This is to certify that the

thesis entitled

A PARALLEL COMPUTATION METHOD
FOR DYNAMIC SYSTEMS
WITH COUPLED NONLINEAR DISSIPATION

presented by

Tong Zhou

has been accepted towards fulfillment
of the requirements for

Masters degree in Science

R C Rosenberg
Major professor

Date May 17, 1984

O-7639

A PARALLEL COMPUTATION METHOD FOR DYNAMIC SYSTEMS WITH COUPLED NONLINEAR DISSIPATION

By

Tong Zhou

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Mechanical Engineering

1984

ABSTRACT

A PARALLEL COMPUTATION METHOD FOR DYNAMIC SYSTEMS WITH COUPLED NONLINEAR DISSIPATION

By

Tong Zhou

Nonlinear algebraic loops in system equations may prevent the subsequent reduction of the equations to an explicit state-space form. It may make system simulation very difficult to accomplish. The incidence of nonlinear algebraic loops in mathematical system equations often is associated with the existence of nonlinear dissipative effects in physical systems.

A computation method for nonlinear algebraic loops stressing parallelism has been developed. The bond graph augmentation method converts an algebraic loop field into a dynamic subsystem that exhibits the proper static characteristics at steady state and employs a two-time-scale integration technique. In seeking computational efficiency, minimizing the augmentation order and optimizing the parameter selection play key roles. An augmentation sequence and a general rule for parameter selection for arbitrary n-th-order subsystems have been suggested and numerically tested in several cases.

To

my motherland China,

my parents W. J. Zhou , H. X. Lee,

and

my wife Shuling

ACKNOWLEDGEMENTS

The author wishes to express sincere thanks to Dr. Ronald C. Rosenberg for his guidance and encouragement over the past years, and for serving as major professor.

Appreciation is also expressed to the Albert H. Case center for providing computer facilities for this research.

CONTENTS

LIST OF TABLES

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES

1 INTRODUCTION

1.1 The Problem of Algebraic Loops in System Equations

A dynamic system consisting of a number of lumped-parameter elements may be described by ordinary differential equations in which time is the independent variable. By using vector notation, an n-th order differential system may be expressed as a set of first-order vector differential equations. This is the state-space representation.

Because the modern trend in engineering systems is toward greater complexity due mainly to the requirements of performing complex tasks with good accuracy, the state-space representation has come to play a very important role in modern system control and system dynamics. The general form of explicit state-space equations is

$$\dot{X}(t) = G_F(X(t),U(t)) \tag{1.1}$$

where the X is an n-dimensional vector, U is an m-dimensional vector, and G is a vector function. The superdot denotes time differentiation.

If the system is linear, Eq.(1.1) takes on a simpler form, as shown below

$$\dot{X} = [A]X + [B]U \tag{1.2}$$

where X and U are defined as before and [A] and [B] are matrices of appropriate dimensions.

In formulating equations for a dynamic system the explicit state-space equations often are preferred. If there are intermediate variables which are introduced in the initial problem formulation, there is the possibility that algebraic loops may exist in the system of equations. This situation may make it impossible to find an explicit form. More generally, the system equations can be written in the form

$$\dot{X}(t) = G_F(X, U, H) \tag{1.3a}$$

$$H(t) = G_L(X, U, H) \tag{1.3b}$$

where H is a vector of dimension r.

For a linear constant-coefficient system, an explicit state-space representation usually can be obtained by eliminating the intermediate variables from the equations. However, if even one of the G functions is a nonlinear function, it might make the elimination of H difficult, or even impossible. The existence of nonlinear algebraic loops in system equations may prevent subsequent reduction of the system equation set to an explicit state-space form. That leads us to search for efficient computational techniques to treat such problems.

1-2 Previous Research Work

The problem of algebraic loops has been studied by many researchers and many simulation programs are available that will diagnose the existence of algebraic loops in the equation set. Among those are CSMP III, CSSL-IV, DARE, and SCEPTRE [1,2,3,4]. Operationally, a loop diagnostic occurs following the equation sorting process. In this process,as the system equation set is manipulated , mutual algebraic loops are identified. Typically, execution of the program is terminated and appropriate modifications must be performed.

