, use the value of the state vector at multiple points in time for establishing the next value of the state, i.e., functionally

$$\underline{x}_n = \underline{f}^* (\underline{x}_{n-1}, \underline{x}_{n-2}, \ldots, \underline{x}_{n-k}, \underline{u}_n, t_{n-1}, t_n) \tag{2.6}$$

where k is the order of the method.

In either case, the quantity $\underline{x}_n$ is not allowed to appear in the right hand side and, therefore, no element of the state vector at time t_n is permitted to be a function of any other element of the state vector at time t_n. Consequently, if two elements x_n^1 and x_n^2 of $\underline{x}$ are constrained by the equation:

$$0 = h (x_n^1, x_n^2) \tag{2.17}$$

explicit integration methods will not assure that the constraint will be satisfied. Consequently, an attempt to use explicit integration techniques in the simulation of the thermal system modeled by Equations (2.5) through (2.7), will not guarantee that Equation (2.8) will be satisfied at each step in time. An additional difficulty in this example is that T_{fi_H}, T_{fi_R} which are needed for explicit numerical integration are not known at the beginning of the time step. Also, explicit integration methods cannot handle without iteration at each time step Equations (2.11a) and (2.11c) since these are of the form

$$\underline{\dot{x}} = \underline{f}\ (\underline{\dot{x}},\ \underline{x},\ \underline{u},\ t) \tag{2.18}$$

It is concluded that explicit numerical integration routines cannot in general handle systems of equations involving algebraic state constraints.

The class of systems under consideration generally exhibit pronounced nonlinear behavior. If a nonlinear system N is linearized around an operating point, the eigenvalues of the linear approximation L are called the eigenvalues of N at the specific operating point. The eigenvalues for a nonlinear system are generally speaking functions of the state and the input, and can consequently vary in an unpredictable manner. When the smallest and largest eigenvalues of a dynamic system are widely separated, the system is said to be stiff (23). It has been shown that explicit integration methods are usually inefficient for, and often incapable of, simulating stiff systems.

Review of Simulation Software

The remainder of this chapter will briefly critique four user-oriented digital simulation packages, which have been chosen to serve as paradigms of their respective classes. The discussion will be used to justify evolving a new approach, and is not meant to denigrate the use of the referenced software for their intended application.

SCEPTRE (31) is chosen as the representative of the generalized network approach. It is, in the words of its developers, "an automatic circuit analysis program capable of determining initial conditions, transient and steady-state responses of large networks." Depending as it does, on network terminology, it suffers from all the drawbacks of the network approach, which have been briefly mentioned earlier and discussed in more detail by Iyengar (32). The state variable concept is used by SCEPTRE; however, the reliance on explicit integration routines (trapezoidal rule and Runge-Kutta) requires, under certain conditions, which are described in terms of 'loops' and 'cut-sets', derivatives of functions to be furnished. In essence, these situations involve algebraic constraints on component state variables. SUPER-SCEPTRE is a preprocessor developed for use with SCEPTRE, aimed at simulating multi-degree of freedom mechanical systems (33). Subsystem models may be given in terms of generalized network parameters. Scalar nonlinear equations may be used to describe circuit elements. Since SUPER-SCEPTRE uses the same numerical techniques as SCEPTRE, it imposes the same restrictions on component interconnections and inputs. Even though SCEPTRE and SUPER-SCEPTRE are claimed to be written to analyze large systems, the model formulation does not display explicitly the topological structure. Additionally, since SCEPTRE uses the network approach, it has no provisions to use empirical and semi-empirical models of components, expressed as sets of differential-algebraic equations. The process of developing an equivalent network from such models is a retrogressive step in system simulation.

MARSYAS (34) developed for simulating 'large' aerospace systems primarily in the frequency domain, uses two canonical forms:

$$\dot{x}_i = A_i x_i + B_i u_i \tag{2.19}$$

$$y_i = C_i x_i \tag{2.20}$$

for linear components and

$$y_i = g_i x_i \tag{2.21}$$

for nonlinear components.

Interconnection between components are described by a vector equation:

$$U = EY + FV + KY \tag{2.22}$$

where U and Y are the consolidated input and output vectors, respectively, and V the external input vector to the large system.

Thus, all nonlinear components have to be dissected down to the level of nonlinear (and linear) elements described by scalar nonlinear equations. Explicit integration routines are used and, consequently, the state of the large system has to be the aggregate of the linear subsystem states. Also, MARSYAS does not allow the imposition of algebraic constraints on state variables via the interconnection equations. Additionally, MARSYAS does not handle static simulation and nor is it geared to handle stiff systems. The use of a preprocessor does, however, permit the storage of skeleton models and the user is allowed to write FORTRAN models as well. Consequently, empirical and semi-empirical models can be adjoined to the simulation package.

HYTRAN (24) is designed for aircraft hydraulic systems and is especially useful for systems having long transmission lines, since it uses the method of characteristics to model them. Prepackaged models of components like pumps, accumulators, etc., are used and the inclusion of empirical and semi-empirical state space models is difficult, if not impossible. The package is not suited for static simulation. HYTRAN also relies on explicit state vector representation for dynamic components and, consequently, cannot handle algebraic state constraints.

HYDSIM II (25) is a package written to simulate complex hydraulic systems using multiport component models. Components are modeled using the canonical form

$$\dot{x} = f(x, y, u, t) \tag{2.23}$$

$$0 = g(x, y, u, t) \tag{2.24}$$

Interconnections between components are modeled by:

$$u = h(x) \quad \text{or} \quad u = h(y) \tag{2.25}$$

The aggregate of equations for the entire system is block-oriented, each block representing a component. The assumption that the dependent port variable at an energy port has to be the independent port variable for the component to which the port is connected, introduces a constraint on the manner in which component models can be connected, i.e., certain types of connections are forbidden. The originator of the software package is cognizant of this restriction, since in the section entitled "Recommendations for Further Study" (25, pp. 59-60), he says:

> However, some of the areas in which improvements would be most beneficial are:
>
> 1. Develop a simulation algorithm which does not require the matching of port-variable dependencies at the component connections.

The matching of port variable dependencies in HYDSIM II ensures that the order of the large system is equal to the sum of the orders of the subsystems. The program package relies on explicit state vector representation and explicit integration (Runge-Kutta and Adams-Moulton) for propagation of state variables. Consequently, systems for which the state vector derivative cannot be written explicitly, e.g., the thermal system described by Equation (2.12), cannot be simulated by HYDSIM II. Additionally, the integration methods used in the package can become very inefficient and even unstable when simulating stiff systems. Another disadvantage of HYDSIM II is its reliance of prepackaged models which makes additions to the library of models difficult.

In summary, it has been shown in this chapter that the process of synthesizing a large system model using subsystem models expressed in the form of differential algebraic equations can result in order reduction, and that simulation software relying on explicit state vector representation are incapable of simulating systems with such order reduction. A second drawback of the present simulation programs which becomes apparent after a qualitative analysis of a system in the class under consideration is their inability to handle stiff systems efficiently.

In the next two chapters a new approach to modeling and simulation of large systems is presented. The new approach is based on concepts drawn from large scale systems theory, which are discussed in Chapter III, and the use of implicit representation for numerical integration and qualitative appraisal, which is the topic of Chapter IV.

CHAPTER III

THE LARGE SCALE SYSTEMS APPROACH

The limitations of the generalized network approach to systems analysis can be traced back, for the most part, to insistence on dissection of a system to basic energy storage and dissipative elements, i.e., inertances, capacitances, resistances, etc. The order of the state vector for a 'large' system is equal to the number of energy storage elements in the network; algebraic constraints on state variables are prohibited, since they violate the restrictions placed on topological structure (31). HYDSIM II (25) invokes 'port-dependency' conditions to prevent the interconnection of two subsystems in a manner which would result in algebraic state constraints. The examples given in the previous chapter demonstrate that it is physically possible to interconnect subsystems in such a way as to impose algebraic state constraints.

The objective of this chapter is to outline an approach which overcomes the above drawbacks. A 'large' system model is one in the form of a set of differential-algebraic equations in which

(i) the equations describing any individual subsystem are identifiable and not affected by changes in the model of other subsystems,

(ii) the equations describing the topological structure of the system are distinct, and

(iii) no restrictions are placed on the demarcation of subsystem boundaries.

A fundamental premise of the large scale system modeling approach is that in analyzing certain types of large physical systems it is usually advantageous to stop the process of dissection at an intermediate point rather than at the lowermost level. When the dissection is stopped at an arbitrary level, the description of subsystems assumes special importance. Once a mathematically adequate subsystem representation has been obtained, details of its internal structure are irrelevant to the description of the behavior of the large system.

Complete freedom in drawing boundaries around subsystems is desirable. Any diminution of this freedom, dictated by the need to meet the requirement of simulation techniques will detract from the usefulness of the modeling process itself. Also, it is desirable that subsystem models be complete and self-contained, and have few constraints on their applicability.

The problem of describing the behavior of a large system represented in the above manner reduces to:

1. Description of subsystems by suitable models;
2. Description of interconnections between subsystems in suitable mathematical terminology; and
3. Evolution of a procedure for generating output trajectories using information about initial conditions and input trajectories.

Each of these interrelated aspects will be considered in turn.

规范的

Subsystem Canonical Representation

A subsystem canonical representation is defined as a standard functional form in which the mathematical models for all subsystems are to be written. The reasons for evolving and scrutinizing canonical forms are to firstly examine the implications of connecting subsystems, in abstract terms, and secondly to examine the advantages of one form over another. Thus, for example, the effect of interconnecting two subsystems on the order of the large system can be examined in general terms, rather than considering each situation ab initio, as was done in Chapter II. Also, some forms may be more amenable for qualitative appraisal of system behavior than others, and other things being equal, such forms would be more attractive to the analyst.

In order for the analysis of large scale systems to be general, it is necessary to use a subsystem model form which can encompass all possible types of components and all possible methods of their interconnection. In the ensuing discussion, the explicit vector differential form, which has formed the basis of much of modern control theory (22, 35, 36) will be used, even though later in the development, an even more general form will be used.

The i'th subsystem will be represented by:

$$s_i \begin{cases} \dot{x}_i = f_i(x_i, u_i, t) & (3.3a) \\ y_i = g_i(x_i, u_i, t) & (3.3b) \end{cases}$$

Given the Lipschitz conditions, it can be shown that a unique solution to Equation (3.3) exists. A vast majority of physical systems, modeled with lumped parameter elements, can be described using the above canonical form.

Equation (3.3) is also suitable for qualitative appraisal of the dynamic behavior of the subsystem, since, in general, the differential equation (3.3a) can be expanded in a Taylor series in the neighborhood of an operating point, and the first term of the expansion used as an approximation of the plant matrix for eigenanalysis. The advantage of Equation (3.3) over 2.5 is that the differential and algebraic equations are decoupled and, consequantly, the latter may be ignored during eigenanalysis. In summary, Equation (3.3) is a good candidate for adoption as the canonical representation for subsystem.

Interconnection Between Subsystems

If there is one feature which can be considered characteristic of large systems, it is the explicit portrayal of the interconnection between the subsystems. In mathematical systems theory (37), three methods of interconnecting system models are discussed; namely, parallel, cascade, and feedback (see Figure 3). Before accepting these methods as being sufficient for the types of systems under study, it is necessary to examine the physical implications of interconnections.

The physical interconnection of fluid power systems is achieved through fluid conduits, mechanical linkages, or electrical wiring. These linking devices, if they are not treated as subsystems in their own right, generally impose equality constraints on certain physical variables associated with the energy or signal ports they link. The mathematical models used to depict the behavior of the subsystems must have, as inputs and outputs, these port variables. Consequently, in the case of most physical systems, the topological information describing the interconnection of subsystems, can be written in the form:

INTERCONNECTION OF SUBSYSTEMS

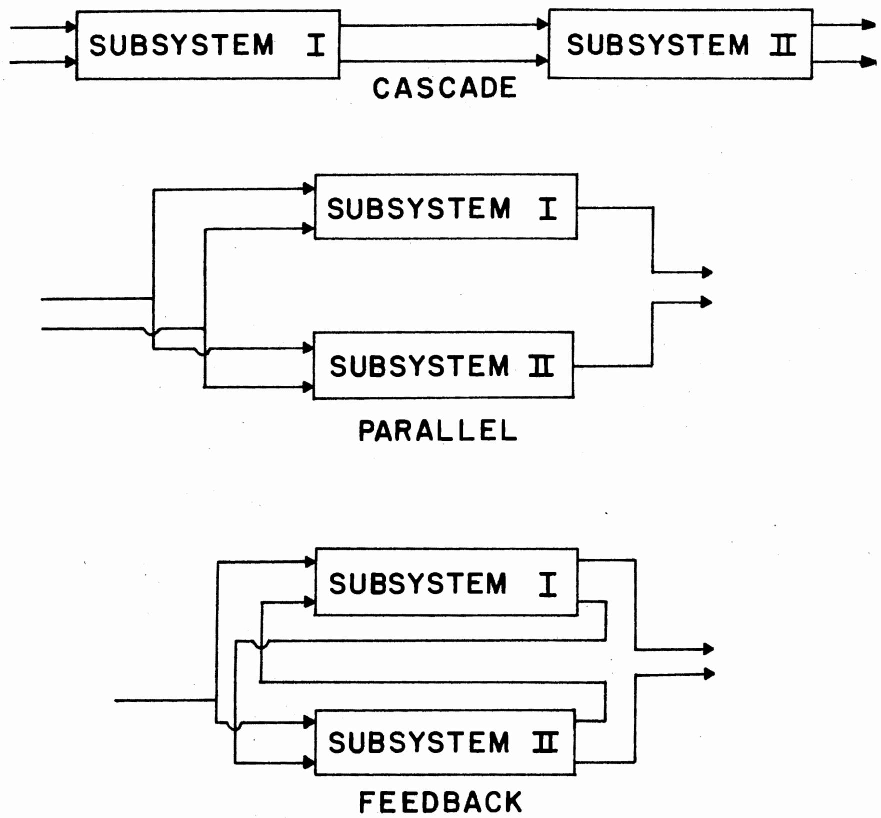


Figure 3. Possible Methods of Interconnecting Subsystems

$$0 = h(y_1, y_2, \ldots, y_n, u_1, u_2, \ldots, u_n, V) \tag{3.4}$$

where:

$V \triangleq (v_1, v_2, \ldots, v_{r_v})$ is the r_v dimensioned external input to the large system.

The large system is now represented by an aggregate of the subsystem models and the topological information as follows:

$$\dot{X} = f(X, U, t) \tag{3.5a}$$

$$Y = g(X, U, t) \tag{3.5b}$$

$$0 = H(Y, U, V) \tag{3.6}$$

where X, Y, and U are the aggregates of the subsystem state, output, and input vectors.

Consider now the connection of the ℓ'th port of the j'th subsystem to the t'th port of the k'th subsystem. The physical connection will impose the constraints

$$\left.\begin{aligned} u_{ja(\ell)} &= y_{kc(t)} \\ u_{kc(t)} &= y_{jb(\ell)} \end{aligned}\right\} \tag{3.7}$$

or

$$\left.\begin{aligned} u_{ja(\ell)} &= u_{kd(t)} \\ Y_{kc(t)} &= y_{jb(\ell)} \end{aligned}\right\} \tag{3.8}$$

where $a(\ell)$, $b(\ell)$, $c(t)$, and $d(t)$ are appropriate integers. Figure 4 shows how the physical constraints due to component interconnection translate into equality constraints on mathematical variables.

If three or more ports, each of a separate subsystem, are

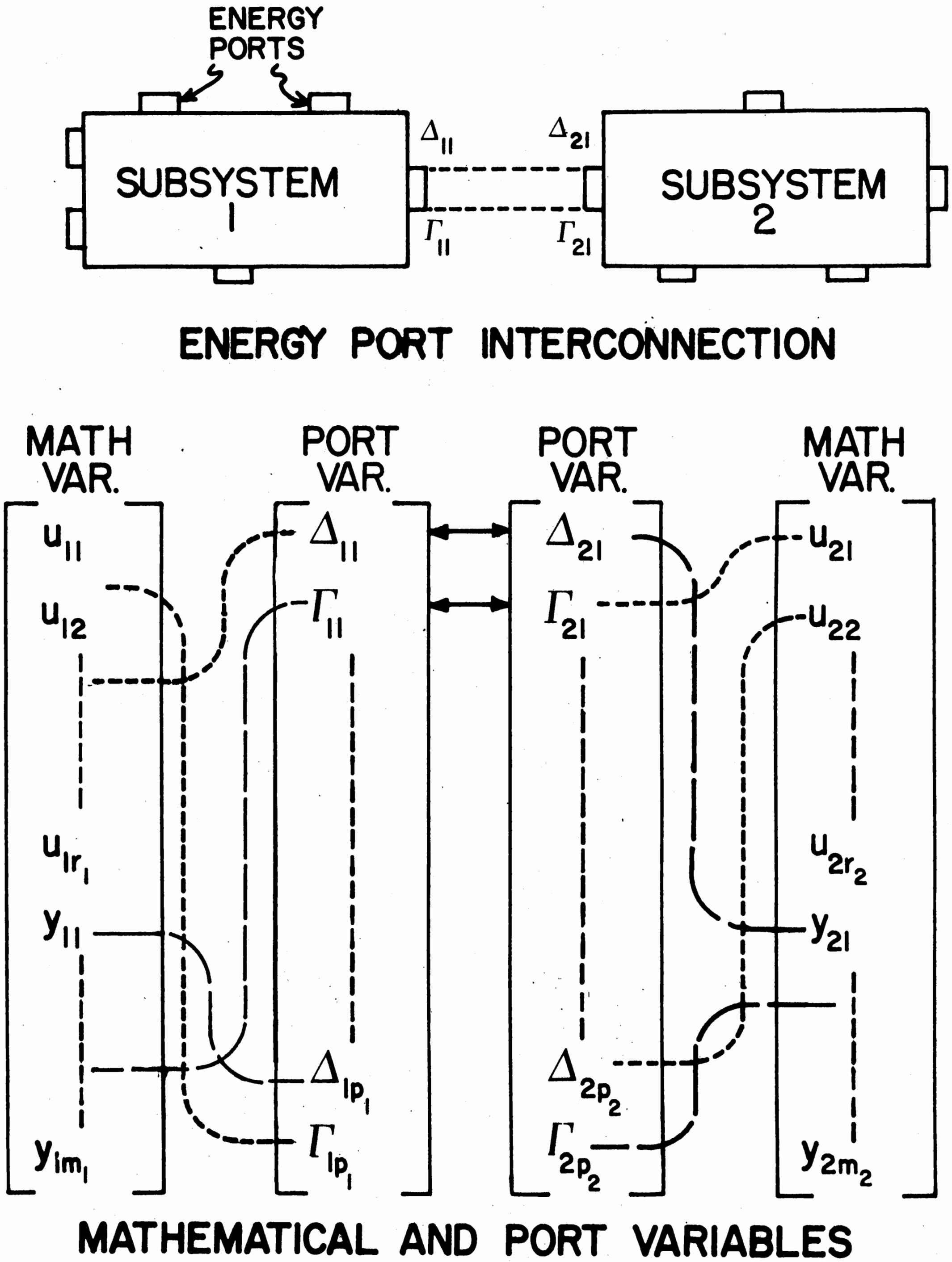


Figure 4. Physical Origin of Equality Constraints in the Topological Description of a Large System

interconnected, or if an external input is present at an energy port, or an external signal input is present, the generalized Kirchoff's laws can be used to write the constraint equations, which will still be algebraic.) It is important to emphasize that the physical interconnection is the basis of the topological constraint, and not vice versa. If two physical components can be physically connected it is natural to insist that they be portrayed in the mathematical description, rather than prohibit certain interconnections because they are mathematically inconvenient.

If all interconnection equations can be written in the form of Equation (3.7), the constraint equation for the large system reduces to

$$U = H_1 Y + H_2 V \tag{3.9}$$

where H_1 and H_2 are appropriate matrices. Ikeda and Kodama (37) give the conditions under which Equations (3.5) and (3.8) will represent a large system whose state vector is the aggregate of the subsystem state vectors.

It has been shown by Iyengar that it is possible to interconnect some subsystems so that the constraint equation for the large system is in the form

$$F[U \vdots Y] = G\ V \tag{3.10}$$

and it is not possible to express U explicitly in the form of Equation (3.9) (38). In this case the order of the large system is less than the sum of the orders of the subsystems. The previous chapter gave examples of large systems involving order reduction. The postulation and proof of a theorem concerning order reduction arising due to the interconnection of two subsystems represented by sets of differential-algebraic equations is contained in Appendix A.

Simulation Procedure

The large system described by Equations (3.5) and (3.6) is characterized by the existence of an aggregate input, U, which is quite distinct from the external input to the large system. By suitably manipulating the interconnection equations, it is often possible, especially in the case of linear systems, to eliminate U entirely.

The retention of the U vector poses a simulation problem when using explicit integration techniques since Equations (3.5) and (3.6) cannot be coded directly as FORTRAN (or equivalent) statements. There are three possible alternatives:

(a) symbolic manipulation of equations,

(b) use of 交搭 staggered elements in the U and Y vectors by introducing artificial delays, or

(c) solving an implicit algebraic equation at each step in time.

Alternative (a) lacks generality, even though its use for linear systems has been demonstrated (38, 39). Alternative (b) is used in some software packages (31), but the accuracy of simulation depends on the selection of the right vector elements to be delayed and the step size. Superficially (c) may appear attractive, but closer scrutiny will reveal that it is not since the implicit algebraic Equation (3.6), is coupled to the differential equation and, consequently, the latter will also have to go through the iterative solution procedure. Alternative (c) is not the same as that used in HYDSIM II (25) since in the latter, the propagation of the state vector does not need iteration at one point in time. HYDSIM II uses the chain rule to develop a 'pseudo' first-order differential equation for the algebraic variables.)

All of the three above alternatives fail when algebraic constraints on the state vector of the large system are present. There is another area of weakness in software depending on explicit integration methods which has been mentioned earlier. Explicit integration methods are unusable for stiff systems (23, 40, 41). Stiff systems are characterized by the existence of widely different real parts of the largest and smallest eigenvalues--typically of two orders of magnitude or more. As explained by Gear (41), Blostein (41), Orlandea et al. (43) explicit integration methods either require very small step sizes and are, consequently, subject to round off errors in digital computation, apart from being inefficient, or go unstable. In the nonlinear systems of the type under consideration, the 'stiffness' changes from region to region in state space and, hence, the importance of using a method which is robust and efficient under the widest range of stiffness.

The need for an algorithm for qualitative appraisal of the dynamic behavior of large systems is also evident. Needless to say the coupling of subsystems implies that, in general, the eigenvalues of the large system will not be the eigenvalues of the subsystems themselves.