Most methods of numerical solution for nonlinear algebraic equations are extended from those used for solving linear systems. The Jacobi method, the Gauss-Seidel method , the successive overrelaxation method and the symmetric successive overrelaxation method are examples[5].

Among the methods for linear systems one which may be applicable to the nonlinear case is the use of an iterative procedure, each step of which involves the solution of linear algebraic equations. For example, consider the system

$$4u_1 - u_2 + (1/10)e^{u_1} = 1 \tag{1.4a}$$

$$-u_1 + 4u_2 + (1/8)u_1^2 = 0 \tag{1.4b}$$

An iterative method can be designed such that

$$4u_1^{(n+1)} - u_2^{(n+1)} + (1/10)e^{u^{(n)}} = 1 \tag{1.5a}$$

$$-u_1^{(n+1)} - 4u_2^{(n+1)} + (1/8)(u^{(n)})^2 = 0 \tag{1.5b}$$

To determine $u_1^{(n+1)}$ and $u_2^{(n+1)}$ from previous $u_1^{(n)}$ and $u_2^{(n)}$ involves solving a system of two linear equations with two unknowns.

Another method is the Newton method for solving a nonlinear system. Consider the case

$$f_1(u_1, u_2) = 0 \qquad (1.6a)$$

$$f_2(u_1, u_2) = 0 \qquad (1.6b)$$

which we may write as $Fu = 0$.

The Newton method is defined by

$$u^{(n+1)} = u^{(n)} - (F'u^{(n)})Fu^{(n)} \qquad (1.7)$$

where $F'u^{(n)}$ is the Jacobian matrix.

The Jacobian method is one of the basic methods for conducting the iteration. For a nonlinear system of equations

$$f_1(u_1, u_2, u_3) = 0 \qquad (1.8a)$$

$$f_2(u_1, u_2, u_3) = 0 \qquad (1.8b)$$

$$f_3(u_1, u_2, u_3) = 0 \qquad (1.8c)$$

We find $u_1^{(n+1)}, u_2^{(n+1)}, u_3^{(n+1)}$ by

$$f_1(u_1^{(n+1)}, u_2^{(n)}, u_3^{(n)}) = 0 \qquad (1.9a)$$

$$f_2(u_1^{(n)}, u_2^{(n+1)}, u_3^{(n)}) = 0 \qquad (1.9b)$$

$$f_3(u_1^{(n)}, u_2^{(n)}, u_3^{(n+1)}) = 0 \qquad (1.9c)$$

At each time step , one has to solve a single nonlinear equation for one unknown.

The Gauss-Seidel method is the same as the Jacobi method except that at each time stage one uses the latest available values. Thus for the above case

$$f_1(u_1^{(n+1)}, u_2^{(n)}, u_3^{(n)}) = 0 \tag{1.10a}$$

$$f_2(u_1^{(n+1)}, u_2^{(n+1)}, u_3^{(n)}) = 0 \tag{1.10b}$$

$$f_3(u_1^{(n+1)}, u_2^{(n+1)}, u_3^{(n+1)}) = 0 \tag{1.10c}$$

The successive overrelaxation method and symmetric overrelaxation method are both slight modifications of the Gauss-Seidel method.

Two methods frequently used in common are the Secant method for simultaneous nonlinear equations developed by Wolfe and Phillip[6], and quadratically convergent Newton-like method based upon Gaussian elimination developed by Brown[7]. The two have been implemented in code as ZSCNT and ZSYSTM respectively and collected into the IMSL subroutine liberary[8].

In Wolfe's method(ZSCNT) at each step of the iterative process there are n+1 trial solutions $x^1, x^2, \ldots, x^{(n+1)}$. Multipliers a_j , $j=1,2, \ldots, n+1$ are determined by solving the linear system

$$\sum_{j=1}^{n+1} a_j = 1 \tag{1.11}$$

$$\sum_{j=1}^{n+1} a_j f_i(x^i) = 0 \qquad i = 1,2,\ldots,n \tag{1.12}$$

Then the new trial solution is defined by

$$x = \sum_{j=1}^{n+1} a_j x_j \tag{1.13}$$

The Brown's method is based on Gaussian elimination in such a way that the most recsent information is always used at each step of algorithm. The modification suggests linearization of the components sequentially, using each linear equation to eliminate a single component of the solution from the remaining non-linear equations, as in Gauss elimination. The system eventually reduces to a single non-linear equation in a single unknown to which one step of the Newton iteration is then applied. The new values of all eliminated components are then obtained in the reverse order by back substitution.