This chapter has outlined the large scale systems approach by examining subsystem canonical representations, the implications of physical interconnections between subsystems and the problems of simulating systems involving implicit constraints on not only algebraic variables but possibly state variables as well. The next chapter develops and explains the philosophy of implicit representation, as applicable to dynamic physical systems, and demonstrates that it has the potential to overcome the drawbacks of explicit state vector representation.

CHAPTER IV

LARGE SCALE SYSTEM SIMULATION USING IMPLICIT REPRESENTATION

The explicit vector differential form of representation for lumped parameter dynamic physical systems appears to arise so naturally that alternatives are rarely considered. However, the main reason for this formulation is that the use of the generalized network approach which almost always relies on explicit numerical integration techniques demand that form (42). In this chapter, a new form, which overcomes the drawbacks of the explicit state vector representation is evolved.

Consider the following canonical form for representing the i'th subsystem:

$$S_i \begin{cases} 0 = f_i\,(\dot{x}_i,\ x_i,\ w_i,\ t) & (4.1a) \\ 0 = g_i(x_i,\ w_i) & (4.1b) \end{cases}$$

where w_i is defined as the aggregate input-output vector, i.e., $w_i \equiv (u_i \vdots y_i)$. It is obvious that Equation (4.1) subsumes the earlier form, Equation (3.3). The introduction of w is given the following justification: When a system is being analyzed, the first requirement is to identify the input and outputs. A system model is expected to show explicitly these inputs and outputs. (However, in the case of a subsystem, it is conceivable that there is some degree of freedom in assigning inputs and outputs (44); i.e., the constitution of the input and output vectors depends upon the other subsystems and possibly the external inputs to the large system. Rather than change the subsystem model whenever the interconnections change, as advocated by Rosenberg (44), it is easier to use a generalized input-output vector for a subsystem. The use of an implicit algebraic equation needs no special defense, since static components are often used in hydraulic systems and the implicit form is more general than the explicit form.

The large system obtained by aggregating subsystems in the above canonical form can be represented by

$$0 = f(\dot{X}, X, W, t) \tag{4.2a}$$

$$0 = g(X, W) \tag{4.2b}$$

$$0 = h(W, V) \tag{4.2c}$$

where

$$W = (U \vdots Y)$$

It is seen that Equation (4.2) subsumes the explicit representation for a large scale system; namely, Equations (3.5) and (3.6). Consequently, any analysis or simulation performed by using Equation (4.2) can still use the explicit representation.

It is well recognized that a qualitative understanding of the behavior of a dynamic system is essential for the selection of digital simulation parameters--step sizes, error bounds, etc. Such an appraisal is relatively simple for systems represented by explicit differential equations. If the above canonical form is used, the first term of the Taylor's series expansion of the differential equation can be obtained only by using the implicit function theorem of differential calculus. Appendix B contains an algorithm, which has been developed and computerized as part of this research, for establishing the eigenvalues of a large system expressed in the above canonical form. The algorithm will be used in appraising the qualitative behavior of example systems.

Digital Simulation

The development of trajectories in dynamic system simulation is an example of the initial value problem in differential equations. It is well known that explicit integration routines are incapable of handling efficiently stiff systems, characterized by widely divergent real eigenvalues (23, 41, 43).

A number of implicit integration techniques, pioneered by Gear (23, 40, 42, 45, 46, 47) have been recently developed to handle stiff systems. It should be mentioned that most of these techniques still require the model to be in explicit vector differential form (40, 41). Gear has extended the implicit integration method to handle differential algebraic systems expressed in the form

$$0 = f(\dot{\underline{X}}, \underline{X}, \underline{V}, t) \tag{4.3}$$

where $\underline{X}$ need not be the state vector (62). Appendix C gives a brief review of the method. It is easily seen that Equation (4.2) can be written in the form (4.3).

Equation (4.3) by itself can be considered as the canonical form for representing not only subsystems, but also the large system. In the first case, it would contain the equations relating the sub-subsystem input to the subsystem state and output. In the second case, it would contain not only the models for all subsystems, but also the topological information. The vector $\underline{X}$ will be defined as the differential-algebraic state vector in order to distinguish it from the genuine state vector for the large system.

As an example of model formulation using implicit state representation the equations for the thermal system analyzed in Chapter II will be reconstituted to conform to Equation (4.3). The input vector to the system is

$$\underline{V} \triangleq [T_{amb_H} \; T_{amb_R} \; \frac{\Sigma Hg}{\rho Q C_p}]$$

The differential algebraic state vector for the system is defined as

$$\underline{X} = [\Delta T_H \; T_{fi_H} \; T_{fo_H} \; \Delta T_R \; T_{fi_R} \; T_{fo_R}]$$

The equations describing the subsystems and topological constraints can now be written as follows:

$$0 = -\dot{x}_1 + a_{11} x_1 + [b_{11} \; b_{12}]\begin{bmatrix} x_2 \\ v_1 \end{bmatrix} \tag{4.4a}$$

$$0 = -x_3 - x_1 \ln\left(\frac{x_2 - v_1}{x_3 - v_1}\right) + x_2 \tag{4.4b}$$

$$0 = -\dot{x}_4 + a_{21} x_4 + [b_{21} \; b_{22}]\begin{bmatrix} x_5 \\ v_2 \end{bmatrix} \tag{4.4c}$$

$$0 = -x_6 + 2x_4 + 2v_2 - x_5 \tag{4.4d}$$

$$0 = -x_3 + x_5 \tag{4.4e}$$

$$0 = -x_2 + x_6 + v_3 \tag{4.4f}$$

It may be noted that implicit representation is not only easy to use, but also exhibits individual subsystem models and topological constraints as partitions of the large system model. The external inputs to the large system, i.e., the elements of $\underline{V}$ could have been specifically excluded from appearing in the subsystem models by appending two additional elements of the differential algebraic state vector and adding two equations to the topological constraints.

It may be noted that Gear's algorithm for differential algebraic systems does not require explicit identification of the state vector. Even though, in principle, it is possible to include in $\underline{X}$ all system variables except the inputs to the large system, it is advantageous from the point of view of simulation, to append to Equation (4.3), an explicit algebraic equation

$$\underline{Z} = g(\underline{X}, \underline{V}, t) \tag{4.4}$$

where $\underline{Z}$ is termed the explicit algebraic variable vector. This vector could be constituted of all subsystem outputs which can be explicitly expressed in terms of the input vector $\underline{V}$ and the differential algebraic vector, $\underline{X}$, and which do not influence any of the subsystem inputs.

Equations (4.3) and (4.4) together constitute the canonical form for the large system. The next chapter illustrates the use of this canonical form for both qualitative analysis as well as digital simulation.

Software Development

The only documented digital simulation package which implements Gear's method for implicit differential algebraic systems in ECAP II (48). This package, written for static and dynamic analysis of electronic networks, could be used for simulating other systems by casting them in the 'network' mold. The drawbacks of the network approach to analyzing large mobile hydraulic systems have already been discussed in

Chapter II and elsewhere (32). An additional drawback to the use of ECAP II is its huge requirement of computer memory, primarily because of its precompiler and bookkeeping technique. Needless to say, these features make it difficult, if not impossible, to extract intermediate variables, and interface other FORTRAN subprograms. Consequently, it was decided to build the large scale system simulation program using Gear's numerical integration program (49), DFASUB, for propagating the differential-algebraic state vector.

Figure 5 exhibits the calling structure of the FORTRAN program evolved for simulating the type of systems under consideration. No attempt has been made to develop a user-oriented package corresponding to HYTRAN, HYDSIM II, or other similar software (24, 25). A brief description of the main program and key subroutines follows.

MAIN Program

This program is used to initialize all parameter, arrays, etc., and read information pertinent to individual components, e.g., actuator sizes, inertia, and drag coefficients, valve metering characteristics, etc. It is also used to read integration control parameters; namely, maximum, minimum, and starting step sizes, allowable error and final time, as well as initial values of elements of the differential-algebraic state vector, and their first derivatives. It also reads the input trajectory.

The program sets up the differential-algebraic state vector in the form needed for numerical integration by DFASUB. If in the course of a trajectory simulation, it is found necessary to change from one representation of a subsystem to another, the main program is used to

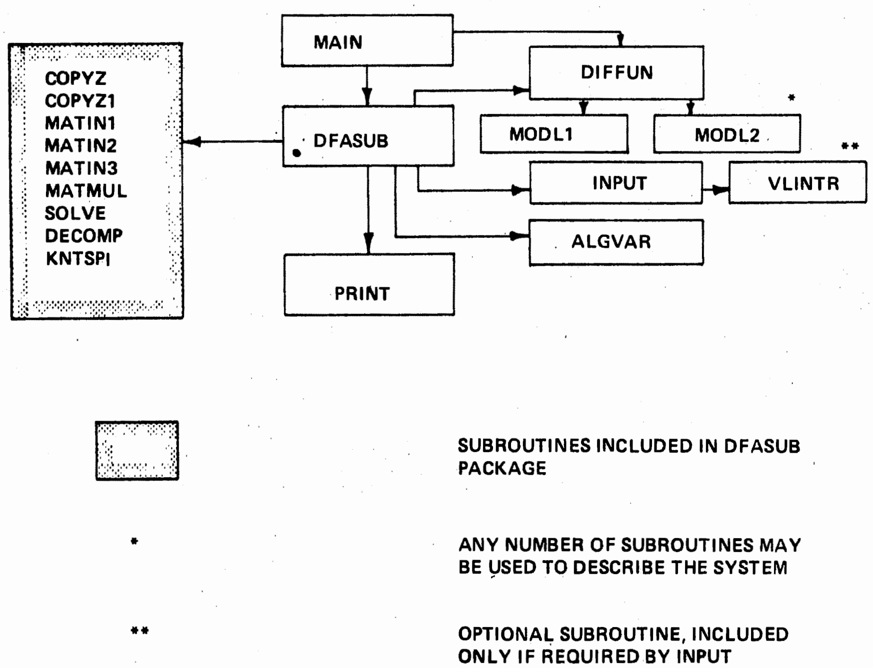


Figure 5. Calling Structure of a Large System Simulation Program Involving Model Switching

ascertain when a switch is needed, and reconstitute the elements of the Y vector, and transmit only the pertinent variables to DFASUB. MODL1 and MODL2 are two representations between which switching is performed according to preselected criteria. Additional models can be added as necessary. Chapter V illustrates the program logic by means of an example simulation involving model switching.

DFASUB

This is the integration routine which develops the trajectories of the differential algebraic state vector, as constituted by MAIN. Since the program is documented elsewhere (48), only the modifications required to handle large systems will be described here. The values of the input vector at any prescribed time are obtained by calling INPUT, as many times as may be necessary for the Newton iteration which is part of implicit integration. Explicit algebraic variables, i.e., those which can be written as explicit functions of the differential-algebraic state and the inputs to the large scale system, are obtained by calling ALGVAR. Print-out of trajectories after a prescribed number of steps is done by alling PRINT. To perform the implicit integration, DFASUB uses the error vector generated by DIFFUN, and a number of matrix manipulation routines enclosed in the shaded box in Figure 5.

DIFFUN

This subroutine furnishes DFASUB with the correction to the differential-algebraic state vector before the latter performs the Newton iteration. The equations describing all the subsystems as well as the topological constraints may be included in DIFFUN. However, in the present version, in order to facilitate switching of component representations it is used as a director subprogram which called the pertinent system model written in one or more subprograms.

ALGVAR

This subroutine is written specifically to evaluate explicit algebraic variables whose inclusion in the differential algebraic-state vector would have resulted in unnecessary matrix manipulation, core-storage and computing time. ALGVAR is called just before the print-out step, so that no calculations need be made for steps which are not printed out.

PRINT

This subroutine performs a dual function. First, it is used to control the print-out of trajectories of the pertinent system variables. Secondly, it is used to check, at each step in time, if the criteria for switching from one representation another have been met, and if so, to return an appropriate message to the MAIN program.

The main program and most of the subprograms, with the exception of DFASUB, were specifically written to simulate the example system presented in the next chapter. However, with changes in the quantities that are printed out at the beginning of a simulation, the program can be used to simulate any large system expressed in the canonical form evolved herein.

CHAPTER V

EXAMPLE SYSTEM ANALYSIS

Even though the variety in circuitry exhibited by mobile hydraulic systems is much more than hydraulic and electro-hydraulic servo-systems, a vast majority of mobile hydraulic systems are characterized by the following features.

1. Modularity: Each actuator, together with its control elements (directional control valve, relief and flow control valves) is a distinct subsystem. Two or more subsystems may be identical.
2. Multiplicity of inputs: Two or more actuators may be in motion at the same time, as a result of human operator or other inputs.
3. Task Oriented Duty Cycle: For a prescribed task, the inputs and the actuator motions form a well-defined cycle. A machine may be capable of a multitude of tasks.

An example system has been chosen to explore the feasibility, efficacy and limitations of digital simulation based on implicit representation. The system, shown in Figure 6, which exhibits all the characteristics detailed above, is the simplified hydraulic circuit of a backhoe. A brief description of its operation will lead to a better appreciation of the modeling and simulation problems involved in describing its behavior.

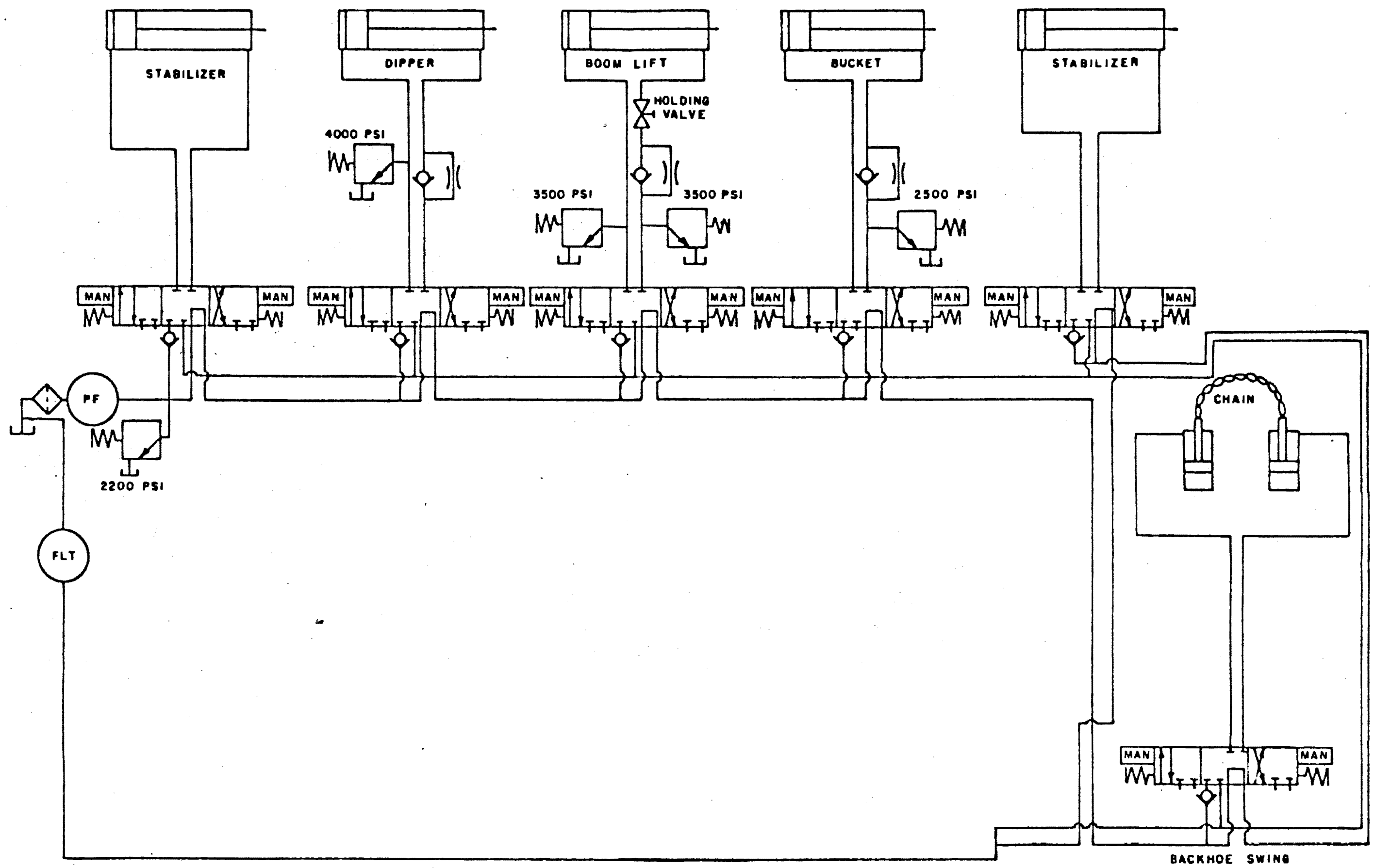


Figure 6. Simplified Hydraulic Circuit Schematic of a Backhoe

The system is open centered and is generally synthesized from off-the-shelf components; namely, pump, directional control valves, pressure and flow control valves, cylinders or motors and fluid conditioners (filters, oil coolers, etc.). The directional control valves, which are usually manually operated, may be of the en bloc or stack design. In either case, using the 'pressure beyond' capability of the open center valve, it is possible to incorporate additional actuator subsystems by merely interposing them in the open center return path.

For simplicity in presentation here only two actuators will be considered to be in operation. Extension to more actuators is straightforward. Figure 7 presents the circuit schematic for the two actuators, and explicitly identifies the pump subsystem, in addition to the actuator subsystems. Figure 8 is a 'network' description of the actuator subsystems, intended firstly to demonstrate that each subsystem is a dynamic system in itself, and secondly to highlight the interconnections between the subsystems. Since the pump is considered to be the first subsystem, the actuator subsystems are labelled as '2' and '3', respectively. From a hierarchical viewpoint, Figure 7 presents one level of dissection of the large system, i.e., into subsystems, while Figure 6 presents the system at the lowermost level of dissection, i.e., at that of basic elements. Figure 8 also illustrates the identical nature of actuator subsystems, i.e., exactly the same equations are used to describe the dynamic behavior of both subsystems. It needs to be emphasized that the actuators can be modeled as identical subsystems only if the topology of the large system is explicitly described. Thus, in Figure 8 P_{s2} and Q_{s2} are the port variables at an energy port of actuator number 2, in precisely the same manner as P_{s1} and Q_{s1} are the port

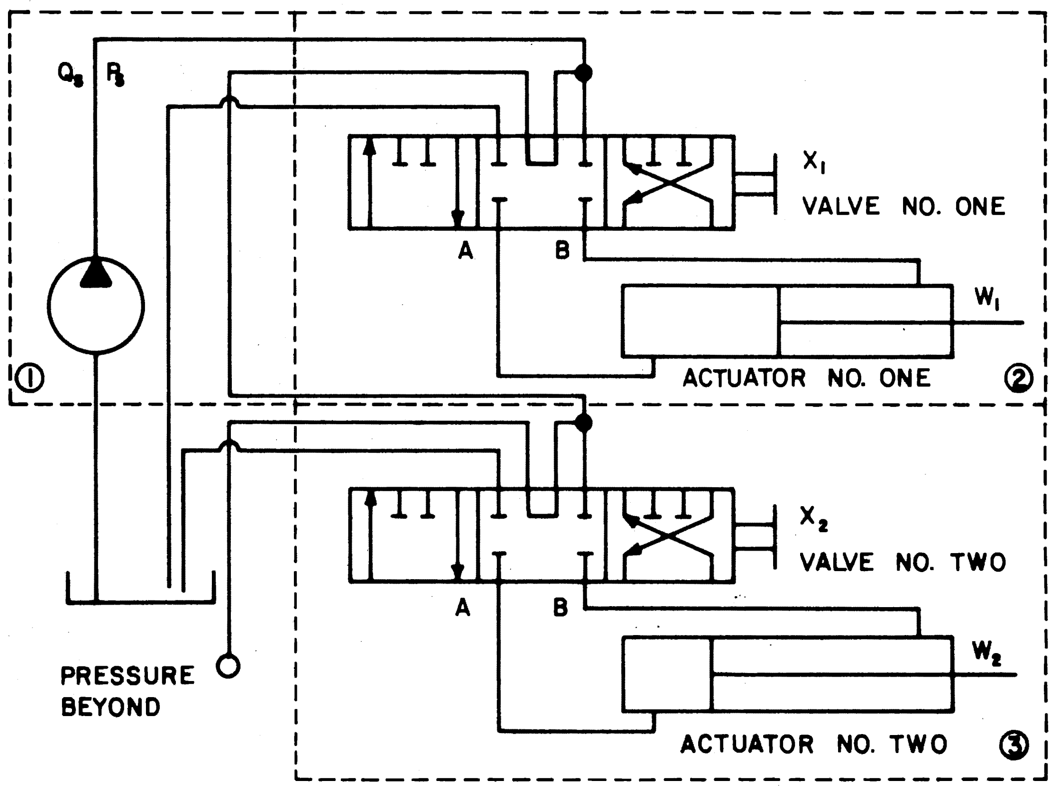


Figure 7. Hydraulic Circuit Schematic for Two-Actuator Open-Center System Showing Demarcations for Individual Subsystems

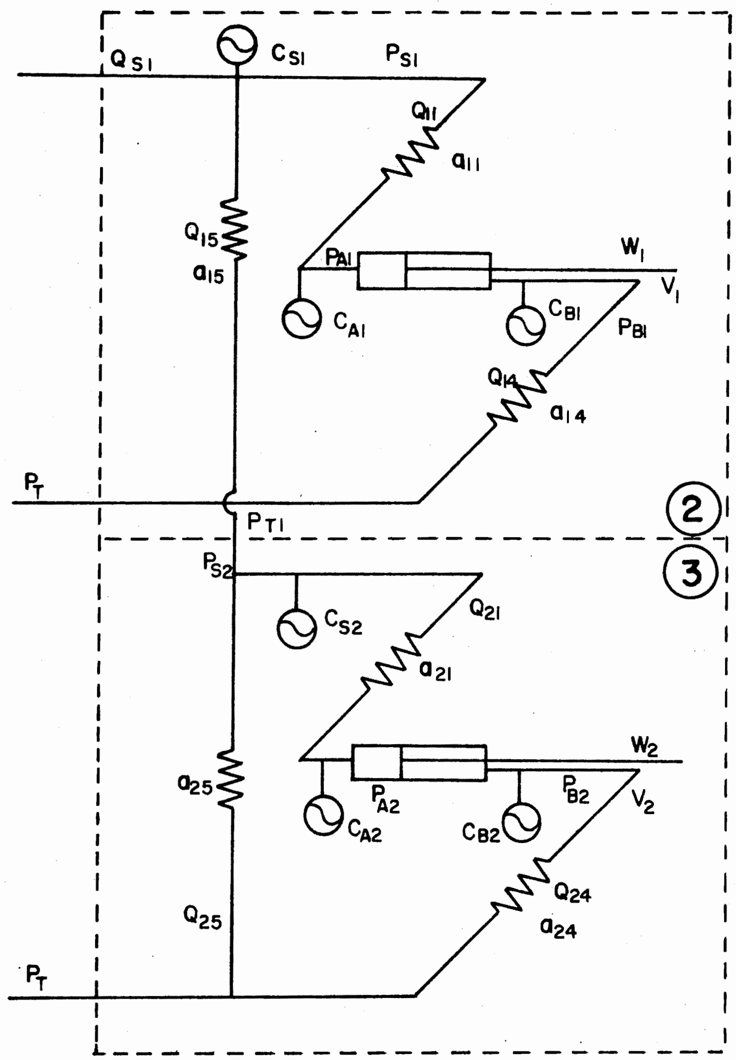


Figure 8. Network Representation of a Two-Actuator Open-Center System

variables for actuator number 1. The fact that $P_{T1} = P_{s2}$ and $Q_{15} = Q_{s2}$ arises as a result of the interconnection of the two subsystems, and is explicitly shown as such, even though at the cost of introducing additional variables.

The selection of static and dynamic effects to be included in a mathematical description of a system is based on the information content desired to be incorporated in the model (50). The actuator subsystem models presented here incorporate the capacitance effects of the line and cylinder volumes on both sides of the piston, in addition to the actuator inertia and drag. The open center valve, which is treated as part of the actuator subsystem, is described by a numerical algebraic model based on the Wheatstone bridge analogy (12). Since the pump is to be treated as an ideal flow source, and there is no interest in establishing the pump input torque, the pump subsystem need not be modeled so as to account for the variables at all its energy ports. Consequently, the pump subsystem will be treated as an ideal flow source, and Q_s will then be an element of the external input vector to the large system.

The sixteen implicit differential algebraic equations used to describe the two actuator open center system can be obtained by modeling firstly the two actuator subsystems, and secondly the topological constraints. The relevant equations which are identified below as elements of a functional equation F are as follows:

Subsystem #2 (Actuator System #1)

$$F_1: \quad 0 = \dot{P}_{S1} + (Q_{S1} - Q_{11} - Q_{15})\,\frac{1}{C_{s1}}$$

$$F_2: \quad 0 = -Q_{11} + ka_{11}(P_{S1} - P_{A1})^{1/2}$$

$$F_3: \quad 0 = -Q_{15} + ka_{15}(P_{S1} - P_{T1})^{1/2}$$

$$F_4: \quad 0 = -\dot{P}_{A1} + (Q_{11} - v_1 A_{A1}) \frac{1}{C_{A1}}$$

$$F_5: \quad 0 = -\dot{v}_1 + (P_{A1} A_{A1} - P_{B1} A_{B1} - W_1 - B_1 v_1) \frac{1}{I_1}$$

$$F_6: \quad 0 = -\dot{P}_{B1} + (v_1 A_{B1} - Q_{14}) \frac{1}{C_{B1}}$$

$$F_7: \quad 0 = -Q_{14} + ka_{14}(P_{B1} - P_T)^{1/2}$$

Subsystem #3 (Actuator System #2)

$$F_8: \quad 0 = -\dot{P}_{S2} + (Q_{S2} - Q_{21} - Q_{25}) \frac{1}{C_{S2}}$$

$$F_9: \quad 0 = -Q_{21} + ka_{21}(P_{S2} - P_{A2})^{1/2}$$

$$F_{10}: \quad 0 = -Q_{25} + ka_{25}(P_{S2} - P_T)^{1/2}$$

$$F_{11}: \quad 0 = -\dot{P}_{A2} + (Q_{21} - v_2 A_{A2}) \frac{1}{C_{A2}}$$

$$F_{12}: \quad 0 = -\dot{v}_2 + (P_{A2} A_{A2} - P_{B2} A_{B2} - W_2 - B_2 v_2) \bar{I}_2$$

$$F_{13}: \quad 0 = -\dot{P}_{B2} + (v_2 A_{B2} - Q_{24}) \frac{1}{C_{B2}}$$

$$F_{14}: \quad 0 = -Q_{24} + ka_{24}(P_{B2} - P_T)^{1/2}$$

Topological Constraints:

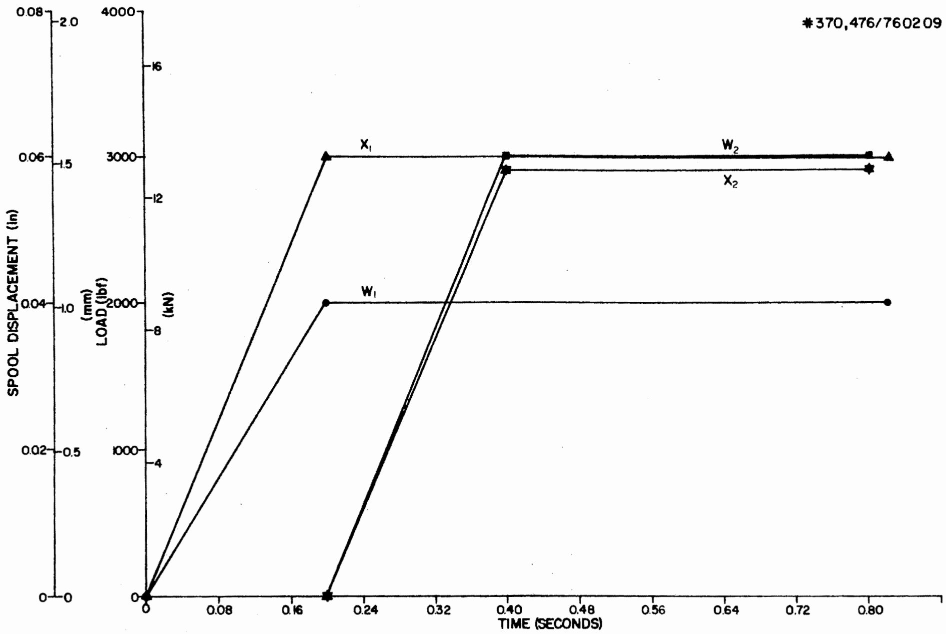
$$F_{15}: \quad 0 = -Q_{S2} + Q_{15}$$

$$F_{16}: \quad 0 = -P_{S2} + P_{T1}$$

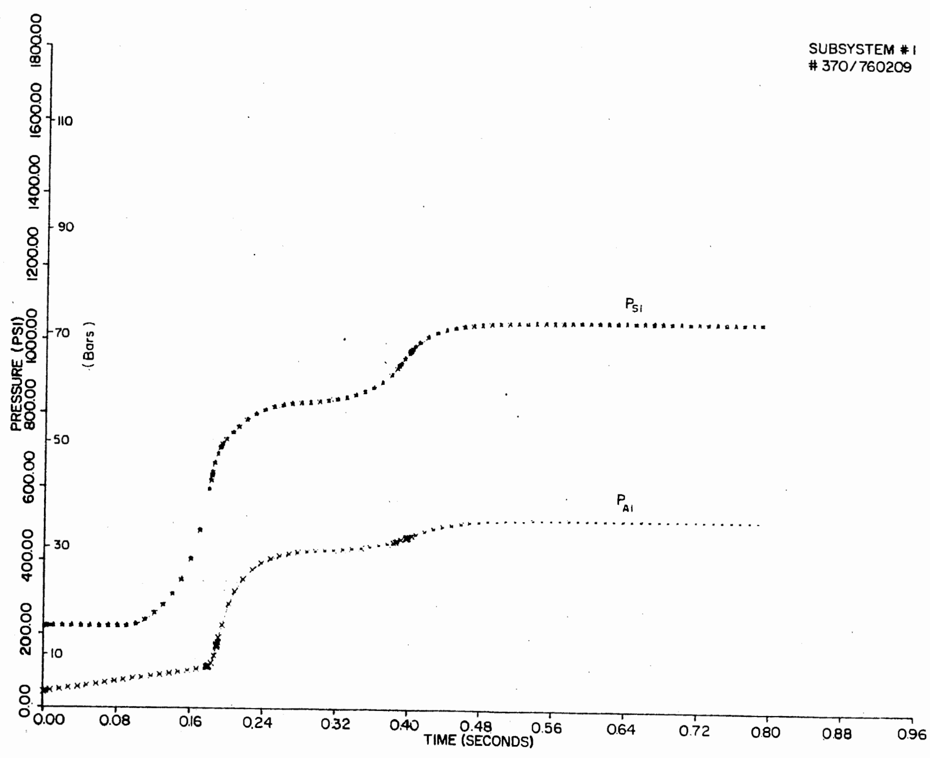
The external input vector to the system is $[Q_{S1} P_T W_1 X_1 W_2 X_2]$ where X_1 and X_2 are the spool displacements for the directional control valves in subsystems two and three, respectively. It is of interest to note that firstly, Q_{S1} and P_T are invariant for a given system, and secondly, that X_1 and X_2 are indirect inputs, in the sense that they determine the metering orifice areas a_{11}, a_{14}, a_{15} and a_{21}, a_{24}, a_{25}, respectively. In the course of a duty cycle, X_1 and X_2 and, consequently, the metering areas are changed by the human operator.

Numerical values of parameters are presented in Appendix E. Figures 9 and 10 present the trajectories of selected system variables for two inputs intended primarily to demonstrate the success of model formulation and the advantages of variable step-size integration (which controls the density of the identification characters in the computer-generated plots of output trajectories). Figure 9(a), for example, presents ramp inputs of spool displacements and load to both actuator subsystems, while Figures 9(b), 9(c), and 9(d) depict corresponding trajectories of cylinder pressures and velocities. The intercation between the two actuator subsystems is evidenced by the change in cylinder pressures and velocity for the first actuator, when the second is put in motion. A typical machine duty cycle would be composed of one or more trapezoidal inputs as depicted in Figure 10(a) and the corresponding outputs as in Figures 10(b), 10(c), and 10(d). These simulations reveal the efficacy of the implicit integration method in handling systems where dynamic and steady-state operation are interspersed.

It should be remarked that the step size is limited to the maximum specified by the analyst, and this parameter may be changed by the analyst, in the course of a trajectory simulation, so as not to waste time in the calculation of unnecessary intermediate steps. However, the maximum step size should be chosen such that changes in the input are taken into consideration in addition to the dynamics of the system.

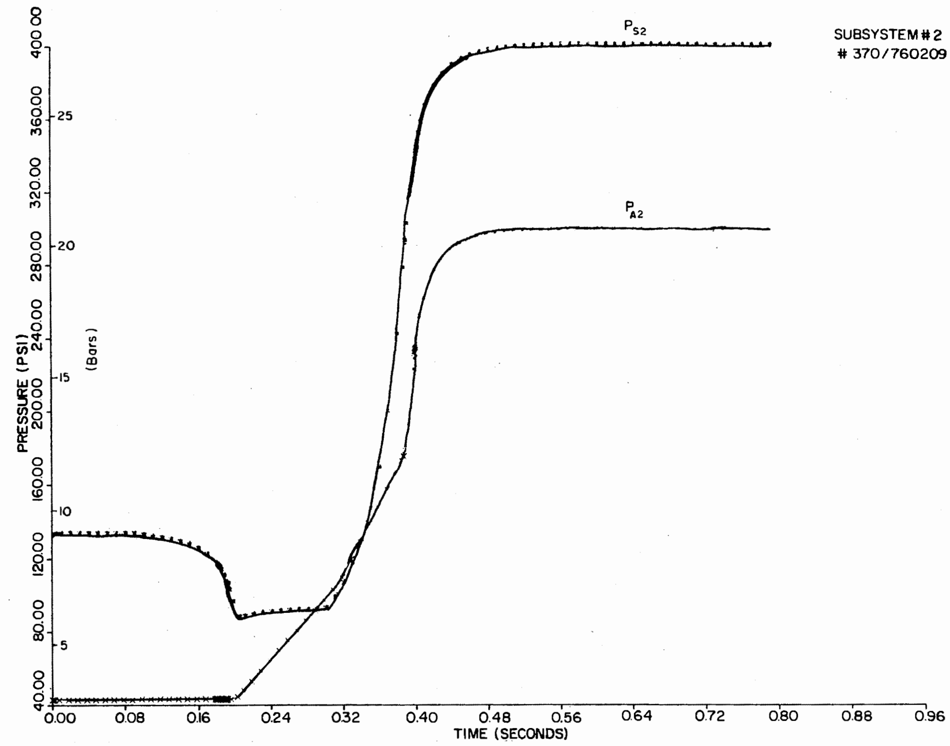


(a)

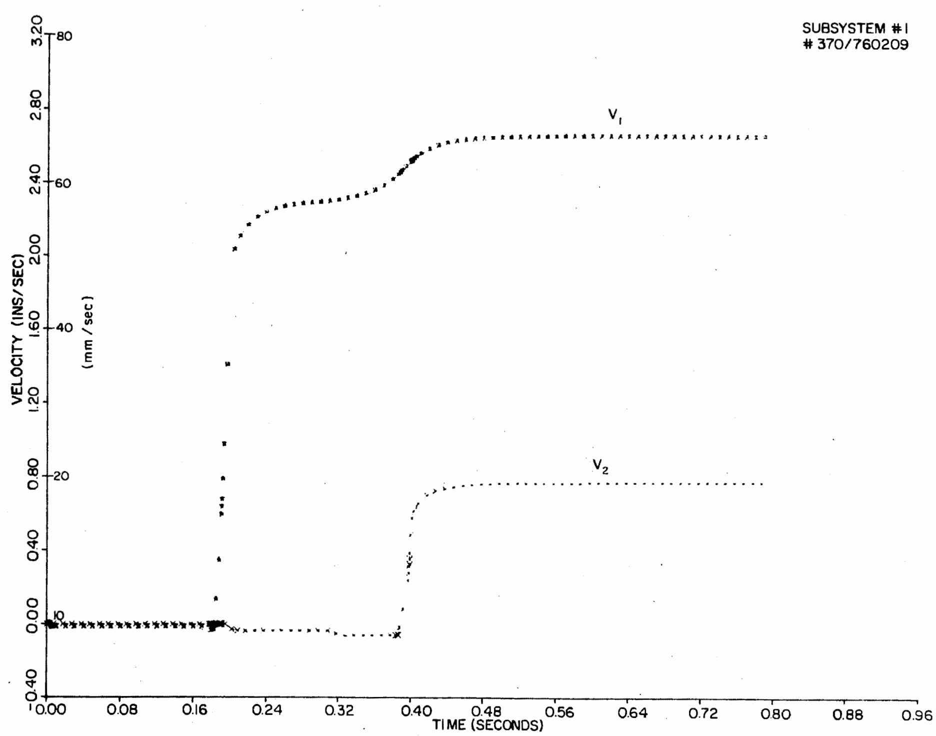


(b)

Figure 9. Simulated Trajectories of Two-Actuator System Variables for Ramp Inputs

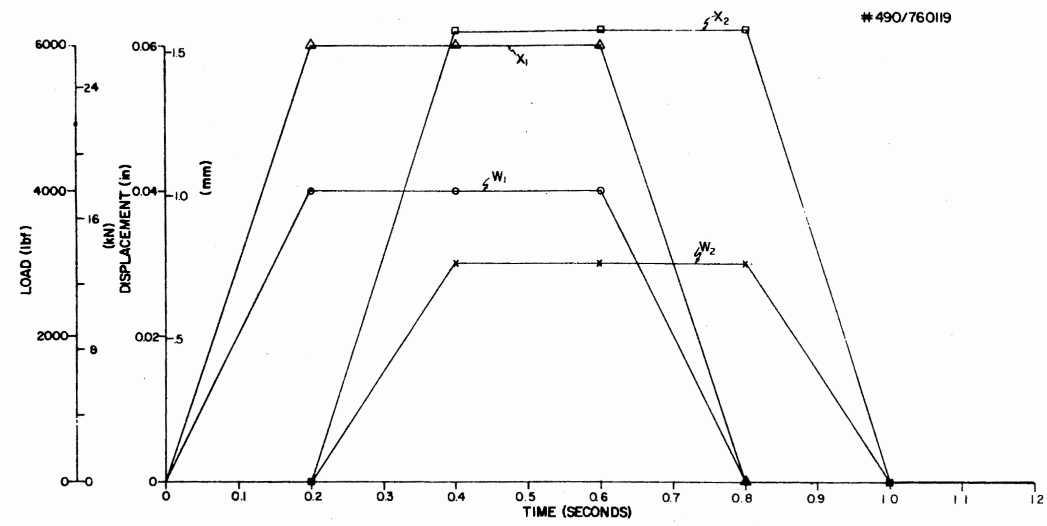


(c)

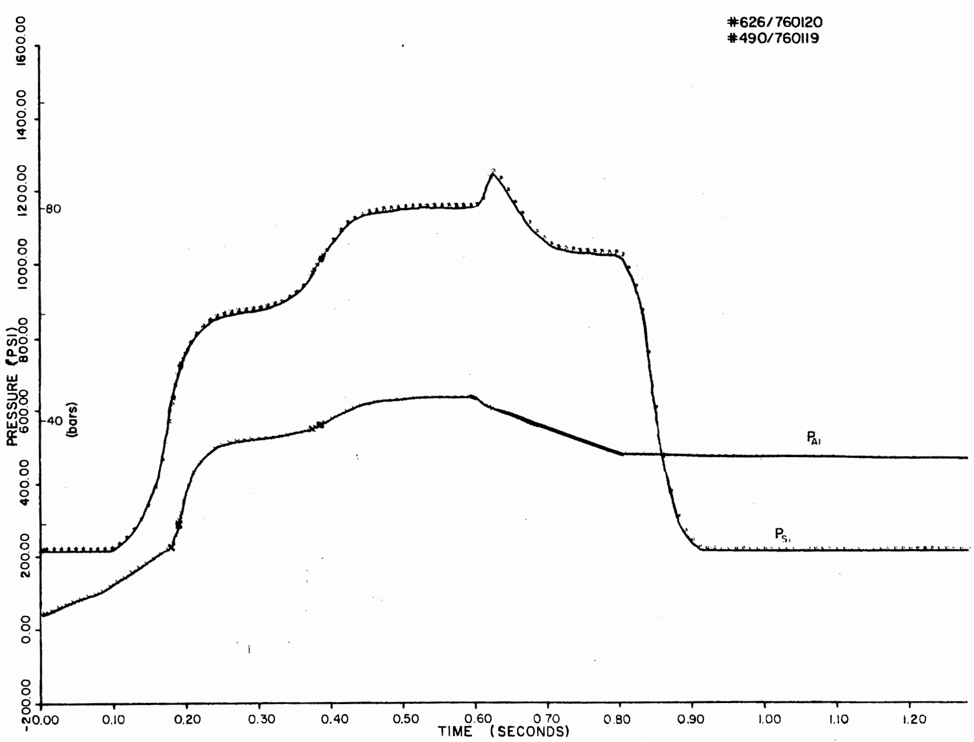


(d)

Figure 9. (Continued)

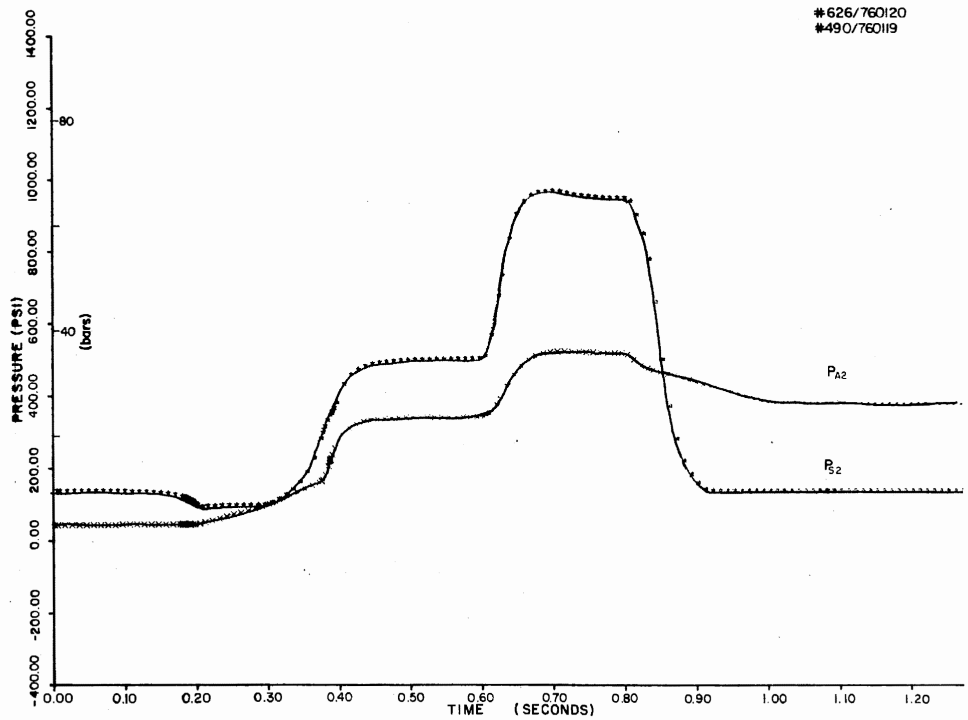


(a)

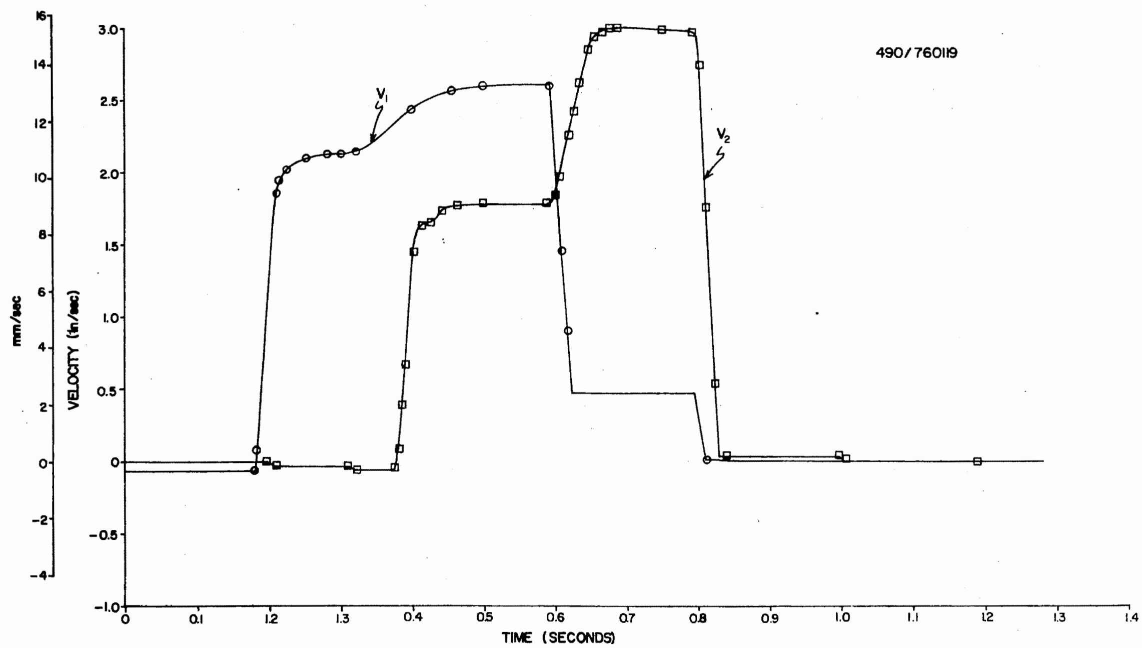


(b)

Figure 10. Simulated Trajectories for Two-Actuator System Variables for Trapezoidal Inputs



(c)



(d)

Figure 10. (Continued)

The successful calculation of trajectories based on specific inputs does not, however, give a good indication of the qualitative aspects of simulation, which are necessary for a general appraisal of a new technique. Application of specially designed benchmark problems to Gear's implicit integration method have revealed some of its strengths and weaknesses (41, 51). The main strength of Gear's method, as indicated by the tests described in the above references, lies in its ability to handle stiff systems which are characterized by non-oscillatory eigenvalues; its main weakness, which is not considered serious (51) was its inefficient simulation of highly oscillatory trajectories. However, these tests by Gear (41), and Enright, Hull, and Lindberg (51) were conducted on explicit differential equations rather than implicit differential-algebraic equations which form the basis of the new approach, i.e., they did not investigate the effect of nonlinear algebraic equations on simulation speed or efficiency.