1.3 Bond Graphs and Algebraic Loops

The existence of algebraic loops in the equations of a physical system may not be detected until the sorting or reducing process starts in most traditional simulation approaches. But their existence can be verified even before equation formulation when the bond graph approach is employed. Bond graphs, which are based on energy storage and power flow, allow system analysts and engineers to construct models of electrical, magnetic, mechanical, hydraulic, pneumatic, thermal and other systems using only a rather small set of ideal elements[9].

The functional nature of the parts of a bond graph model can be classified into the source field, energy storage field and dissipation field, while the manner in which the parts interact can be represented by the junction structure. All of this is done in a graphical format. A bond graph model and its key vectors may be represented schematically as shown in Figure 1-1. The key vectors are labeled on each arrow: U is the input to the junction structure from the source field; V is the output from the junction structure to the source field; Z is the co-energy variables vector; $\dot{X}$ is the time derivative of the energy variables vector; D_o is the input to the junction structure from the dissipation field (generally a mixture of efforts and flows) and D_i is the input to the dissipation field from the junction structure (generally a mixture of efforts and flows). From the fiqure we see that X is found from the junction structure as follows

$$\dot{X} = G_f(Z, D_o, U) \tag{1.14a}$$

or

$$\dot{X} = S_{11}Z + S_{12}D_o + S_{13}U \tag{1.14b}$$

if TF and GY elements are constant.

For the storage and dissipation field vectors, we have

$$Z = Q_f(X) \tag{1.15}$$

$$D_o = Q_L(D_i) \tag{1.16}$$

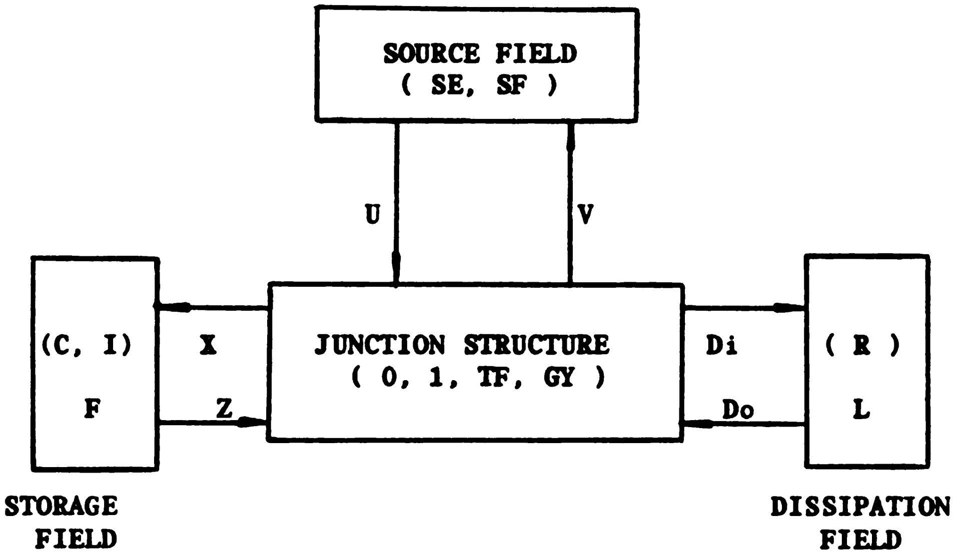


Figure 1-1 Bond graph structure and key variables

and $$U = U(t) \tag{1.17}$$

for the source field.

From the above schematic diagram , D_i , the input to the dissipation field from the junction structure is given by

$$D_i = G_L(Z,\ D_o,\ U) \tag{1.18a}$$

or $$D_i = S_{21}Z + S_{22}D_o + S_{23}U \tag{1.18b}$$

if the TF and GY elements are constant.