In order to exploit to the fullest extent the advantages offered by implicit integration for simulating large mobile hydraulic systems and also to compensate for its disadvantages, the following areas were considered worthy of investigation:

1. qualitative study, through eigenanalysis, of the example system to examine its stiffness characteristics
2. study of effect of hard constraints on simulation by implicit integration
3. study of feasibility and utility of switching models in the middle of a trajectory

In the ensuing sections the results of the investigation will be summarized.

Qualitative Behavior of Two-Actuator Open-Center System

It is well known that the dynamic behavior of a linear time-invariant system can be qualitatively appraised by a scrutiny of the eigenvalues of the plant matrix. The qualitative behavior of a non-linear system in a prescribed region of state space can be obtained by linearization if the equations describing it are in the explicit vector differential form, i.e., Equation (3.3) and the functions are continuous and differentiable (52). Appendix B develops the expression for the local plant matrix of a nonlinear implicit differential algebraic system. A computer program written to perform the necessary matrix manipulations and solve the characteristic equation was used to analyze both single and double actuator open center systems at various points on their operating region. Since the actual numerical values of the eigenvalues depends not only on the operating region in the state space, but also the system parameters, general conclusions regarding all open center systems cannot be drawn. Nevertheless the results obtained by analyzing specific systems are very instructive. For example, Iyengar (26) has shown that a single actuator system with 'small' inertia and drag can exhibit stiffness ratios of the order of 10^7 or more, and would consequently be difficult, if not impossible, to simulate by explicit integration methods. It has also been shown by Iyengar (53) that 'small' inertia and drag do not necessarily lead to non-oscillatory eigenvalues. Consequently, it is conceivable that simulation of open center systems by implicit integration may be slowed down, and even made inefficient, due to the presence of complex eigenvalues with large imaginary parts.

Since in the course of the simulation of a trajectory, the eigenvalues can vary continuously, implicit integration offers the advantage of being the only method which has the potential of simulating an entire trajectory with no human interference. It should also be noted that in order to portray oscillatory trajectories faithfully, any numerical integration scheme will have to use step sizes significantly smaller than the time period of oscillations.

Table I summarizes the results of eigenanalysis performed at selected times of trajectories developed for the example system. It is of interest to note:

1. the maximum allowable step size is used even when the system is fairly stiff
2. the step size is not always curtailed by the presence of complex eigenvalues with large imaginary parts
3. the mere inclusion of the second actuator subsystem, which may be inoperative at the time under consideration; can change the stiffness ratio of the system
4. a very small or zero value for metering orifice a_{15} always resulted in a small step size, even though the stiffness ratio was not far different from other regions in state space for which a much larger step size was used.

The general conclusion that can be drawn from the qualitative analysis are:

1. Open center mobile hydraulic systems can exhibit wide range of stiffness ratios and can have eigenvalues with large and small imaginary parts.

TABLE I

SUMMARY OF SIMULATION RUNS ON EXAMPLE SYSTEM

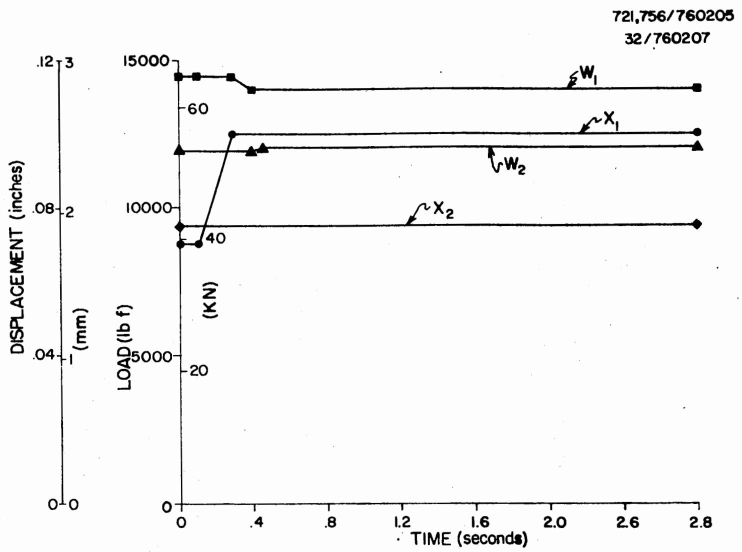
S. No.	Simulation Run No.	Time	Actuator #1					Actuator #2					Current Step Size	Eigenvalues		Stiffness Ratio
			Spool Disp.	Metering Areas			Load	Spool Disp.	Metering Areas			Load		Largest	Smallest[d]	
			x_1	a_{11}	a_{14}	a_{15}	W_1	x_2	a_{g1}	a_{g4}	a_{25}	w_2		λ_{max}	λ_{min}	σ
1	193/760203	0.615	0.0650	0.0288	0.0288	0.01570	2000.0	0.060	0.0175	0.0175	0.0314	1000.0	0.0050[a]	- 783.2 +j1959	- 34.6	22.63
2	283/760202	0.000	0.0750	0.1092	0.1092	0.00000	0000.0	0.000	0.0000	0.0000	0.1324	0.0	0.0001	- 1045.4 ± 1999.2	-179.0	5.84
3	126/760129	0.243	0.0615	0.0209	0.0209	0.02664	4866.5	0.000	0.0000	0.0000	0.1324	0.0	0.0050[a]	- 1232.0	- 43.0	28.65
4	126/760129	0.243	0.0165	0.0209	0.0209	0.02664	4866.5	-	-	-	-	-	-	- 782.4[c] ±j1958	- 44.0[c]	17.78
5	126/760129	0.721	0.0670	0.0333	0.0333	0.00940	8000.0	0.064	0.0265	0.0265	0.0188	1000.0	0.0050[a]	- 882 ±j1984	- 33.0	26.73
6	126/760129	1.240	0.0000	0.0000	0.0000	0.13240	0000.0	0.000	0.0000	0.0000	0.1324	0.0	0.0050[a]	- 1985.3	289.6	6.86
7	764/760129	0.226	0.0677	0.0348	0.0348	0.00720	4513.0	0.000	0.0000	0.000	0.1324	0.0	0.0050[a]	- 3460.5	- 43.2	80.10
8	764/760129	0.291	0.0872	0.0796	0.0796	0.00000	5812.0	0.000	0.0000	0.0000	0.1324	0.0	1.9×10^{-4}	- 888.9 ± 1985	90.7	9.80
9	764/760129	0.291	0.0872	0.0796	0.0796	0.00001	5812.0	0.000	0.0000	0.0000	0.1324	0.0	1.9×10^{-4}	- 888.9 ±j1985.3	90.7	9.80
10	764/760129	0.291	0.0872	0.0796	0.0796	0.00000	5812.0	-	-	-	-	-	-	- 888.9[c] ±j1985.3	- 90.6[c]	9.80

[a]Maximum allowed step size.

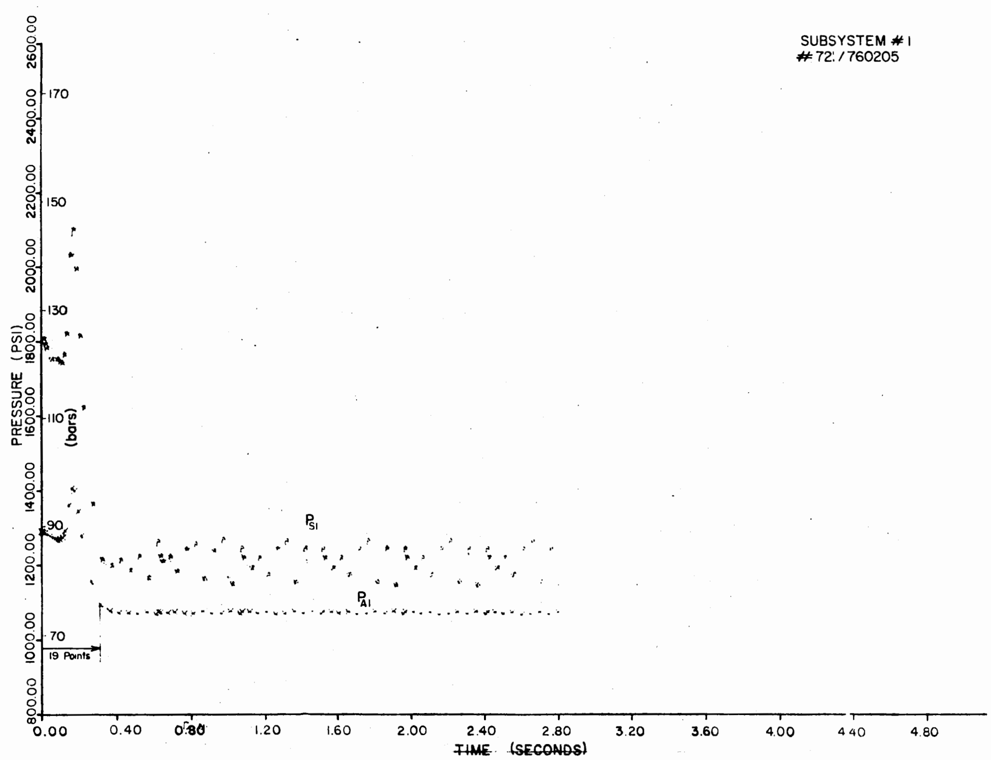
[b]Defined as $\dfrac{|\lambda_{max}|_{\text{real part}}}{|\lambda_{min}|_{\text{real part}}}$

[c]Eigenvalues for first actuator subsystem only.

[d]Smallest non-zero eigenvalues.

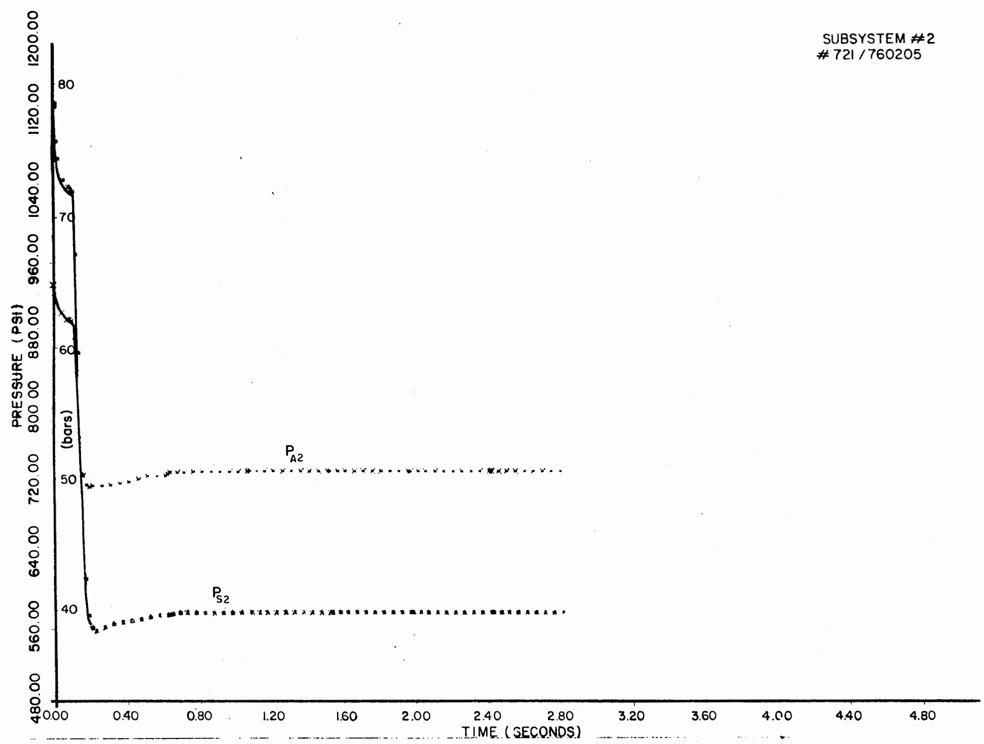


(a)

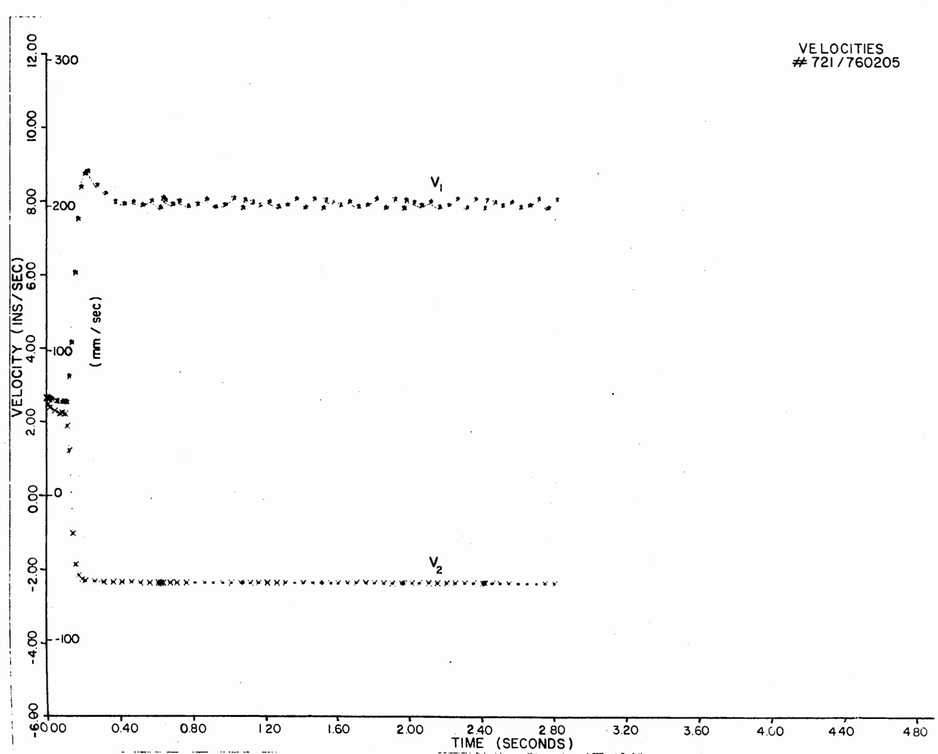


(b)

Figure 11. Trajectories Generated Without Imposition of Hard Constraints on System Variables



(c)



(d)

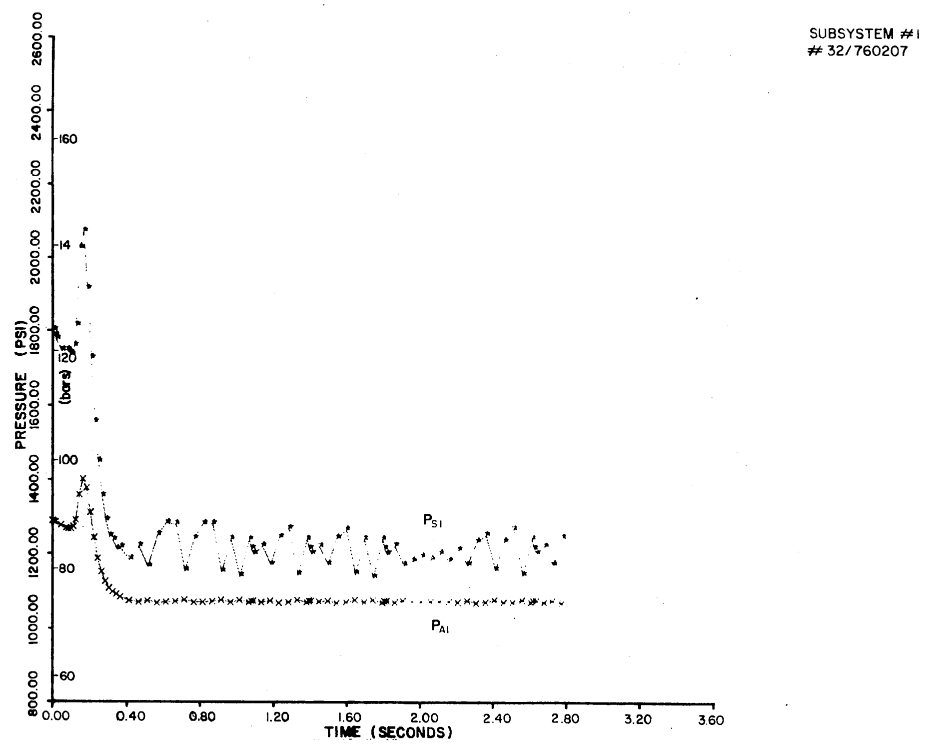
Figure 11. (Continued)

2. Implicit integration will not fail for such systems, though it could necessitate unacceptably small step sizes.

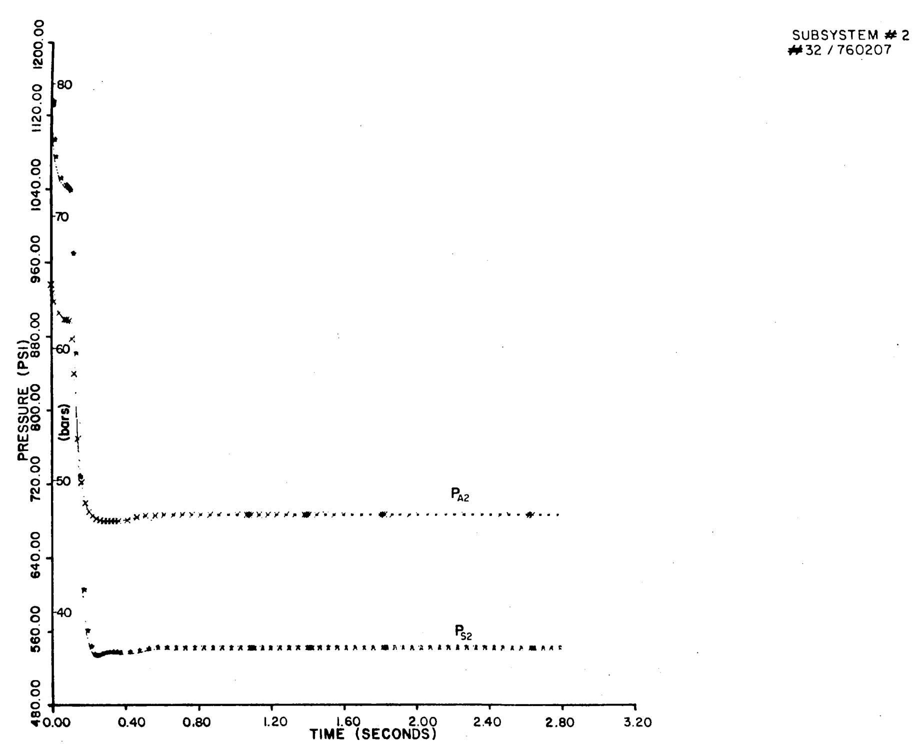
3. The step size is not exclusively dependent on the eigenvalues of the system.

Effect of Hard Constraints

The implicit integration method depends on a Newton-like iteration to solve a set of nonlinear equations at each step in time (40, 48). Consequently, the imposition of hard constraints on any element of the differential algebraic state vector can be expected to result in non-convergence under certain circumstances. However, for the dynamic systems under consideration, the discontinuity is arrived at only gradually in the generation of the trajectories and, consequently, it should be possible to stay within a prescribed tolerance band around the hard constraint, if step sizes are kept sufficiently small. This conjecture is borne out by the example trajectories shown in Figures 11 and 12. The input was chosen so as to cause P_{B2} to fall below zero if no hard constraint was imposed. Figure 10 presents some of the state variables for the above condition. The steady-state value of P_{B2}, reached at 0.813 seconds, was -55.2 psi. Figure 12 presents the same state variables with the imposition of hard constraints on all pressures, i.e., any pressure below zero was corrected to be zero in subroutine DIFFUN (or MODL1 or MODL2). In view of the finite error bound specified for the nonlinear equation solving routine, the final value of the constrained states can deviate from the hard constraints up to a maximum of the specified amount. In the simulation shown in Figure 12, the actual

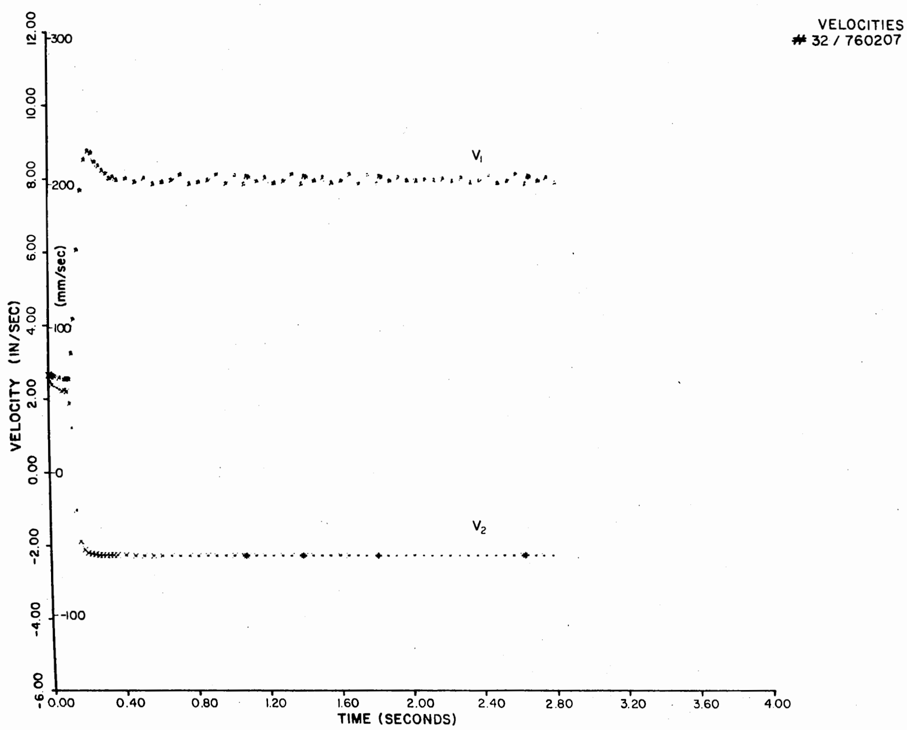


(a)



(b)

Figure 12. Trajectories Generated Using Same Inputs as in Figure 11, With Hard Constraints on System Pressures



(c)

Figure 12. (Continued)

value of P_{B2} ranged from -12.9 to 0.0004 psi, the deviation from zero becoming less as time progressed. It is interesting to note that firstly other pressures and actuator velocities are not significantly affected by P_{B2} being zero, and secondly the step size for steady-state operation was 0.005 seconds, the maximum value specified with and without hard constraints. Hence, it is concluded that under proper circumstances, implicit integration can adequately handle hard constraints of the type encountered in large mobile hydraulic systems.