Combining Eqs.(1.16) and (1.18b) we get

$$D_i = S_{21}Z + S_{22}Q_L(D_i) + S_{23}U \tag{1.19}$$

Depending upon the interaction between S_{22} and Q_L it may or may not be possible to solve for D_i from Eq.(1.19) and then find D_o. To illustrate a case let us first consider a physical device shown in Figure 1-2(a). In Figure 1-2(b), the corresponding bond-graph model has been built. The I element represents the inertial effect and the compliance effect is indicated by a C element in the mechanical system. The R elements represent energy dissipative effects. The SF element indicates an imposed velocity on the left (massless) plate as an input.

Associated with every bond are two power variables --- effort and flow, which are force and velocity in this mechanical device, respec-

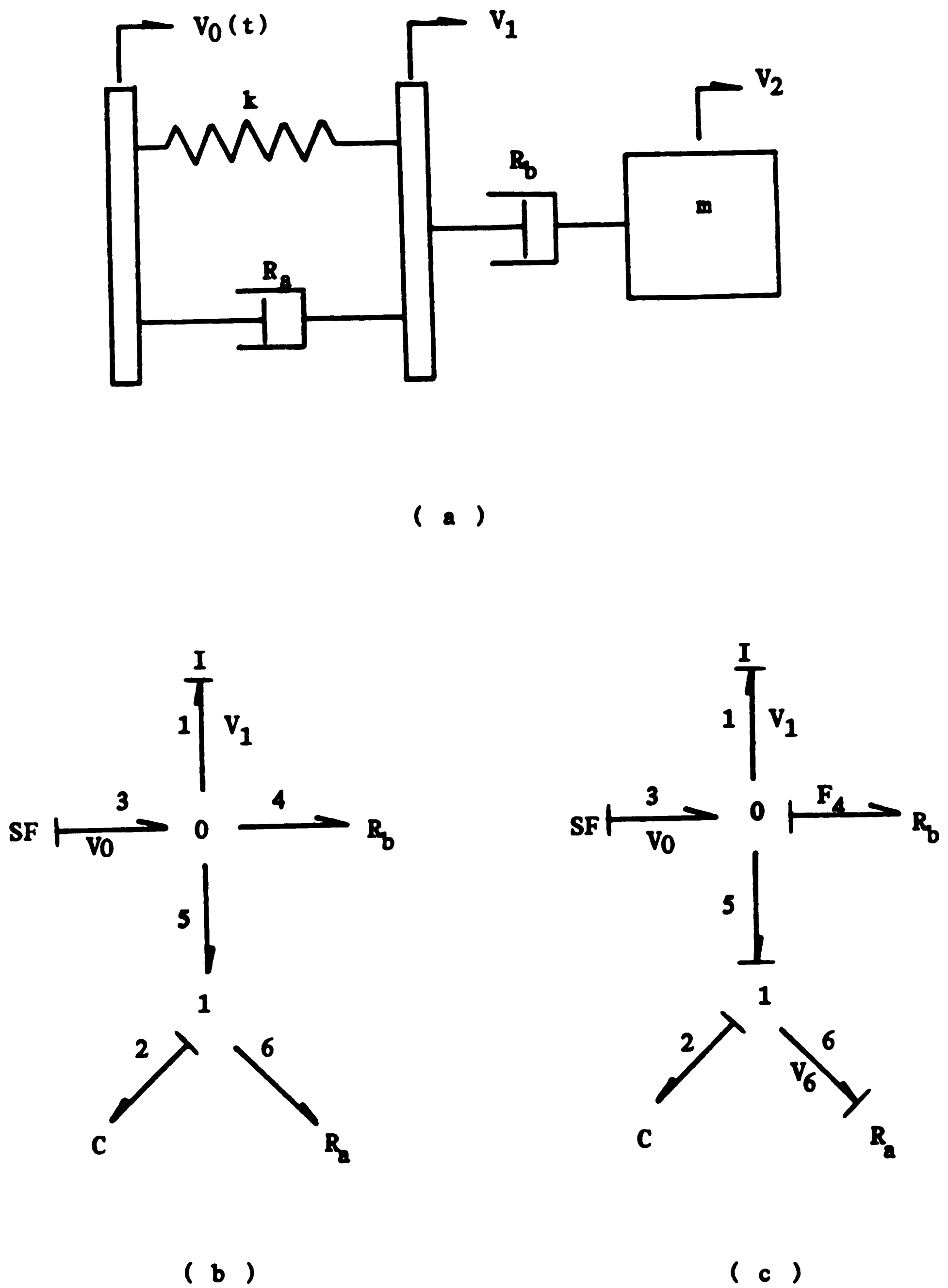


Figure 1-2 R-field in a physical system

(a) Physical system

(b) Bond graph model with acausal bonds

(c) Bond graph model with causality assigned tively. In this example there are six bonds; hence, there are six force variables and six velocity variables. In addition there are two (energy) state variables, p and δ, representing the momentum and the spring deflection, respectively. There are 14 equations imposed by the bond-graph structure through junction constraints and the field constitutive relationships. It is desired that the state-space equations be obtained in an explicit form as follows

$$\dot{p}_1 = g_1(p_1, \delta_2, V_0) \tag{1.20a}$$

$$\dot{\delta}_2 = g_2(p_1, \delta_2, V_0) \tag{1.20b}$$

where superdot denotes a time derivative.

Causality can be assigned to the bond-graph of Figure 1-2(b) according to the general rules[9]. After finishing the first step (assigning required causality to the source SF) and the second step (assigning the integral causality to the storage elements C and I), we find that the causality does not fully extend through the graph. Some acausal bonds (bonds 4,5,and 6) will be left (Figure 1-2(b)).

At this stage, we realize that an R-field exists in this system. This implies that there will be an algebraic loop in the system equations.