Model Switching

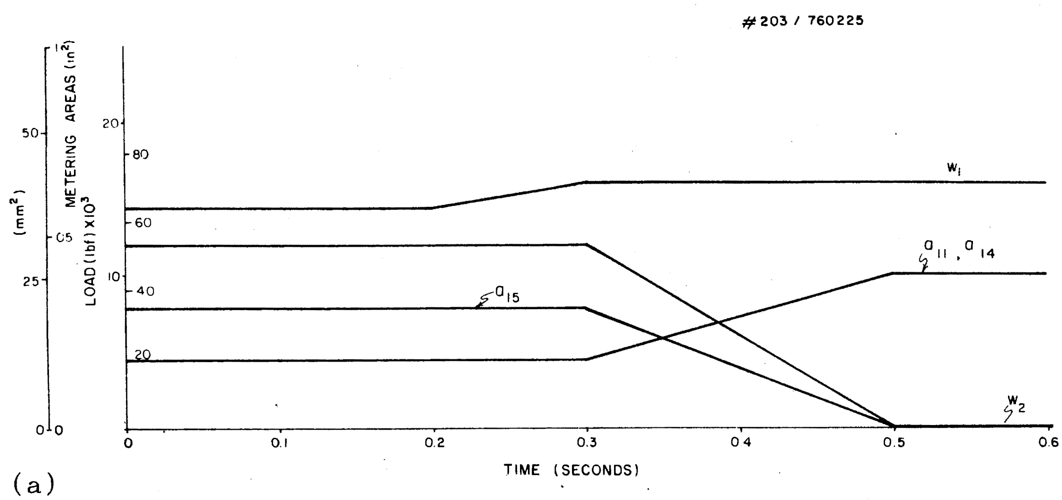
It is not uncommon for mathematical functions describing fluid power components to display discontinuities. The portrayal of hysteresis, stiction, coulomb friction, etc., is usually performed by changing the functions used to describe the phenomena as dictated by physical considerations. Similar changes in functional representation are also necessary at the subsystem level in order to exploit the hierarchical structure of a large scale system. One of the compelling reasons for exploring the use of Gear's algorithm for simulating large mobile hydraulic systems was its insensitivity to the relative numbers of algebraic and differential equations (41, 42, 43). This feature is exploited by ECAP II for obtaining steady-state solutions for electronic networks (48).

If the equations used to describe the large system do not involve the derivative of the differential-algebraic state vector Y, the state vector of the system is of zero dimension, i.e., the system is purely static in nature. Since conventional explicit integration routines, e.g., Runge-Kutta, Adams-Bashforth, etc., cannot handle state vectors of zero dimension, the advantages of Gear's method for differential algebraic systems is obvious.

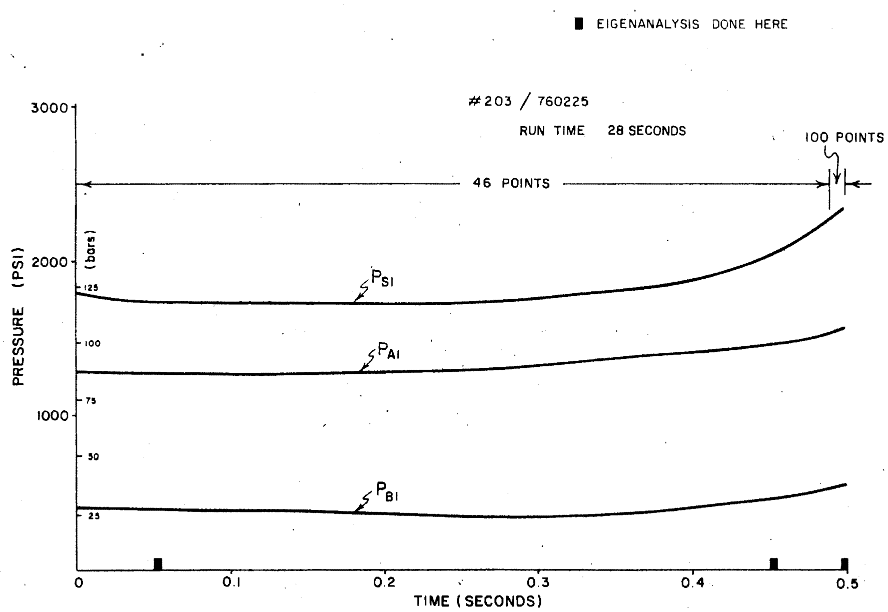
It has been mentioned in Chapter I that large mobile hydraulic systems are characterized by the interposition of static behavior between periods of dynamic operation. Figure 10 is an example of such operation. Since Gear's method is indifferent as to whether a specific variable is genuinely dynamic or otherwise, it would be reasonable to conjecture that one model, with a prescribed differential-algebraic state vector, would be adequate to simulate both static and dynamic phases of a trajectory. Example trajectories presented earlier (Figures 9 and 10) and reported by Iyengar elsewhere (32, 54) show that this is indeed true under certain circumstances.

However, it was noted that whenever the spool position of the directional control valves reached values such that a_{15} and/or a_{25} became very small (typically 2×10^{-4} sq. ins.), the simulation step size would become extremely small (see Figure 13), or an abnormal termination flag would be returned by the integration subroutine DFASUB. Results of eigenanalysis of the system, at various points in the trajectories, some of which are presented in Table II demonstrated that the system was not necessarily stiff under the above circumstances. Failure or inefficiency of the simulation was traced to ill-conditioning of the PW matrix for the differential-algebraic system (49). The Gear method uses a Gaussian elimination algorithm for inversion of PW, which fails when the matrix is ill-conditioned.

It is of interest to examine the PW matrix for the two actuator open center system. As indicated in Appendix C, the PW matrix is the Jacobian of the discretized version of the implicit differential



(a)



(b)

Figure 13. Trajectories to Selected Input, Showing the Effect of a_{15} Becoming Nearly Zero

algebraic system and is given by

$$PW \equiv J - \left[\frac{\partial F}{\partial \underline{y}} - \frac{\alpha_o}{h\beta_o}\frac{\partial F}{\partial \dot{\underline{y}}}\right]$$

where F is the implicit functional representation, h the current step size, and α_o and β_o coefficients in the algorithm (49).

The sixteen equations which comprise F for the example system, have been presented earlier as the mathematical models for the actuator subsystems and the topological constraints. Figure 14 presents the PW matrix and indicates thereon the sixteen elements of the differential algebraic state vector and correspond to the y's in the matrix entries. It may be noted that PW is a sparse matrix with predominantly diagonal submatrices, which correspond to the subsystems. Non-zero entries in off-diagonal matrices indicate the coupling between subsystems. The lowest diagonal submatrix, which is the contribution of the topological constraints is seen to be invariant.

Simulation of the example system, starting from different initial conditions, and using various input trajectories, invariably resulted in either extremely small step sizes or abnormal termination of simulation, when a_{15} and/or a_{25} became zero or very small (typically 0.001 sq. ins.). Abnormal termination messages suggested that the PW matrix was ill conditioned. An inspection of the stiffness ratios for the PW matrix from runs which stalled, presented in Table II, confirms that the PW matrix may be ill conditioned even though the local plant matrix of the differential algebraic system is not extremely stiff. A scrutiny of the contents of the PW matrix for the example system, pres nted in Figure 14, shows that if the step size, h, is sufficiently small, $\frac{\alpha_o}{h\beta_o}$ would become

TABLE II

SUMMARY OF EIGENVALUES FOR EXAMPLE SYSTEM

Run No.	Time	Step Size	Eigenvalues of System λ_{max}	λ_{min}	Stiffness Ratio σ_1	Eigenvalues of PW Mat $\bar{\lambda}_{max}$	$\bar{\lambda}_{min}$	Stiffness $\bar{\sigma}$**	Remarks
801/760209	0.151	0.05	-7.73×10^3 $\pm j1.96 \times 10^3$	-29.00	266.5	-791.68 $\pm 1.9 \times 10^3$	0.973 $\pm j259$	813	Normal simulation
"	0.716	76×10^{-6}	6.96×10^3	-27.00	257.67	-18.1×10^3 $\pm j1.9 \times 10^3$	0.388	46.6×10^3	$a_{15} = 0.0$
"	0.716	76×10^{-6}	6.96×10^3	-27.00	257.67	-18.1×10^3 $\pm j1.9 \times 10^3$	0.390	46.3×10^3	$a_{15} = 100 \times 10^{-6}$
"	0.728	9.4×10^{-6}	50.66×10^3	-27.29	1856.4	-18.1×10^3 $\pm j1.9 \times 10^3$	0.399	45.36×10^3	$a_{15} = 100 \times 10^{-6}$; Simulation stalled
203/760225	0.0511	0.05	5.94×10^3	-34.96	170	-792 $\pm j1.9 \times 10^3$	.913 $\pm j0.246$	867	Normal simulation
"	0.453	0.05	2.8×10^3	-76.2	36.75	-792 $\pm j1.9 \times 10^3$	1.0 $\pm j0.05$	792	Normal simulatio
"	0.5	$.57 \times 10^{-3}$	15.3×10^3	-30.3	503	-4.4×10^3 $\pm j1.9 \times 10^3$	0.95	4.63×10^3	$a_{15} = 100 \times 10^{-6}$; Simulation stalled

*Defined as $(|\lambda_{max}|_{real} / |\lambda_{min}|_{real})$.

**Defined as $(|\bar{\lambda}_{max}|_{real} / |\lambda_{min}|_{real})$.

	P_{S1}	Q_{11}	Q_{15}	P_{A1}	v_1	P_{B1}	Q_{14}
F_1	$\frac{\alpha_o}{h\beta_o}$	$\frac{-1}{C_{S1}}$	$\frac{-1}{C_{S1}}$	0	0	0	0
F_2	$\frac{-ka_{11}}{2(y_1-y_4)^{1/2}}$	1	0	$\frac{ka_{11}}{2(y_1-y_4)^{1/2}}$	0	0	0
F_3	$\frac{-ka_{15}}{2(y_1-y_{15})^{1/2}}$	0	1	0	0	0	0
F_4	0	$\frac{1}{C_{A1}}$	0	$\frac{\alpha_o}{h\beta_o}$	$\frac{-A_{A1}}{C_{A1}}$	0	0
F_5	0	0	0	$\frac{A_{A1}}{I_1}$	$\frac{\alpha_o}{h\beta_o} - \frac{B_1}{I_1}$	$\frac{-A_{B1}}{I_1}$	0
F_6	0	0	0	0	$\frac{A_{B1}}{C_{B1}}$	$\frac{\alpha_o}{h\beta_o}$	$\frac{-1}{C_{B1}}$
F_7	0	0	0	0	0	$\frac{-ka_{14}}{2(y_8-P_T)^{1/2}}$	1
F_8	0	0	$\frac{1}{C_{X2}}$	0	0	0	0
F_9	0	0	0	0	0	0	0
F_{10}	0	0	0	0	0	0	0
F_{11}	0	0	0	0	0	0	0
F_{12}	0	0	0	0	0	0	0
F_{13}	0	0	0	0	0	0	0
F_{14}	0	0	0	0	0	0	0
F_{15}	0	0	-1	0	0	0	0
F_{16}	0	0	0	0	0	0	0

P_{S2}	Q_{21}	Q_{25}	P_{A2}	v_2	P_{B2}	Q_{24}	Q_{S2}	P_{T1}
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	$\frac{ka_{15}}{2(y_1-y_{16})^{1/2}}$
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
$\frac{\alpha_o}{h\beta_o}$	$\frac{-1}{C_{S2}}$	$\frac{-1}{C_{S2}}$	0	0	0	0	0	0
$\frac{-ka_{21}}{(y_8-y_{11})^{1/2}}$	1	0	$\frac{ka_{21}}{(y_8-y_{11})^{1/2}}$	0	0	0	0	0
$\frac{-ka_{25}}{(y_8-P_T)^{1/2}}$	0	1	0	0	0	0	0	0
0	$\frac{1}{C_{A2}}$	0	$\frac{\alpha_o}{h\beta_o}$	$\frac{-A_{A2}}{C_{A2}}$	0	0	0	0
0	0	0	$\frac{A_{A2}}{I_2}$	$\frac{\alpha_o}{h\beta_o} \cdot \frac{B_2}{I_2}$	$\frac{-A_{B2}}{I_2}$	0	0	0
0	0	0	0	$\frac{A_{B2}}{C_{B2}}$	$\frac{\alpha_o}{h\beta_o}$	$\frac{-1}{C_{B2}}$	0	0
0	0	0	0	0	$\frac{-ka_{24}}{2(y_{13}-P_T)^{1/2}}$	1	0	0
0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	-1

Figure 14. PW Matrix for Two-Actuator Open-Center System

extremely large, and could be expected to adversely affect the condition of the matrix. An important conclusion from the qualitative analysis presented above, is that the efficiency of simulation depends not only on the differential equations in the system, but the algebraic equations as well. The use of a Newton-like iteration dictates that the step size be small enough to permit convergence to the 'correct' solution, within the prescribed number of steps.

The possibility of Gear's method becoming inefficient, when using a single mathematical representation, was foreseen in preliminary trials with DFASUB and, consequently, one of the areas proposed for investigation was that of model switching. Explicit integration methods normally permit models to be switched provided the order of the system is not altered. Since Gear's method is indifferent to the number of differential variables, i.e., the actual order of the system, it permits switching between dynamic and static models provided the criteria for switching are explicitly furnished. In fact, switching can be relatively easily accomplished by repeated calls to DFASUB by the main program and using a flag to indicate that the last set of values of the differential algebraic state should be used as the initial values after a switch.

An investigation of the two actuator open center system reveals, however, that when a_{15} or a_{25} is zero, the model for the relevant subsystem becomes simpler and the length of the differential algebraic state vector can be reduced by excluding the bypass flow, Q_{15} or Q_{25} as the case may be. On the conjecture that such a reduction in the unknown vector length could conceivably overcome the problem of an ill conditioned PW matrix, the main simulation program was modified to perform model switching with reduction in length of the differential algebraic vector. Originally, the switching criterion was the value of a_{15} (Subsystem #3 was kept inactive for the trials), i.e., a small value of a_{15} was used to signal the switch to the model in which Q_{15} was absent. However, it was found that if Q_{15} was, in fact, appreciable at the time of switching, the simulation was unsuccessful due to lack of convergence of the Newton iteration. When the switching criterion was changed to a combination of small a_{15} and Q_{15}, simulation proceeded very satisfactorily, as attested by the sample run shown in Figure 15.

Appendix D contains extracts from the FORTRAN listing of the main program to show the relative ease with which model switching can be performed. It is only necessary to:

(i) furnish suitable switching criteria

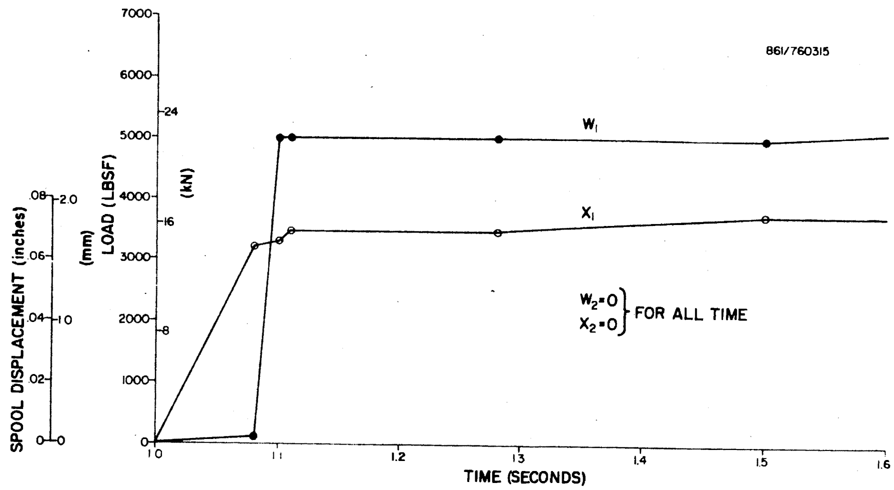
(ii) reconstitute the differential algebraic state vector for the large system if it is different from the old one

(iii) furnish initial conditions for the new differential algebraic state vector.

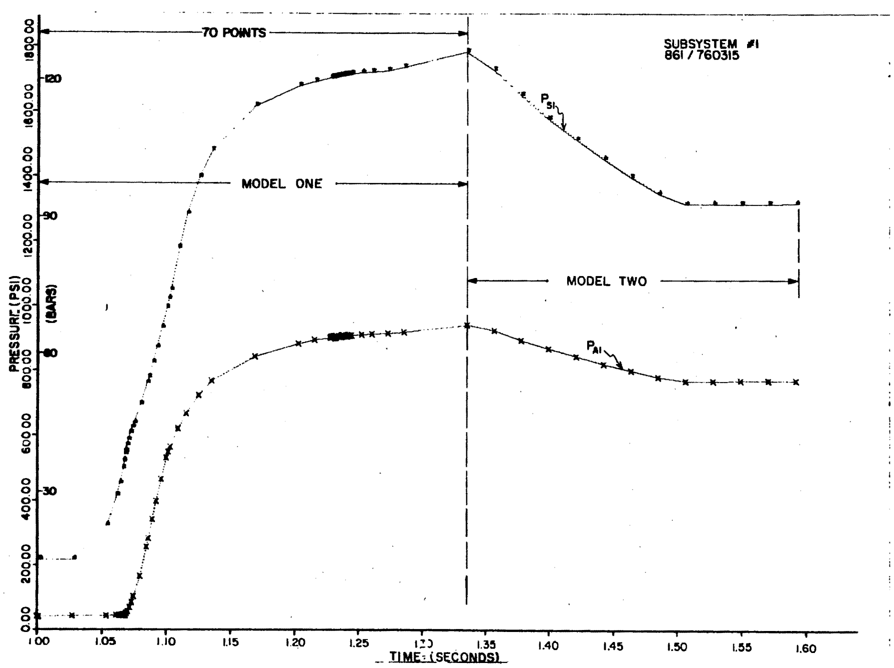
A subsequent call to DFASUB recommences trajectory simulation with a fresh set of integration parameters provided if necessary.

The above functions can be performed by a suitably coded director program which would effectively function like the supervisor in a hierarchical system.

This chapter has presented, as an example of model formulation and simulation of a large mobile hydraulic system, the digital simulation of a two actuator open-center system. The primary intention of the exercise was to demonstrate the feasibility and utility of the new approach, which relies on implicit representation of subsystems as well as the large

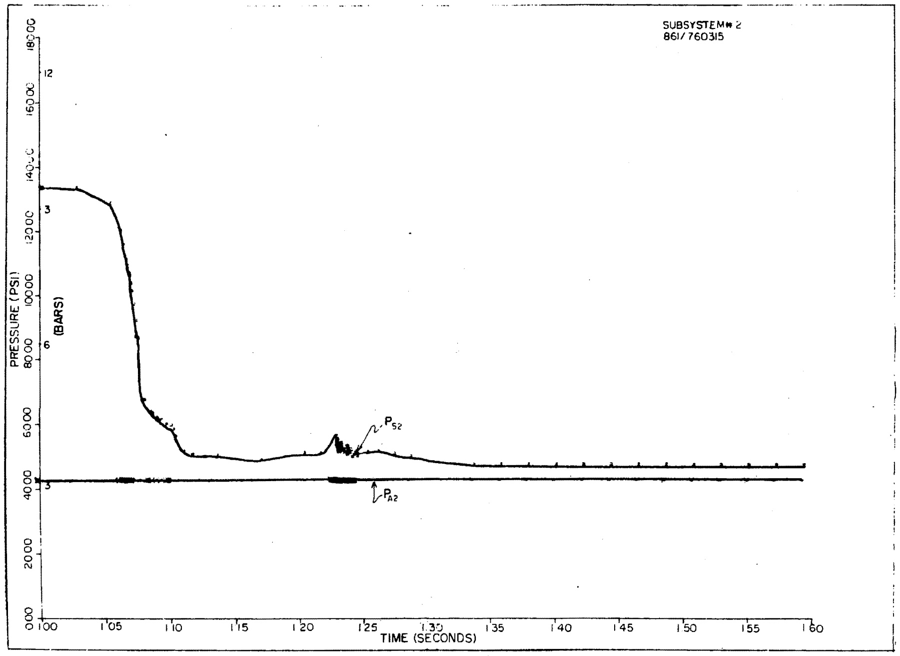


(a)

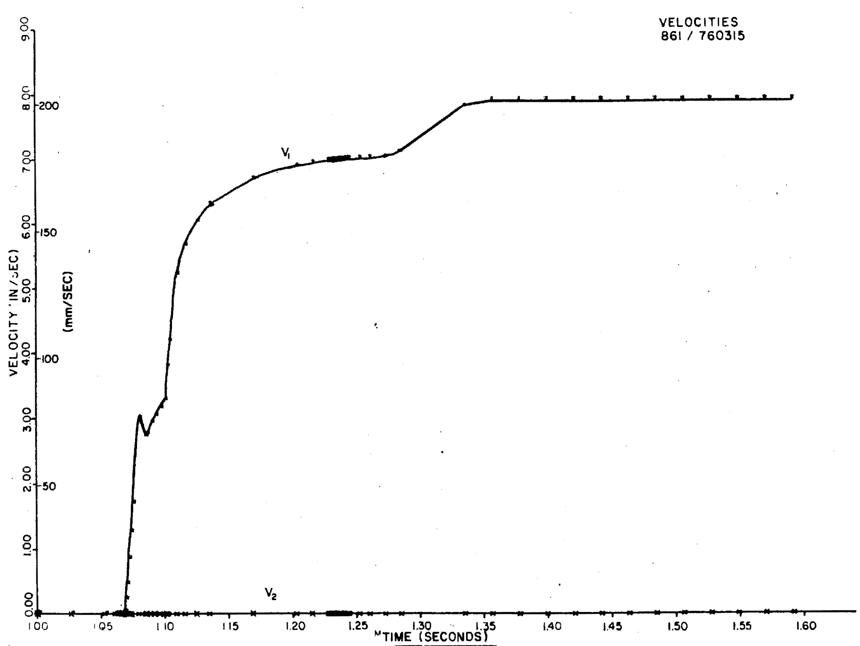


(b)

Figure 15. Simulation, With Same Inputs as Shown in Figure 13, But With Model Switching



(c)



(d)

Figure 15. (Continued)

system, and uses Gear's method for trajectory generation. The chapter also presents results on the qualitative behavior of the example system arrived at by a process of eigenanalysis of the large system represented in the implicit differential-algebraic form. Even though such qualitative analysis cannot be performed without incorporating the numerical values of system parameters and inputs, the results demonstrate that the class of systems under investigation can become stiff and, consequently, unamenable to simulation by explicit integration routines. The advantage of Gear's method, which relies on implicit integration, is in this respect obvious. Typical simulation runs have also been presented to show that hard constraints can be imposed on elements of the differential-algebraic state vector without necessarily disrupting the trajectory generation. Since the Gear method relies on a Newton-like iteration to solve a set of non-linear equations at each step in time, it is reasonable to expect problems when hard constraints on variables are imposed.