Suppose we continue the causality assignment by imposing an arbitrary causal orientation on one of the two R elements, say, R_4. Then we extend the causal implication through the graph using the constraint elements (0's and 1's). Now the causality assignment has been completed(Figure 1-2(c)). The state vector X and input vector U are identified as follows

$$X = \begin{bmatrix} p_1 \\ \delta_2 \end{bmatrix} \qquad U = [\ V_0\ ] \tag{1.21}$$

If we define F_4 and V_6 as auxiliary variables, then the system equations are

$$\dot{p}_1 = F_4 \tag{1.22a}$$

$$\dot{\delta}_2 = V_6 \tag{1.22b}$$

and the constitutive equations are

$$F_4 = g_4(V_4) = g_4(V_0 - p_1/m_1 - V_6) \tag{1.22a}$$

$$V_6 = g_6(F_6) = g_6(F_4 - k\delta_2\ ) \tag{1.22b}$$

Assume that both R_4 and R_6 are linear,that is,

$$F_4 = R_4V_4 \tag{1.23a}$$

$$V_6 = R_6^{-1}F_6 \tag{1.23b}$$

After some manipulations to eliminate the auxiliary variables F_4 and V_6, an explicit state-space equation set can be developed; namely,

$$\dot{p}_1 = -[R_4R_6/(R_4+R_6)m]p_1 + [R_4k/(R_4+R_6)]\delta_2 + [R_6R_4/(R_4+R\{6\})]V_0 \tag{1.24a}$$

$$\dot{\delta}_2 = -[R_4/(R_4+R_6)m]p_1 - [k/(R_4+R_6)]\delta_2 + [R_4/(R_4+R_6)]V_0 \tag{1.24b}$$

Now, suppose that the R_4 and R_6 are non-linear. Then we may have difficulty solving the auxiliary equations to get an explicit state form. In general explicit analytic solutions of non-linear coupled equations are difficult, if not impossible, to achieve.

From the development above, we see that the process of causality assignment is an aid in the process of identifying the algebraic loops in dynamic systems. Furthermore, algebraic loops in the mathematical sense are physically related to the existence of dissipation fields. Reading the partial causally-assigned bond-graph, we easily can identify the R-fields from other (dynamic) fields. This work can be done by a digital computer automatically.

1.4 An Approach Stressing Parallelism

An approach to the simulation of this kind of systems containing non-linear algebraic loops is the bond graph augmentation method[10]. The basic philosophy is to convert an algebraic loop subsystem into a dynamic subsystem that conserves the intrinsic static characteristics at its steady state and to employ a two-time-scale integration technique.