Perhaps one of the most important findings of this chapter is that model switching is not only easily done, but can be judiciously used, for example, to overcome the problem of an ill-conditioned PW matrix. Since Gear's method does not differentiate between algebraic and differential variables, it is seen that the new approach permits switching between dynamic and static models under the control of a suitably coded director program.

CHAPTER VI

CONCLUSIONS AND RECOMMENDATIONS

The analysis of large mobile hydraulic systems has been done, until recently, in a rather perfunctory manner. Fluid power system designers, on one hand, have rested content with performing 'worst' case analysis using simplistic models of the man-machine system, primarily for sizing components and ascertaining energy conversion efficiency. Most systems analysts have almost completely ignored the 'real-world' aspects of such systems, firstly due to their preoccupation with dynamic analysis to the exclusion of static performance, and secondly due to their belief in the efficacy of the network approach; namely, the decomposition of all physical systems to their basic elements before developing the system model. Where models have been developed for multiport components restrictions have been imposed on the manner of their interconnection (24, 25).

Much of the software written for the dynamic simulation of lumped parameter physical systems, relies on the network formulation of the system equations, whether it is done by the program user or generated by the computer. One of the main objectives of the research described herein was to indicate the limitations of the generalized network approach and, consequently, all the digital simulation software based thereon. These limitations necessitated the development of a new approach to the development of a unified scheme for simulating both static and dynamic behavior of large systems, synthesized from models of their subsystems.

The contributions of this research are as follows:

1. Postulation and proof of the Order Reduction Theorem, which states that when two subsystems, each expressed in the explicit state differential-algebraic form, are interconnected to form a large system, so that outputs are linearly related, the order of the large system is less than the sum of the orders of the subsystems.
2. Establishment of new canonical forms for representing subsystems and large system, which not only allow complete freedom in demarcating subsystem boundaries (and therefore arbitrary interconnection of subsystem models), but also explicitly depict the large system topology.
3. Discovery that order reduction in large hydraulic systems, brought about by interconnection of subsystem models, is nontrivial.
4. Formulation of a simulation algorithm for time-domain analysis of large mobile hydraulic systems.
5. Demonstration that large mobile hydraulic systems can display stiff behavior.
6. Discovery that the numerical integration method advanced by Gear for differential-algebraic equations is applicable even for systems having hard constraints on variables.
7. Development of an algorithm for eigenanalysis of systems described in the new canonical form, and demonstration of its utility for:

(i) qualitative appraisal of dynamic behavior,

(ii) establishment of stiffness of system.

8. Recognition, for the first time, that the step-size in implicit integration is dependent not only upon the dynamics of the system, but also the nature of the algebraic equations in the model.

9. Establishment that simulation of large systems by implicit integration can be substantially accelerated by appropriate switching between models of different orders.

Recommendations for Further Investigations

There are two broad areas where further investigations can be expected to yield valuable payoffs:

1. Mathematical Analysis: Exploration of the implicit differential-algebraic representation, in contrast to the explicit vector form which has formed the basis of most time-domain control theory can result in algorithms useful for optimal control and parameter identification. The concept of a differential-algebraic vector, which could possibly involve constraints on the differential algebraic state of the system would lead to more general theorems than those which presume the existence of an explicit state vector of known dimension. The concept of state constraints could be extended to the more general case of implicit inequality constraints.

2. Computer Software Development: This could focus attention on exploiting sparse matrix techniques for handling the

Jacobian of the differential algebraic system, as also the development of programs for optimal ordering of the set of equations describing the large system. In particular, characteristic features of parts of the matrix, e.g., that due to linearity and invariance of the topological constraint equations could be exploited to result in more efficient simulation.

BIBLIOGRAPHY

(1) Nightingale, J. M. "New Techniques for Hydraulic Servo Design." *Control Engineering*, Vol. 1, No. 1 (September, 1954), pp. 73-78.

(2) Seleno, A. A. "Hydraulic Servo Drives." *Machine Design*, Vol. 35, No. 24 (October 10, 1963), pp. 183-200.

(3) Taylor, L. D. "Jet Engine Control." *Applied Hydraulics*, Vol. 9, No. 5 (May, 1956), pp. 119-123.

(4) McConnell, G., and C. Wentworth. "Positioning With Hydraulic Cylinders." *Control Engineering*, Vol. 12, No. 3 (March, 1965), pp. 91-96.

(5) Alpay, S. A., and T. Mitchell. "Digital Simulation of a Hydraulic Control System." *Hydraulics and Pneumatics*, Vol. 24, No. 3 (March, 1971), pp. 81-83.

(6) Annual Report ER72-F-2, Ford Fluid Control Circuits, prepared for Ford Tractor Division. Stillwater, Oklahoma: Fluid Power Research Center, Oklahoma State University, 1971.

(7) Annual Report, Hydraulic Specification Program (Phase II), prepared for U. S. Army MERDC, AD 737 730. Stillwater, Oklahoma: Fluid Power Research Center, Oklahoma State University, 1971.

(8) Morris, H. C. "Digital Computer Simulation of Relief Valve Response." *Society of Automotive Engineers*, Paper No. 730445, 1973.

(9) Bowns, D., and J. Worton-Griffiths. "The Dynamic Characteristics of a Hydrostatic Transmission." *Proceedings I, Mechanical Engineers*, Vol. 186, No. 55 (1972), pp. 755-773.

(10) Hsu, J. C., and J. R. Hammerle. "Computer Aids Systems Analysis." *Hydraulics and Pneumatics*, Vol. 27, No. 8 (August, 1974), pp. 55-57.

(11) Goldenberg, K., and D. A. Collins. "Computer Optimization of Hydraulic Systems Pressure for Development and Qualification Testing." *Journal of Testing and Evaluation*, Vol. 2, No. 4 (July, 1974), pp. 229-234.

(12) Iyengar, S. K. R., and D. G. Miller. "Static Analysis of Directional Control Valves Using Analytical and Test Data," Paper Number P73-SP-3. Stillwater, Oklahoma: Basic Fluid Power Research Program, Fluid Power Research Center, Oklahoma State University, 1973.

(13) Smolina, M. I., et al. "Static Characteristics of Indirect Action Pressure-Control Valves." Russian Engineering Journal, Vol. LI, No. 9 (1971), pp. 66-72.

(14) Iyengar, S. K. R. "A Prognostic Simulation Technique for Fluid Power Systems." Society of Automotive Engineers, No. 750821 (1975).

(15) Shearer, J. L., A. T. Murphy, and H. H. Richardson. Introduction to System Dynamics. Reading, Mass.: Addison-Wesley, 1967.

(16) Sanford, R. S. Physical Networks. Englewood Cliffs, N. J.: Prentice-Hall, 1965.

(17) Sanford, R. "Combined Electrical Mechanical and Hydraulic Systems." Proceedings of the 16th Midwest Symposium on Circuit Theory, Vol. 2. Waterloo: Department of Systems Design, University of Waterloo, SIAM, and IEE, April 12-13, 1973.

(18) Rosenberg, R. C. "State Space Formulation for Bond Graph Models of Multiport Systems." Transactions of American Society of Mechanical Engineers, Journal of Dynamic Systems, Measurement and Control (March, 1971), pp. 34-40.

(19) Koenig, H. E., Y. Tokad, and H. K. Kesavan. Analysis of Discrete Physical Systems. New York: McGraw-Hill, 1967.

(20) Kuh, E. S., and R. A. Rohrer. "The State-Variable Approach to Network Analysis." Proceedings of Institution of Electrical and Electronic Engineers, Vol. 53, No. 7 (July, 1965), pp. 672-686.

(21) Tou, J. T. Modern Control Theory. New York: McGraw-Hill Book Co., 1964.

(22) Zadeh, L., and E. Polak. System Theory. New York: McGraw-Hill Book Co., 1961.

(23) Calahan, D. A. Computer-Aided Network Design. New York: McGraw-Hill Book Co., 1972.

(24) Aircraft Hydraulic System Transient Analysis (HYTRAN). Report MDC A3060, Vol. 2. Prepared under Aero-Propulsion Contract No. F33615-74-C-2016. St. Louis, Mo., 1975.

(25) Smith, C. K. "Digital Computer Simulation of Complex Hydraulic Systems Using Multiport Component Models." (Unpublished Ph.D. Dissertation, Oklahoma State University, Stillwater, 1975.)

(26) Iyengar, S. K. R. "Algebraic Constraints on State Variables-Their Physical Origin and Impact on Analysis and Simulation." Seventh Annual Pittsburgh Conference on Modeling and Simulation, April 26-27, 1976. To be published by the Instrument Society of America.

(27) Miller, D. G. "Development and Verification of Dynamic Thermal Models for Fluid Power Components and Systems." (Unpublished M.S. Thesis, Oklahoma State University, Stillwater, 1974.)

(28) Grove, W. E. Brief Numerical Methods. New Jersey: Prentice-Hall, Inc., 1966.

(29) Hamming, R. W. Numerical Methods for Scientists and Engineers. New York: McGraw-Hill Book Co., Inc., 1962.

(30) Ralston, A. A First Course in Numerical Analysis. New York: McGraw-Hill Book Co., Inc., 1966.

(31) Bowers, J. C., and S. R. Sedore. SCEPTRE: A Computer Program for Circuit and Systems Analysis. Englewood Cliffs, New Jersey: Prentice-Hall, 1971.

(32) Iyengar, S. K. R. "The Role of Modeling and Simulation in the Design and Appraisal of Fluid Power Systems." American Society of Mechanical Engineers Paper No. 76-DET-92 presented at the Design Engineering Technical Conference, Montreal, Quebec, Canada, Sept. 26-29, 1976.

(33) Bowers, J. C., J. E. O'Reilly, and G. A. Shaw. User's Manual for SUPER-SCEPTRE, AD 782 251, 1975.

(34) Trauboth, H., and N. Prasad. "MARSYAS - A Software System for Digital Simulation of Physical Systems." Proceedings of Spring Joint Computer Conference (AFIP), 1970, pp. 223-235.

(35) Desoer, C. A. Notes for a Second Course on Linear Systems. New York: Van Nostrand-Reinhold, 1970.

(36) Sage, A. P. Optimal Systems Control. Englewood Cliffs, New Jersey: Prentice-Hall, 1968.

(37) Wymore, A. W. A Mathematic Theory of Systems Engineering - The Elements. New York: Wiley, 1967.

(38) Iyengar, S. K. R. "Order Reduction in the Synthesis of Large-Scale Linear Systems," Paper P75-28, Annual Report No. 9. Stillwater: Basic Fluid Power Research Program, Fluid Power Research Center, Oklahoma State University, Stillwater, 1975.

(39) Lorenzo, C. F., and J. P. Riehl. "Applications of Symbolic Computing Methods to the Dynamic Analysis of Large Systems." Proceedings of International Federation of Automatic Control, Paper 27.1. Paris: 5th World Congress, June 12-17, 1972.

(40) Gear, C. W. Numerical Initial Valve Problems in Ordinary Differential Equations. Engelwood Cliffs, New Jersey: Prentice-Hall, Inc., 1971.

(41) Gear, C. W. "The Automatic Integration of Large Systems of Ordinary Differential Equations." Digest Record of the 1969 Joint Conference on Mathematical and Computer Aids to Design. Anaheim, California: Association for Computing Machinery/ Society for Industrial and Applied Mathematics/Institution of Electrical and Electronic Engineers, 1969.

(42) Blostein, M. L. "Recent Advances in Computer-Aided Analysis of Non-linear Networks." Network and Signal Theory. J. K. Skwirzynski and J. O. Scanlan (eds.). London: Peter Peregrenes, Ltd., 1972, pp. 97-104.

(43) Orlandea, M. A. Chace, and D. A. Calahan. "A Sparsity-Oriented Approach to the Dynamic Analysis and Design of Mechanical Systems - Par 1." American Society of Mechanical Engineers, Paper No. 76-DET-19. Montreal, Canada: Design Engineering Technical Conference, September 26-29, 1976.

(44) Rosenberg, R. C. "The Bond Graph as a Unified Data Base for Engineering System Design." American Society for Mechanical Engineers, Paper No. 74-DET-85, 1974.

(45) Brayton, R. K., F. G. Gustavson, and G. D. Hachtel. The Use of Variable-Order Variable-Step Backward Differentiation Methods for Nonlinear Electrical Networks. Systems, Networks and Computers International Conference, organized by the Institution of Electrical and Electronic Engineers, Oaxtepec, Mexico, 1971.

(46) Brayton, R. K., F. G. Gustavson, and G. D. Hachtel. "A New Efficient Algorithm for Solving Differential Algebraic Systems Using Implicit Backward Differentiation Formulas." Proceedings of the Institution of Electrical and Electronic Engineers, Vol. 60, No. 1 (January, 1972), pp. 98-108.

(47) Van Bokhoven, W. M. G. "Linear Implicit Differential Formulas of Variable Step and Order." Institution of Electrical and Electronic Engineers Transactions on Circuits and Systems, Vol. CAS-22, No. 22 (February, 1975), pp. 109-115.

(48) Branin, F. H., et al. "ECAP II - A New Electronic Circuit Analysis Program." Institution of Electrical and Electronic Engineers Journal of Solid-State Circuits, Vol. SC-6, No. 4 (Aug., 1971), pp. 146-166.

(49) Brown, R. L., and C. W. Gear. "Documentation for DFASUB - A Program for the Solution of Simultaneous Nonlinear Equations," UIUCDCS-R-73-575. Urbana, Illinois: Department of Computer Science, University of Illinois at Urbana-Champaign, 1973.

(50) Iyengar, S. K. R. Mathematical Modeling of Fluid Power Systems. Fluid Power Systems and Controls Conference, University of Wisconsin, Madison, May 24-26, 1976.

(51) Enright, W. H., T. E. Hull, and B. Lindberg. "Comparing Numerical Methods for Stiff Systems of O.D.E.'s." BIT, Vol. 15 (1975), pp. 10-48.

(52) Iyengar, S. K. R. "Qualitative Appraisal of Large Nonlinear Dynamic Systems Through Eigenanalysis," Paper No. P76-42. 10th Annual Fluid Power Research Conference. Stillwater: Fluid Power Research Center, Oklahoma State University, 1976.

(53) Iyengar, S. K. R. "Analysis of Multiple Actuator Open-Center Systems," Paper No. P76-38. 10th Annual Fluid Power Research Conference. Stillwater: Fluid Power Research Center, Oklahoma State University, 1976.

(54) McLachlan, E. K. Calculus of Several Variables. Belmont, Calif.: Brooks/Cole Pub. Co., 1968.

APPENDIX A

ORDER REDUCTION THEOREM

If two physical subsystems comprising a large system and represented by continuous and differentiable differential-algebraic equations are so connected that a scalar state-dependent output of one is linearly related to a scalar state-dependent output of the other, the order of the large system is less than the sum of the orders of the subsystems.

Proof:

Let the two subsystems be expressed in the following canonical form:

$$\mathrm{I}\quad\begin{cases} \dot{x}_1 = f_1(x_1, u_1, t) & \text{(A-1.1)} \\ y_1 = g_1(x_1, u_1, t) & \text{(A-1.2)} \end{cases}$$

$$\mathrm{II}\quad\begin{cases} \dot{x}_2 = f_2(x_2, u_2, t) & \text{(A-2.1)} \\ y_2 = g_2(x_2, u_2, t) & \text{(A-2.2)} \end{cases}$$

Since all functions are continuous and differentiable, they can be expanded in a Taylor's series about an operative point $x_1(0)$, $u_1(0)$ to obtain linearized models as follows:

$$\mathrm{I}\quad\begin{cases} \Delta\dot{x}_1 = f_1^1 \Delta x_1 \big|_{x_1(0),u_1(0)} + f_1^2 \Delta u_1 \big|_{x_1(0),u_1(0)} & \text{(A-3.1)} \\ \Delta y_1 = g_1^1 \Delta x_1 \big|_{x_1(0),u_1(0)} + g_1^2 \Delta u_2 \big|_{x_1(0),u_1(0)} & \text{(A-3.2)} \end{cases}$$

$$
\text{II} \quad \begin{cases} \Delta\dot{x}_2 = f_2^1 \Delta x_2 \big|_{x_2(0),u_2(0)} + f_2^2 \Delta u_2 \big|_{x_2(0),u_2(0)} & \text{(A-4.1)} \\ \Delta y_2 = g_2^1 \Delta x_2 \big|_{x_2(0),u_2(0)} + g_2^2 \Delta u_2 \big|_{x_2(0),u_2(0)} & \text{(A-4.2)} \end{cases}
$$

Let the two subsystems be connected so that the i'th element of Δy_1 is equal to the j'th element of Δy_2, and the k'th element of Δu_1 is equal to the l'th element of Δu_2. The case when the topological constraints involve a linear combination of these variables is a trivial extension.

Equating the two outputs gives

$$
g_{1i}^1 \Delta x_1 \big|_{x_1(0),u_1(0)} + g_{1i}^2 \Delta u_1 \big|_{x_1(0),u_1(0)} = g_{2j}^1 \Delta x_2 \big|_{x_2(0),u_2(0)} + g_{2j}^2 \Delta u_2 \big|_{x_2(0),u_2(0)}
$$

which implies that Δx_1 is not linearly independent of Δx_2. Consequently $(\Delta x_1 \vdots \Delta x_2)$ is not the state vector of the linearized large system. Therefore $(x_1 + \Delta x_1 \vdots x_2 + \Delta x_2)$ cannot be the state vector for the nonlinear differential algebraic system.

It may be noted that by equating the two inputs and eliminating Δy_{1i} and Δy_{2j} it is possible to establish the exact order and the exact state vector for the large system. Such a proof by construction would be usable, however, only for linear systems (38).

APPENDIX B

EIGENANALYSIS OF IMPLICIT DIFFERENTIAL ALGEBRAIC SYSTEMS OF EQUATIONS

The eigenvalues of the explicit differential algebraic system represented by

$$\dot{X} = f(X,U,t) \tag{B-1.1}$$

$$Y = f(X,U,t) \tag{B-1.2}$$

can be easily obtained by taking the first term in the Taylor's series expansion of f. It may be noted that Equation (B-1.2) is not needed for the eigenanalysis.

When a system is represented by a set of implicit differential algebraic equations, however, the coupling between the differential and algebraic equations makes eigenanalysis a little more involved.

Let the system be written in the following canonical form:

$$0 = f(\dot{X},X,Y,U) \tag{B-2.1}$$

$$0 = g(X,Y,U) \tag{B-2.2}$$

$$0 = h(Y,U,V) \tag{B-2.3}$$

It is shown in Chapter IV that this is the form for a large scale system, when the individual subsystem models are incorporated in f and g, and all the interconnection information is contained in h.

Since Equations (B-2.1), (B-2.2), and (B-2.3) are implicit, some algebraic manipulation is needed in order to establish the Jacobian of of the system. (The eigenvalues of the Jacobian are the eigenvalues of the system.) The Jacobian can be established by using the implicit function theorem (54).

Taking the differential of Equation (B-2.1) gives:

$$0 = \frac{\partial f}{\partial \dot{X}} \Delta \dot{X} + \frac{\partial f}{\partial X} \Delta X + \frac{\partial f}{\partial Y} \partial Y + \frac{\partial f}{\partial U} \Delta U$$

or

$$\Delta \dot{X} = - \left(\frac{\partial f}{\partial \dot{X}}\right)^{-1} \left[\frac{\partial f}{\partial X} \Delta X + \frac{\partial f}{\partial Y} \Delta Y + \frac{\partial f}{\partial U} \Delta U\right] \qquad \text{(B-3.1)}$$

Similarly from Equations (B-2.2) and (B-2.3) we get

$$\frac{\partial g}{\partial Y} \Delta Y + \frac{\partial g}{\partial U} \Delta U = - \frac{\partial g}{\partial X} \Delta X \qquad \text{(B-3.2)}$$

and

$$\frac{\partial h}{\partial Y} \Delta Y + \frac{\partial h}{\partial U} \Delta U = 0 \qquad \text{(B-3.3)}$$

From Equations (B-3.2) and (B-3.3) we get

$$\frac{\partial g}{\partial Y} \Delta Y + \frac{\partial g}{\partial U} \left(- \frac{\partial h}{\partial U}\right)^{-1} \frac{\partial h}{\partial Y} \Delta Y = - \frac{\partial g}{\partial X} \Delta X$$

or

$$\Delta Y = - \left(\frac{\partial \tilde{g}}{\partial Y}\right)^{-1} \left(\frac{\partial g}{\partial X}\right) \Delta X \qquad \text{(B-4)}$$

where

$$\left(\frac{\partial \tilde{g}}{\partial Y}\right) \triangleq \left[\frac{\partial g}{\partial Y} - \left(\frac{\partial g}{\partial U}\right)\left(\frac{\partial h}{\partial U}\right)^{-1} \left(\frac{\partial h}{\partial Y}\right)\right]$$

Consequently

$$\Delta U = \left(\frac{\partial h}{\partial U}\right)^{-1} \left(\frac{\partial h}{\partial Y}\right)\left(\frac{\partial \tilde{g}}{\partial Y}\right)^{-1} \left(\frac{\partial g}{\partial X}\right) \Delta X \qquad \text{(B-5)}$$

Using these values for ΔY and ΔU in Equation (B-2.1) gives

$$\Delta \dot{X} = - \left(\frac{\partial f}{\partial \dot{X}}\right)^{-1} \left[\frac{\partial f}{\partial X} - \left\{\frac{\partial f}{\partial Y} - \left(\frac{\partial f}{\partial U}\right)\left(\frac{\partial h}{\partial U}\right)^{-1} \left(\frac{\partial h}{\partial Y}\right)\right\}\left(\frac{\partial \tilde{g}}{\partial Y}\right)^{-1} \left(\frac{\partial g}{\partial X}\right)\right] \Delta X \qquad \text{(B-6)}$$

where the additional terms are irrelevant for eigenanalysis.

The coefficient of ΔX may be considered to be $\frac{\partial \tilde{f}}{\partial X}$ and is nothing but the desired Jacobian.

Once the numerical values of all the matrices in Equations (B-5) and (B-6) are established, it is fairly easy to evaluate and perform eigenanalysis on the system.

APPENDIX C

IMPLICIT INTEGRATION OF DIFFERENTIAL-ALGEBRAIC SYSTEMS

Comprehensive discussions of implicit integration have been given by Gear (40, 41), Blostein (42), Branin et al. (48), and others. What follows is a brief review of the Gear method as applied to differential algebraic systems expressed in the implicit form. The main intention is to supplement the discussion of its application to large mobile hydraulic systems, as detailed in Chapters IV and V. The terminology used by Brown and Gear (49) will be retained except as otherwise noted.

The differential algebraic equations used to describe the systems under consideration are considered to be written in the form

$$f\,(\underline{y}\,,\ \underline{y}\,,\ t) = 0 \qquad \text{(C-1)}$$

where $\underline{y}$ will be referred to as the differential algebraic state vector. The actual order of the dynamic system under consideration, and the establishment of the state vector are unnecessary in the implementation of Gear's method, and will only enter in the initial remarks on explicit integration.

The time domain simulation of systems represented in the form of Equation (C-1) can generally be posed as an initial value problem; i.e., considering a scalar variable y, given that $y(t_o) = y_o$ and Equation (C-1), establish y(t) for $t_o \leq t \leq t_f$.

Explicit numerical integration routines attempt to establish the value of y at a given time t_n, i.e., y_n, in terms of previous values of y, i.e., y_{n-1}, y_{n-2}, ..., y_{n-k-1} and $\dot{y}_{n-1}$, ..., $\dot{y}_{n-k-1}$. Symbolically,

$$y_n = f^*(y_{n-1}, y_{n-2}, \ldots, y_{n-k-1}, \dot{y}_{n-1}, \ldots, \dot{y}_{n-k-1}) \quad \text{(C-2)}$$

where f^* may be a composite function, and derivatives are obtained by using Equation (C-1). Consequently, y_n is explicitly determined by the previous values of y and its derivatives. Explicit numerical integration requires that y be the state of the dynamic system, and if the eigenvalues of the Jacobian of f are far apart, say more than two orders of magnitude, simulation can become very inefficient or fail (43).

The basic idea of implicit integration is to evaluate $\underline{y}_n$ and $\underline{y}_n$ simultaneously, using the differential algebraic equation and a suitable multistep formula for predicting both of them from previous values in time. The most commonly used multistep formula is:

$$y_n = -\sum_{i=1}^{k} a_i y_{n-1} - h \sum_{i=0}^{k} b_i \dot{y}_{n-i} \quad \text{(C-3)}$$

where a_i, b_i are appropriate coefficients, k is the order and h the current step size.

In Gear's method, $b_i = 0$ except for i=0, and the above equation is rewritten for the vector case as

$$h\underline{y}_n = -\frac{1}{\beta_o}(\alpha_0 \underline{y}_n + \alpha_1 \underline{y_{n-1}} + \ldots \alpha_k \underline{y_{n-k}} \quad \text{(C-4)}$$

It may be noted that $\underline{y}_n$ which appears on the right hand side in Equation (C-4), has not yet been computed. Equations (C-1) and (C-4) are now combined to give

$$F_n(\underline{y}_n) \triangleq f(\underline{y}_n\,,\ -\frac{\alpha_0}{h\beta_0}\underline{y}_n + \sum_n\,,\ t_n) = 0 \tag{D-5}$$

where

$$\sum_n \triangleq -\frac{1}{h\,\beta_0}[\alpha_1\underline{y_{n-1}} + \dots\ \alpha_k\underline{y_{n-k}}]$$

is known.

Gear's method uses the k'th order predictor

$$\underline{y}_{n,(0)} = h\bar{\beta}_1\ \underline{\dot{y}_{n-1}} + \bar{\alpha}_1\ \underline{y_{n-1}} + \dots\ \bar{\alpha}_k\underline{y_{n-k}} \tag{C-6}$$

to solve Equation (D-5) with a Newton-like iteration written as

$$\underline{y}_{n,(m+1)} = \underline{y}_{n,(m)} - J^{-1}[F_n(\underline{y}_{n,(m)})] \tag{C-7}$$

where $\underline{J} \triangleq \frac{\partial F_n}{\partial \underline{y}}$ and m is the iteration number.

In order to simplify the computer algorithms for error analysis and variation of order, Gear uses the Nordseick vector $\frac{Zn}{h}$, which for a scalar variable y_n is defined as

$$\underline{Zn} \triangleq [y_n,\ hy_n^{(1)},\ h^2y_n^{(2)}/2\ \dots\ h^k y_n^{(k)}/k!]^T \tag{C-8}$$

where $y_n^{(k)}$ is the k'th derivative of y at the n'th time step. It is shown by Gear that the Nordseick vector is uniquely determined by the quantities y_n, $\dot{y}_n$, y_{n-1}, ..., y_{n-k+1}.

In the computer implementation of Gear's algorithm for differential algebraic systems (49), a double dimensioned array Y(J,I) holds the values of the differential algebraic state vector and its derivatives, exactly as shown in Equation (C-8). The subroutine DIFFUN contains the model for the system in the implicit form, and utilizes an array DY(I) to retain the correction to the values in Y(J,I) before the Newton iteration. The Jacobian, J, which is stored in a double dimensioned array PW, may be explicitly written as follows:

$$J = \left[\frac{\partial f}{\partial \underline{y}} - \frac{\alpha_0}{h\beta_0}\frac{\partial f}{\partial \underline{\dot{y}}}\right] \qquad \text{(C-9)}$$

The Jacobian inversion is carried out by two subroutines DECOMP and SOLVE. These have been modified such that if any pivoting problems are faced and inversion is unsuccessful, an error message is returned and simulation stopped.

The advantages of Gear's method over conventional methods for differential algebraic systems are:

1. The form of the equations is more general, and no distinction need be made between differential and algebraic variables.
2. Large scale system models can be written in DIFFUN (or be c called by it) to explicitly exhibit subsystem models as well as topological constraints.
3. Algebraic constraints on state variables pose no special problems.
4. Stiff systems are handled efficiently, with no need for user interruption.
5. Step size and order of integration are changed automatically, as warranted by system dynamics and error criteria.
6. An estimate of the error is available at all times.
7. Derivatives of the differential and algebraic variables, up to the order of the current integration step, are available at all times.
8. If necessary, interpolation can be done to obtain variables within a step, by using a Taylor series expansion, the terms

of which have already been established during integration.

9. Developments in sparse matrix handling procedures can be exploited, to simulate large systems very efficiently.

10. Has the potential for model switching.

APPENDIX D

SELECTED COMPUTER PROGRAM LISTINGS

This appendix contains excerpts from the main program and key subprograms used to simulate the two actuator open center system. Since all of these modules were written primarily to verify the algorithms conceived for analyzing large mobile hydraulic systems, by using implicit representation and Gear's integration technique, they are only cursorily documented. A user-oriented package could be evolved from these programs by systematizing the input/output and interchange of information between the different modules. What follows is a brief explanation of the excerpts, which should be read along with the documentation for DFASUB (49) for a better appreciation of the program logic.

MAIN

The accompanying excerpt from the main program presents the significant variables. The comment statements are, for the most part, self-explanatory. YL is a vector of linear differential algebraic variables, which can be evaluated by DFASUB without resorting to the Newton iteration. This feature is not utilized in the present version of the program. The static numerical model for the two directional control valves is stored in a multi-dimensional array DORF, while the input time-histories (i.e., spool displacements and external loads) are stored in a common storage labeled INPTS.

```
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION Y(7,20),YL(2),SAVE(7,20),YLSV(2),PW(350)
     1,  G(16,2),T(20),DY(20),ERSV(20),ERROR(20),F1(20),EQN(20),
     2  VAR(20),YMAX(20),PTITLE(20)
      DIMENSION  Z(10)
      COMMON/JACOB/PPW(350)
      COMMON/PARM/COEFF,DPT,QS,CS1,CA1,CB1,CS2,CA2,CB2,XI1,XI2
      COMMON/CYLDT/AA1,AB1,AA2,AB2,B1,B2
      COMMON/INPTS/EXTLD(2,50),SPDISP(2,50),TINTP(50),NINTP
      COMMON/VLVS/DORF(2,5,2,20),NVLDT
      COMMON/INITVL/YY(7,20),YYL(2)
      COMMON/CRDR/NMAX,N,IPERM(20)
      COMMON/CPRINT/MPRNT,MPUN
      COMMON/DWRT/KKKK
      EQUIVALENCE (PW(1), PPW(1) )
      DATA Y,YL,SAVE,YLSV,G,DY,T,F1/376*0.0D+0/
C***********************************************************************
C
C     G          ARRAY OF GLOBAL VARIABLES, USED BY DFASUB
C     H          STEP SIZE, USED BY DFASUB
C     N          NUMBER OF EQUATIONS IN DIFFUN
C     T          ARRAY OF TIME-DEPENDENT VARIABLES (T(1)=TIME),FOR DFASUB
C     Y          ARRAY OF OUTPUT VARIABLES, PRINTED BY DFASUB
C     CS1        UPSTREAM CAPACITANCE, FOR SUBSYSTEM #1
C     CS2        UPSTREAM CAPACITANCE, FOR SUBSYSTEM #2
C     CA1        PORT A CAPACITANCE, FOR SUBSYSTEM #1
C     CA2        PORT A CAPACITANCE, FOR SUBSYSTEM #2
C     CB1        PORT B CAPACITANCE, FOR SUBSYSTEM #1
C     CB2        PORT B CAPACITANCE, FOR SUBSYSTEM #2
C     DY         ARRAY OF ERRORS,, EVALUATED IN DIFFUN, FOR DFASUB
C     F1         DUMMY ARRAY FOR DY, USED BY DFASUB
C     R1         AREA OF PISTON ROD, CYLINDER IN SUBSYSTEM #1
C     R2         AREA OF PISTON ROD, CYLINDER IN SUBSYSTEM #2
C     YL         ARRAY OF LINEAR VARIABLES, USED IN DFASUB
C     AA1        AREA OF HEAD END, CYLINDER IN SUBSYSTEM #1
C     AA2        AREA OF HEAD END, CYLINDER IN SUBSYSTEM #2
C     AB1        AREA OF ROD END, CYLINDER IN SUBSYSTEM #1
C     AB2        AREA OF ROD END, CYLINDER IN SUBSYSTEM #2
C     DPT        TANK PORT PRESSURE
C     EPS        ERROR CRITERION ARRAY, USED IN DFASUB
C     EQN        ARRAY USED BY DFASUB
C     VAR        ARRAY USED BY DFASUB
C     DORF       METERING ORIFICE DATA FOR OPEN CENTER VALVES
C     ERSV       ARRAY USED BY DFASUB
C     HMAX       MAX. ALLOWED STEP SIZE, USED BY DFASUB
C     HMIN       MIN. ALLOWED STEP SIZE, USED BY DFASUB
C     NUMT       NUMBER OF INPUT TIME FUNCTIONS
C     ROD1       ROD DIAMETER, CYLINDER FOR SUBSYSTEM #1
C     ROD2       ROD DIAMETER, CYLINDER FOR SUBSYSTEM #2
C     SAVE       ARRAY USED BY DFASUB
C     TEND       FINAL TIME, USED BY DFASUB
C     YLSV       ARRAY USED BY DFASUB
C     YMAX       ARRAY USED BY DFASUB
C     COEFF      ORIFICE FLOW CONVERSION FACTOR
```

```
C       CYLD1     DIAMETER OF CYLINDER, SUBSYSTEM #1
C       CYLD2     DIAMETER OF CYLINDER, SUBSYSTEM #2
C       ERROR     ARRAY USED BY DFASUB
C       EXTLD     ARRAY CONTAINING EXTERNAL LOADS, AS TIME FUNCTIONS
C       KFLAG     FLAG INDICATING COMPLETION CODE, OUTPUT OF DFASUB
C       NINTP     NUMBER OF INTERPOLATION POINTS IN INPUT TRAJECTORIES
C       NVLDT     NUMBER OF DATA POINTS FOR EACH METERING ORIFICE OF VALVES
C       TINTP     ARRAY CONTAINING TIME CORRESPONDING TO INPUT POINTS
C       JSTART    FLAG, USED BY  DFASUB
C       MPRNT     PRINT-OUT INTERVAL
C       MPUN      CARD PUNCH INTERVAL
C       KKKK IS THE FLAG FOR STORAGE ON DASD
C
C
C       IPERM IS VECTOR CONTAINING SUBSCRIPTS FOR  THE Y AND DY VECTORS
C       OF DIFFUN. WHEN MODEL IS CHANGED, IPERM WILL HAVE TO BE REDEFINED
C
C
C*********************************************************************
C       INITIALIZE IPERM
        DO 3000 I=1,20
3000    IPERM(I)= I
        KKKK= 250
        KKKK=1
        MPRNT=10
        MPRNT=5
        MPRNT=2
        MPRNT=1
        READ(5,1) PTITLE
1       FORMAT(20A4)
        WRITE(6,2) PTITLE
2       FORMAT(1H1,/////,5X,20A4,////)
        READ(5,9) NVLDT
C       NVLDT          NUMBER OF DATA POINTS FOR EACH METERING ORIFICE
        DO 3 I=1,2
        DO 3 J=1,5
C       INDEX I        REFERS TO VALVE
C       INDEX J        REFERS TO METERING ORIFICE
        READ(5,10)  (DORF(I,J,1,L),DORF(I,J,2,L), L=1,NVLDT)
3       CONTINUE
        DO 4 I=1,2
        DO 4 J=1,5
4       WRITE(6,999) (DORF(I,J,1,L), DORF(I,J,2,L), L=1,NVLDT)
999     FORMAT(//,'  METERING ORIFICE DATA',//  ,(1P8E13.3))
C       CYLD1          DIAMETER OF CYLINDER #1
C       CYLD2          DIAMETER OF CYLINDER #2
C       ROD1           ROD DIAMETER FOR CYLINDER #1
C       ROD2           ROD DIAMETER FOR CYLINDER #2
        READ(5,10) CYLD1, ROD1, CYLD2, ROD2
        AA1= 3.14159*CYLD1**2/4.0
        R1= 3.14159*ROD1**2/4.0
        AB1= AA1- R1
        AA2= 3.14159*CYLD2**2/4.0
        R2= 3.14159*ROD2**2/4.0
```

```
      T(1)= 1.0
      MODEL=1
      MAXDER=5
      M1= MMM
      CALL DFASUB(DY,EPS,EQN,ERROR,ERSV,
     +   F1,G,H,HMAX,HMIN,
     +   JSTART,KFLAG, M,MAXDER,M1,N,
     +   NL,PW,SAVE,T,NUMT,TEND,VAR,Y,YL,
     +   YLSV,YMAX,Z,MODEL)
      IF(MODEL.EQ.3) GO TO 300
      GO TO 9900
  300 CONTINUE
C
C     STATIC MODEL FOR SYSTEM WHEN A15=0.0
C
      STORE= Y(1,3)
      Y(1,3)= Y(1,4)
      Y(2,3)= Y(2,4)
      Y(1,4)= Y(1,5)
      Y(2,4)= Y(2,5)
      Y(1,5)= Y(1,6)
      Y(2,5)= Y(2,6)
      Y(1,6)= Y(1,7)
      Y(2,6)= Y(2,7)
      Y(1,7)= STORE
      JSTART=0
      N=6
      M=N
      NL=0
      M1= N
      IPERM(3)= 7
      IPERM(4)= 3
      IPERM(5)=4
      IPERM(6)=5
      IPERM(7)= 6
      HINV= 1.0/ H
      CALL DIFFUN(T,G,DY,Y,YL,HINV,MODEL)
      EPS=0.02
      CALL DFASUB(DY,EPS,EQN,ERROR,ERSV,
     1   F1,G,H,HMAX,HMIN,
     2   JSTART,KFLAG,M,MAXDER ,M1,N,
     3   NL,PW,SAVE,T,NUMT,TEND,VAR,Y,
     4   YL,YLSV,YMAX,Z,MODEL)
 9900 CONTINUE
      WRITE(6,998)  KFLAG
  998 FORMAT(1H1, ////,' ****** KFLAG=', I5)
      IF(KFLAG.EQ.-1) WRITE(6,9001)
 9001 FORMAT(//,10X,'THE ERROR TEST FAILED FOR H.GT.HMIN')
      IF(KFLAG.EQ.-2) WRITE(6,9002)
 9002 FORMAT(//,10X,'TOO MANY FLOATING-POINT EXCEPTIONS OCCURRED DURING
     +THE LAST STEP')
      IF(KFLAG.EQ.-3) WRITE(6,9003)
 9003 FORMAT(//,10X,'THE CORRECTOR FAILED TO CONVERGE FOR H.GT.HMIN')
      IF(KFLAG.EQ.-4) WRITE(6,9004)
 9004 FORMAT(//,10X,'THE CORRECTOR FAILED THREE TIMES WITH EVEN THE FIRS
     +T ORDER METHOD')
      WRITE(6,9999)
 9999 FORMAT(////////)
C123456789012345678901234567890123456789012345678901234567890123456789012
      STOP
      END
```

Line 272 indicates that the first model, (stored in MODL1) is to be utilized for the first call to DFASUB, in line 275. If DFASUB indicates that model switching is required, the flag MODEL will be set to the appropriate number before control is returned to the main program. If model switching is to be done, it is necessary to rearrange the differential algebraic state vector, so that only pertinent variables are included in the next call to DFASUB. This rearrangement is done in lines 286 through 295. Equation in line 297 indicates that the length of the diffeeential algebraic state vector is now six and not sixteen as was used in the first call. Line 309 calls DFASUB after the model switching, and the flag JSTART has to be set to zero so that DFASUB recognizes that a fresh start is to be made. Lines 315 through 325 present the error messages returned by DFASUB for abnormal termination of the simulation.