This approach divides a large coupled nonlinear system into a group of static subsystems and a group of dynamic subsystems. Every subsystem in the static group is independent of the others, as is every subsystem in the dynamic group. However the members of one group will communicate with the members of the other group. At each global time step, the static characteristics of every augmented subsystem at its steady state can be computed independently by the same algorithm. Therefore they may be done concurrently or in parallel. The parallelism also can be applied to the computation of the dynamic performances of the dynamic group at each global time step. The two-time-scale integration technique places special emphasis on the parallelism, which will become more practical with advent of the parallel computers.

The method includes mainly the following steps:

1) Make a bond-graph model for the non-linear system;
2) Identify and isolate the algebraic loops before equation formulation;
3) Dynamically augment the isolated static subsystems;
4) Select the proper parameters for the dynamicizers;
5) Use a two-time-scale integration scheme to get simulation results.

In the following chapters we will discuss the details of this approach.

2 BOND GRAPH AUGMENTATION METHOD

2.1 Identification and Partitioning of R-fields

A bond graph gives a topological picture of how the power (or energy) flows in a particular physical system. Since every bond contains the power variables (effort and flow) and the causality shows input/output relationship between the element pair, we can identify R-fields readily. Let us consider an example. The bond-graph model is shown in Figure 2-1(c), which represents a mechanical system (Figure 2-1(b)) or an electrical system (Figure 2-1(a)).

After assigning causality to the source element SF and energy elements C and I, an R-field has been revealed. In this example , to the right of bond 12 is a causally complete segment, while to the left is a causally incomplete segment (Figure 2-2). The two segments share bond 12. If we break the bond-graph into two subgraphs at bond 12 , we see that at any instant the output of the dynamic subsystem represented by the right part is just the input to the static subsystem indicated by the left part, and vice versa. The subsystems interface at bond 12. For example , at time $t=t_0$, e_{12} is obtained from the dynamic field and input as an effort source to the static field. If the output of the static field can be calculated in some way which will be discussed in the later section, then the output will act as a flow source at time $t=t+\Delta T$ to the dynamic field, by integrating one time step, ΔT, the dynamic field produce a new output to the

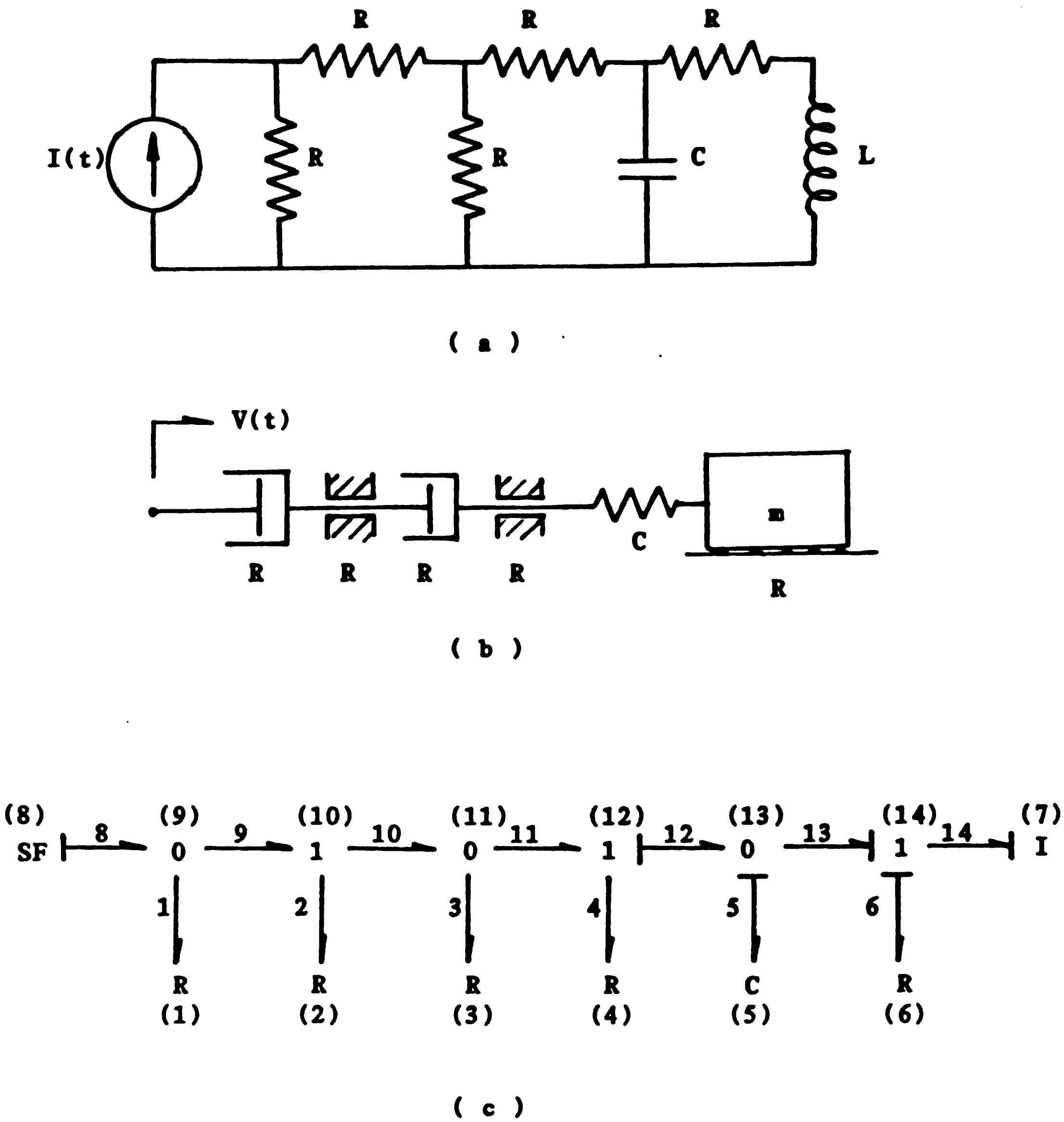


Figure 2-1 Physical systems with a coupled R-field

(a) Electrical Circuit

(b) Mechanical system

(c) Bond graph model

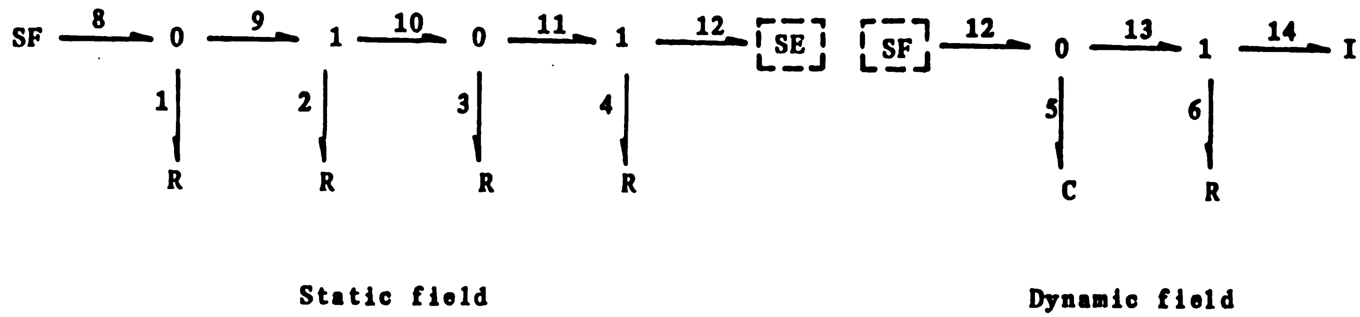


Figure 2-2 The partitioned subsystems

static field again. In this manner, the simulation of the global system may be completed by interaction of its two subsystems as long as the static field can be manipulated easily and efficiently.

In the general sense a global bond-graph can be partitioned into causally complete and incomplete segments. The causally complete segments generally are comprised of energy elements (C and I), dissipative elements (R), input elements (SE and SF) and the junction structure (0,1, TF,GY), while the causally incomplete segments contain the dissipation fields, input fields and associated junction structure.