DFASUB

This subroutine is substantially the same as that presented by Brown and Gear (49). The changes made in order to tailor it for large scale systems is the extension of the argument list to include the explicit algebraic output variable vector Z and the model flag MODEL. The former is evaluated in ALGVAR, while subroutine PRINT returns a value in MFLAG, which is compared with the number furnished by the main program in MODEL. If MFLAG is different from MODEL, it indicates that model switching is necessary, and the value of MFLAG is returned to the main program via MODEL.

```
      SUBROUTINE DFASUB (DY,EPS,EQN,ERROR,ERSV,                        DFAS1010
     +   F1,G,H,HMAX,HMIN,                                             DFAS1020
     +   JSTART,KFLAG,M,MAXDER,M1,N,                                   DFAS1030
     + NL,PW,SAVE,T,NUMT,TEND,VAR,Y,
     +   YL,YLSV,YMAX,Z,MODEL)
C***************************************************************************** DFAS1070
C*                                                                      DFAS1080
C*    THE PARAMETERS TO THE SUBROUTINE DIFSUB HAVE                      DFAS1090
C*    THE FOLLOWING MEANINGS:                                           DFAS1100
C*                                                                      DFAS1110
C*       N        THE NUMBER OF VARIABLES.                              DFAS1120
C*       NL       THE NUMBER OF LINEAR VARIABLES                        DFAS1130
C*      (NY = N-NL  IS THE NUMBER OF VARIABLES WITH DERIVATIVES)        DFAS1140
C*       M1       THE NUMBER OF EQUATIONS TO TAKE PART IN THE ERROR TEST. DFAS1150
C*       TEND     END CRITERION                                         DFAS1160
C*       T        THE INDEPENDENT VARIABLE.                             DFAS1170
C*       Z        ALGEBRAIC VECTOR  CALCULATED BY SUBROUTINE ALGVAR
C*       G        AN ARRAY OF GLOBAL VARIABLES                          DFAS1180
C*       Y        A 7 BY NY ARRAY CONTAINING THE DEPENDENT VARIABLES    DFAS1190
C*                  AND THEIR SCALED DERIVATIVES. Y(J+1,I) CONTAINS     DFAS1200
C*                  THE J-TH DERIVATIVE OF Y(I) SCALED BY               DFAS1210
C*                  H**J/FACTORIAL(J), H THE CURRENT STEP SIZE.         DFAS1220
C*                   ON THE FIRST ENTRY, THE CALLER SUPPLIES            DFAS1230
C*                   Y(1,I) AND Y(2,I), UNSCALED. (IF THE CALL TO       DFAS1240
C*                   DIFSUB WAS PRECEDED BU A CALL TO DIFMF3, THE       DFAS1250
C*                   USER NEED NOT TOUCH Y AT ALL). THE PROGRAM         DFAS1260
C*                   WILL SCALE Y(2,I) BY H. ON ANY SUBSEQUENT          DFAS1270
C*                   ENTRY, THE PROGRAM ASSUMES THAT THE Y VALUES       DFAS1280
C*                   HAVE NOT BEEN CHANGED SINCE THE LAST EXIT          DFAS1290
C*                   FROM DIFSUB, AND WILL SCALE THESE VALUES IF        DFAS1300
C*                  THE CALLER HAS CHANGED H.                           DFAS1310
C*                     IF IT IS DESIRED TO INTERPOLATE TC NON-MESH      DFAS1330
C*                   POINTS THESE VALUES CAN BE USED. IF THE CURRENT    DFAS1340
C*                  STEP SIZE IS H AND THE VALUE AT T+E IS NEEDED,      DFAS1350
C*                  FORM S = E/H AND THE COMPUTE                        DFAS1360
C*                                   NQ                                 DFAS1370
C*                     Y(I)(T+E) = SUM Y(J+1,I)*S**J                    DFAS1380
C*                                   J=0                                DFAS1390
C*                CALLER MUST SUPPLY VALUES FOR THESE VARIABLES.        DFAS1410
C*       SAVE     AN ARRAY OF LENGTH AT LEAST 7*N.                      DFAS1420
C*       H        THE STEP SIZE TO BE ATTEMPTED ON THE NEXT STEP.       DFAS1430
C*                H MAY BE ADJUSTED UP OR DOWN BY THE USER DOES         DFAS1440
C*                IN ORDER TO ACHEIVE AN ECONOMICAL INTEGRATION.        DFAS1450
C*                HOWEVER, IF THE H PROVIDED BY THE USER DOES           DFAS1460
C*                NOT CAUSE A LARGER ERROR THAN REQUESTED, THAN         DFAS1470
C*                WILL BE USED. TO SAVE COMPUTER TIME, THE USER IS      DFAS1480
C*                ADVISED TO USE A FAIRLY SMALL STEP FOR THE FIRST      DFAS1490
C*              CALL. IT WILL BE AUTOMATICALLY INCREASED LATER.         DFAS1500
C*       HMIN     THE MINIMUM STEP SIZE THAT WILL BE USED FOR THE       DFAS1510
C*                INTEGRATION. NOTE THAT ON STARTING THIS MUST BE       DFAS1520
C*                MUCH SMALLER THAN THE AVERAGE H EXPECTED SINCE        DFAS1530
C*                A FIRST ORDER METHOD IS USED INITIALLY.               DFAS1540
C*   HMAX      THE MAXIMUM ALLOWABLE STEP SIZE ,                        DFAS1550
C*   EPS       THE ERROR TEST CONSTANT. SIGLE STEP ERROR ESTIMATES      DFAS1560
```

```
      C*              DIVIDED BY YMAX(I) MUST BE LESS THAN THIS                  DFAS1570
      C*              IN THE EUCLIDEAN NORM. THE STEP AND/OR ORDER IS            DFAS1580
      C*              ADJUSTED TO ACHEIVE THIS.                                  DFAS1590
      C*   YMAX       A VECTOR OF LENGTH NY WHICH CONTAINS THE MAXIMUM           DFAS1600
      C*              OF EACH Y SEEN SO FAR. ON THE FIRST CALL, THESE            DFAS1610
      C*              WILL BE INITIALIZED AS YMAX(K) = MAX(1,|Y(1,I)|)           DFAS1620
      C*   ERROR      A VECTOR OF LENGTH NY.                                     DFAS1630
      C*   KFLAG      A COMPLETION CODE WITH THE FOLLOWING MEANINGS:             DFAS1640
      C*                    +1  THE INTEGRATION WAS SUCCESSFUL.                  DFAS1650
      C*                                -1  THE ERROR TEST FAILED FOR H > HMIN.  DFAS1660
      C*                                -3  THE CORRECTOR FAILED TO CONVERGE FOR DFAS1670
      C*                                      H > HMIN.                          DFAS1680
      C*                                -2  TOO MANY FLOATING-POINT EXCEPTIONS   DFAS1690
      C*                                      OCCURRED DURING LAST STEP.         DFAS1700
      C*                                -4  THE CORRECTOR FAILED THREE TIMES WITH DFAS1710
      C*                                      EVEN THE FIRST-ORDER METHOD.       DFAS1720
      C*   JSTART   AN INPUT INDICATOR WITH THE FOLLOWING MEANINGS:              DFAS1730
      C*                                0  PERFORM THE FIRST STEP. THE ROUTINE   DFAS1740
      C*                                      INITIALIZES ITSELF, SCALES THE DERIV- DFAS1750
      C*                                   ATIVES IN Y(2,I) AND THEN PERFORMS    DFAS1760
      C*                                   THE INTEGRATION UNTIL T > TEND.       DFAS1770
      C*                                   ANY SUBSEQUENT CALLS SHOULD BE MADE   DFAS1780
      C*                                   WITH JSTART = 1.                      DFAS1790
      C*                                1  CONTINUE FROM  THE LAST STEP, INTEGRATING DFAS1800
      C*                                   UNTIL T > TEND.                       DFAS1810
      C*                          JSTART IS SET TO NQ, THE CURRENT ORDER OF      DFAS1820
      C*                                   THE METHOD, AT EXIT.                  DFAS1830
      C*   MAXDER THE MAXIMUM DERIVATIVE THAT SHOULD BE USED IN THE              DFAS1840
      C*                                      METHOD. IT MUST NOT EXCEED 6.      DFAS1850
      C*   PW       A VECTOR OF LENGTH N**2+20 (REAL*4),                         DFAS1860
      C*          GENERATED BY MATSET AND USED BY MATINV,MATMUL.                 DFAS1870
      C YLSV    A VECTOR OF LENGTH NL.                                           DFAS1880
      C*   DY       A VECTOR OF LENGTH M, OUTPUT OF DIFFUN.                      DFAS1890
      C*   ERSV     A VECTOR OF LENGTH NY.                                       DFAS1900
      C*   F1       A VECTOR OF LENGTH N, OUTPUT OF MATMUL.                      DFAS1910
      C*   EWN,VAR  VECTORS USED BY MATSET.                                      DFAS1920
      C
      C     MODEL IS THE U
      C*    MODEL      INPUT FROM CALLING PROGRAM WHICH INDICATES THE
      C*               SYSTEM MODEL TO BE USED
      C*    MFLAG      FLAG FROM VARIOUS SUBROUTINES TO INDICATE WHEN MODELS
      C*               NEED TO BE SWITCHED
      C*               WHEN MODEL.NE.MFLAG CONTROL IS RETURNED TO CALLING
      C*               PROGRAM WITH VALUE OF MFLAG IN MODEL
      C*
      C*                                                                         DFAS1930
      C*****************************************************************************
            IMPLICIT REAL*8 (A-H,O-Z)                                            DFAS1060
            REAL*4 AMAX1
            DIMENSION Z(10)
            DIMENSION T(1),G(1),Y(7,1),YL(1),SAVE(7,1),YMAX(1)                   DFAS1070
            DIMENSION ERROR(1),PW(1),YLSV(1),DY(1),ERSV(1)                       DFAS1080
            DIMENSION F1(1),EQN(1),VAR(1),A(7),PERTST(6,3)                       DFAS1090
            DATA PERTST /4.0,9.0,16.0,25.0,36.0,49.0,9.0,16.0,                   DFAS1100
```

DIFFUN

In the original version of DFASUB (49) this subroutine contained the model for the differential algebraic system. In the present version of the program, this subroutine functions as a director program, merely calling MODL1 and MODL2– subprograms which store the alternative models. This subroutine is called by DFASUB and depending on the value of MODEL calls the appropriate model subprogram.

MODL1

This subroutine contains the model for the two actuator open center system, in the form required by DFASUB, when a_{15} is non-zero. The model corresponds to the equations presented in Chapter V. It is seen that the models for the two actuator subsystems and the topological constraints are explicitly presented so as to reflect the large scale structure of the system.

MODL2

This subroutine contains the model for the situation in which both a_{15} and Q_{15} are nearly zero. The length of the differential algebraic vector Y is only six, as compared to sixteen in MODL1. This is because the closure of a_{15} and the absence of any inputs to actuator number two degenerates the large system to a single actuator system, and consequently no topological constraints need be explicitly shown.

```
FORTRAN IV G LEVEL  21                DIFFUN            DATE = 76221           21/36/18

                  SUBROUTINE DIFFUN(T,G,DY,Y,YL,HINV,MODEL)
                  IMPLICIT REAL*8(A-H,O-Z)
                  DIMENSION G(1,1),DY(16),Y(7,16),YL(2),T(2),Z(10)
                  COMMON/PARM/COEFF,DPT,QS,CS1,CA1,CB1,CS2,CA2,CB2,XI1,XI2          DIFF1030
                  COMMON/CYLDT/AA1,AB1,AA2,AB2,B1,B2
                  COMMON/INITVL/YY(7,20),YYL(2)
                  COMMON/ORDR/NMAX,N,IPERM(20)
                  GO TO  (100,200,200,200) , MODEL
           100    CALL MODL1(T,G,DY,Y,YL,HINV)
                  RETURN
           200    CALL MODL2(T,G,DY,Y,YL,HINV)
                  RETURN
                  END
```

```
               ABDPSA= DABS( DPSA1)                                             DIFF2040
               SIGN11= DPSA1/ABDPSA                                             DIFF2050
               Q11= COEFF*T(6)*DSQRT(ABDPSA)*SIGN11                             DIFF2060
               GO TO 11                                                         DIFF2070
      10       Q11= 0.0                                                         DIFF2080
      11       CONTINUE                                                         DIFF2090
               DPST1= Y(1,1) - Y(1,16)                                          DIFF2100
               IF(DPST1.EQ.0.0) GO TO 20                                        DIFF2110
               ABDPS1= DABS(DPST1)                                              DIFF2120
               SIGN15= DPST1/ABDPS1                                             DIFF2130
               Q15= COEFF*T(8)*DSQRT(ABDPS1)*SIGN15                             DIFF2140
               GO TO 21                                                         DIFF2150
      20       Q15= 0.0                                                         DIFF2160
      21       CONTINUE                                                         DIFF2170
               DY(2)= -Q11 + Y(1,2)                                             DIFF2180
               DY(3)= -Q15 + Y(1,3)                                             DIFF2190
               DY(4)= -Y(2,4)*HINV +(Y(1,2)-Y(1,5)*AA1)/CA1
               DY(5)= -Y(2,5)*HINV+(Y(1,4)*AA1-Y(1,6)*AB1-T(3)-B1*Y(1,5))/XI1
               DY(6)= -Y(2,6)*HINV+(-Y(1,7)+Y(1,5)*AB1)/CB1
               DPBT1= Y(1,6) - DPT                                              DIFF2230
               IF(DPBT1.EQ.0.0) GO TO 30                                        DIFF2240
               ABDPB1= DABS(DPBT1)                                              DIFF2250
               SIGN14= DPBT1/ABDPB1                                             DIFF2260
               Q14= COEFF*T(7)*DSQRT(ABDPB1)*SIGN14                             DIFF2270
               GO TO 31                                                         DIFF2280
      30       Q14=0.0                                                          DIFF2290
      31       CONTINUE                                                         DIFF2300
               DY(7)= -Q14 + Y(1,7)
      C        EQUATIONS FOR SUBSYSTEM #2                                       DIFF2310
               QS2= Y(1,3)                                                      DIFF2320
               DY(8)= -Y(2,8)*HINV+(QS2-Y(1,9)-Y(1,10))/CS2
               DPSA2= Y(1,8) - Y(1,11)                                          DIFF2340
               IF (DPSA2.EQ.0.0) GO TC 40                                       DIFF2350
               ABDPS2= DABS(DPSA2)                                              DIFF2360
               SIGN21= DPSA2/ABDPS2                                             DIFF2370
               Q21= CCEFF*T(9)*DSQRT(ABDPS2)*SIGN21                             DIFF2380
               GO TO 41                                                         DIFF2390
      40       Q21= 0.0                                                         DIFF2400
      41       CONTINUE                                                         DIFF2410
               DY(9)= -Q21 + Y(1,9)                                             DIFF2420
               DPST2= Y(1,8) -DPT                                               DIFF2430
               IF(DPST2.EQ.0.0) GO TO 50                                        DIFF2440
               ABDPT2= DABS(DPST2)                                              DIFF2450
               SIGN25= DPST2/ABDPT2                                             DIFF2460
               Q25= COEFF*T(11)*DSQRT(ABDPT2)*SIGN25                            DIFF2470
               GO TO 51                                                         DIFF2480
      50       Q25=0.0                                                          DIFF2490
      51       CONTINUE                                                         DIFF2500
               DY(10)= -Q25 + Y(1,10)                                           DIFF2510
               DY(11)= -Y(2,11)*HINV+(Y(1,9)-Y(1,12)*AA2)/CA2
               DY(12)= -Y(2,12)*HINV+(Y(1,11)*AA2-Y(1,13)*AB2-T(5)-B2*Y(1,12))/
     1           XI2
               DPBT2= Y(1,13) - DPT                                             DIFF2540
               IF(DPBT2.EQ.0.0) GO TO 60                                        DIFF2550
               ABDPB2= DABS(DPBT2)                                              DIFF2560
               SIGN24= DPBT2/ABDPB2                                             DIFF2570
               Q24= COEFF*T(10)*DSQRT(ABDPB2)*SIGN24                            DIFF2580
               GO TC 61                                                         DIFF2590
```

```
                                                                        DIFF2590
 60     Q24=0.0                                                         DIFF2600
 61     CONTINUE                                                        DIFF2610
        DY(14)= -Q24 + Y(1,14)                                          DIFF2620
        DY(13)= -Y(2,13)*HINV+(-Y(1,14)+Y(1,12)*AB2)/CB2
C       EQUATIFNS FCR TOPOLOGICAL CONSTRAINTS                           DIFF2640
        DY(15)= -QS2 + Y(1,15)                                          DIFF2650
        DY(16)= -Y(1,16) + Y(1,8)                                       DIFF2660
C                                                                       DIFF2670
C                                                                       DIFF2680
C23456789012345678901234567890123456789012345678901234567890123456789012DFAS1000
        RETURN                                                          DIFF2690
        END
```

```
FORTRAN IV G LEVEL  21                MODL2              DATE = 76221            21/36/18

 0001              SUBROUTINE MODL2(T,G,DY,Y,YL,HINV)
 0002              IMPLICIT REAL*8(A-H,O-Z)
 0003              DIMENSION G(1,1),DY(16),Y(7,16),YL(2),T(2),Z(10)
 0004              COMMON/ORDR/NMAX,N,IPERM(20)
 0005              COMMON/PARM/COEFF,DPT,QS,CS1,CA1,CB1,CS2,CA2,CB2,XI1,XI2             D
 0006              COMMON/CYLDT/AA1,AB1,AA2,AB2,B1,B2
 0007              COMMON/INITVL/YY(7,20),YYL(2)
 0008               COMMON/MODL/MFLAG
             C*************************************************************************
             C
             C      EQUATIONS DESCRIBING TWO ACTUATOR OPEN CENTER SYSTEM WITH
             C      VALVE SPOOL FULLY EXTENDED
             C
             C      Y(1,1)                SUPPLY PRESSURE      (PS1)
             C      Y(1,2)                ORIFICE #1 FLOW      (Q11)
             C      Y(1,3)                PORT A PRESSURE      (PA1)
             C      Y(1,4)                ACTUATOR VELOCITY    (V1)
             C      Y(1,5)                PORT B PRESSURE      (PB1)
             C      Y(1,6)                ORIFICE #4 FLOW      (Q14)
             C
             C      ALL VARIABLES FOR #2 ACTUATOR ARE INVARIANT
             C
             C      HARD CONSTRAINTS ON STATE VARIABLES
             C
 0009              IF(Y(1,1).LT.0.0D+0)  Y(1,1)= 0.0
 0010              IF(Y(1,3).LT.0.0D+0)  Y(1,3)= 0.0
 0011              IF(Y(1,5).LT.0.0D+0)  Y(1,5)= 0.0
             C
             C
 0012              QS1= QS
 0013              QS1= QS
 0014              DY(1)= QS1 - Y(1,2)
 0015              DPSA1= Y(1,1) - Y(1,3)
 0016              IF (DPSA1.EQ.0.0D+0)  GO TO 10
 0017              ABDPSA= DABS(DPSA1)
 0018              SIGN11= ABDPSA/DPSA1
 0019              Q11= COEFF*T(6)*DSQRT(ABDPSA)*SIGN11
 0020              GO TO 11
 0021        10    Q11= 0.0
 0022        11    CONTINUE
 0023              DY(2)= -Q11 + Y(1,2)
 0024              DY(3)= Y(1,2) - Y(1,4)*AA1
 0025              DY(4)=  Y(1,3)*AA1 - Y(1,5)*AB1- T(3)-B1*Y(1,4)
 0026              DY(5)= Y(1,6)- Y(1,4)*AB1
 0027              DPBT1= Y(1,5) -DPT
 0028              IF(DPBT1.EQ.0.0D+0) GO TO 30
 0029              ABDPB1= DABS(DPBT1)
 0030              SIGN14= ABDPB1/DPBT1
 0031              Q14= COEFF*T(7)*DSQRT(ABDPB1)*SIGN14
 0032              GO TO 31
 0033        30    Q14=0.0
 0034        31    CONTINUE
 0035              DY(6)= -Q14 + Y(1,6)
             C
 0036              RETURN
 0037              END
```

APPENDIX E

NUMERICAL VALUES FOR PARAMETERS IN EXAMPLE SYSTEM

This appendix tabulates all the physical variables and parameters of the open-center hydraulic system which was analyzed as an example system in Chapter V. Quantities which can vary in the course of a trajectory simulation, i.e., inputs and outputs, do not have any numerical value assigned to them in the table.

QUANTITY	NAME		NUMERICAL VALUE	UNITS
	ALGEBRAIC	COMPUTER		
ORIFICE CONSTANT	k_o	COEFF	104.284	
METERING AREA, ORIFICE NO. 1 VALVE, SUBSYSTEM NO. 2	a_{11}	A_{11}		m^2
METERING AREA, ORIFICE NO. 4 VALVE, SUBSYSTEM NO. 2	a_{14}	A_{14}		m^2
METERING AREA, ORIFICE NO. 5 VALVE, SUBSYSTEM NO. 2	a_{15}	A_{15}		m^2
ACTUATOR AREA, HEAD SIDE SUBSYSTEM NO. 2	A_{A1}	AA1	10.26×10^{-3}	m^2
ACTUATOR AREA, ROD SIDE SUBSYSTEM NO. 2	A_{B1}	AB1	8.839×10^{-3}	m^2
CAPACITANCE, SUPPLY LINE, SUBSYSTEM NO. 2	C_{S1}	CS1	534.2×10^{-15}	m^5/N
CAPACITANCE, LINE FROM PORT 'A' OF VALVE, SUBSYSTEM NO. 2	C_{A1}	CA1	2.374×10^{-15}	m^5/N
CAPACITANCE, LINE FROM PORT 'B' OF VALVE, SUBSYSTEM NO. 2	C_{B1}	CB1	4.75×10^{-15}	m^5/N
INERTIA OF MOVING PARTS IN ACTUATOR SUBSYSTEM NO. 2	I_1	X11	17.53	kg
DRAG RESISTANCE OF ACTUATOR SUBSYSTEM NO. 2	B_1	DRAG 1	5258.000	NS/m
TANK PORT PRESSURE	P_T	DPT	3.45	bars
ACTUATOR VELOCITY, SUBSYSTEM NO. 2	v_1	V1		m/S
ACTUATOR VELOCITY, SUBSYSTEM No. 3	v_2	V2		m/S
SPOOL DISPLACEMENT, VALVE IN SUBSYSTEM NO. 2	x_1	SPOOL 1		m

QUANTITY	NAME ALGEBRAIC	NAME COMPUTER	NUMERICAL VALUE	UNITS
SPOOL DISPLACEMENT, VALVE IN SUBSYSTEM NO. 3	x_2	SPOOL 2		m
SUPPLY FLOW FROM SUBSYSTEM NO. 1	O_{S1}	OS1	2.081×10^{-3}	m^3/sec
SUPPLY PRESSURE, SUBSYSTEM NO. 2	P_{s1}	DPS1		bars
PORT 'A' PRESSURE, SUBSYSTEM NO. 2	P_{A1}	DPA1		bars
PORT 'B' PRESSURE, SUBSYSTEM NO. 2	P_{B1}	DPB1		bars
BYPASS PORT PRESSURE, SUBSYSTEM NO. 2	P_{T1}	DPT1		bars
SUPPLY PRESSURE, SUBSYSTEM NO. 3	P_{S2}	DPS2		bars
PORT 'A' PRESSURE, SUBSYSTEM NO. 3	P_{A2}	DPA2		bars
PORT 'B' PRESSURE, SUBSYSTEM NO. 3	P_{B2}	DPB2		bars
EXTERNAL LOAD, SUB-SYSTEM NO. 3	w_1	W1		N
METERING AREA, ORIFICE NO. 1 VALVE, SUB-SYSTEM NO. 3	a_{21}	A21		m^2
METERING AREA, ORIFICE NO. 4 VALVE, SUB-SYSTEM NO. 3	a_{24}	A24		m^2
METERING AREA, ORIFICE NO. 5 VALVE, SUB-SYSTEM NO. 3	a_{25}	A25		m^2
ACTUATOR AREA, HEAD SIDE SUBSYSTEM NO. 3	A_{A2}	AA2	10.26×10^{-3}	m^2
ACTUATOR AREA, ROD SIDE SUBSYSTEM NO. 3	B_{B2}	AB2	8.839×10^{-3}	m^2

QUANTITY	NAME		NUMERICAL VALUE	UNITS
	ALGEBRAIC	COMPUTER		
CAPACITANCE, SUPPLY LINE SUBSYSTEM NO. 3	C_{S2}	CS2	23.74×10^{-15}	m^5/N
CAPACITANCE, LINE FROM PORT 'A' OF VALVE, SUBSYSTEM NO. 3	C_{A2}	CA2	2.374×10^{-15}	m^5/N
CAPACITANCE, LINE FROM PORT 'B' OF VALVE, SUBSYSTEM NO. 3	C_{B2}	CB2	4.75×10^{-15}	m^5/N
INERTIA OF MOVING PARTS IN ACTUATOR, SUBSYSTEM NO. 3	I_2	X12	17.53	kg
DRAG RESISTANCE OF ACTUATOR SUBSYSTEM NO. 3	B_1	DRAG 1	5258.000	NS/m

VITA

Seshadri Kalyan Ram Iyengar

Candidate for the Degree of

Doctor of Philosophy

Thesis: DEVELOPMENT OF A UNIFIED APPROACH TO THE SIMULATION OF STATIC AND DYNAMIC BEHAVIOR OF LARGE MOBILE HYDRAULIC SYSTEMS

Major Field: Mechanical Engineering

Biographical:

Personal Data: Born in Bhadravathi, India, September 14, 1942, the son of Mr. and Mrs. M. C. Seshadri Iyengar.

Education: Passed the Overseas School Certificate Examination of the University of Cambridge (U.K.) in December, 1956; attended St. Aloysius College, Jabalpur, India, and passed the Intermediate Examination in Science in March, 1959; graduated from the University of Jabalpur, India, with the Degree of Bachelor of Engineering (Honours) in 1963; received the Master of Science degree at Oklahoma State University in July, 1972; completed the requirements for the Doctor of Philosophy degree in May, 1977.

Professional Experience: Graduate Engineer, Rourkela Steel Plant, Division of Hindustan Steel Limited, India, from August, 1963, to February, 1965; commissioning and subsequently maintenance engineer, Alloy Steels Plant, Division of Hindustan Steel Limited, from March, 1965, to August, 1970; research assistant at the Fluid Power Research Center, Oklahoma State University, September, 1970 to June, 1975; Research Engineer at the Fluid Power Research Center, Oklahoma State University, July, 1975, to date; Industrial consultant 1974 to date.

Professional Affiliations: American Society of Mechanical Engineers, Society of Automotive Engineers, American Management Associations, National Fluid Power Association, Phi Kappa Phi.