In Figure 2-3 the concept of a general partitioned bond-graph is illustrated. The interaction between the i-th dynamic subsystem and the j-th static subsystem can be defined in vector notation. Each subsystem may be viewed as an independent system with both input and output vectors ascribed to it.

2.2 Dynamic Augmentation of Static Subsystems

The Basis of Augmentation

The basic idea in solving the algebraic loops is to convert a coupled static field into a dynamic subsystem by introducing some dynamic effects, to find the steady state output under a set of constant inputs, and then to interact with other subsystems in the global

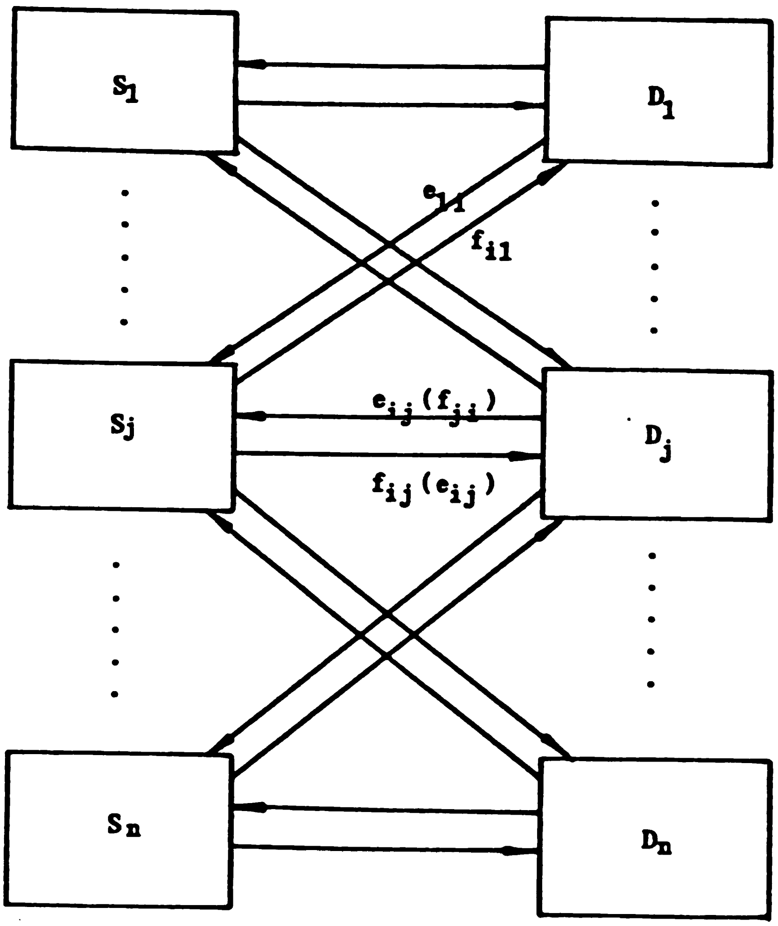


Figure 2-3 The interactions among the partitioned subsystems

S_i : the i-th static subsystem

D_j : the j-th dynamic subsystem

system.

The dynamic augmentation can be done by introducing I- and C-elements in such a way that the causality of the static field is fully completed and the boundary conditions are not changed.

Consider a static field and a proposed dynamically augmented subsystem associated with the static field (Figure 2-4). From the definitions of the junction elements the following statement can be made:

$$\sum \text{efforts} = 0 \qquad \text{1-junction} \qquad (2.1a)$$

$$\sum \text{flows} = 0 \qquad \text{0-junction} \qquad (2.1b)$$

for the static field,

$$\sum \text{efforts} = \dot{p}_1 \qquad \text{1-junction} \qquad (2.2a)$$

$$\sum \text{flows} = \dot{q}_2 \qquad \text{0-junction} \qquad (2.2b)$$

and for the augmented system. Note that p_1 and q_2 measure the energy in I and C, respectively.

The condition of steady state requires that

$$\dot{p}_1 = 0$$

$$\dot{q}_2 = 0$$

Consequently at steady state the state equations will be identical to the junction constraint equations in the static field; i.e.

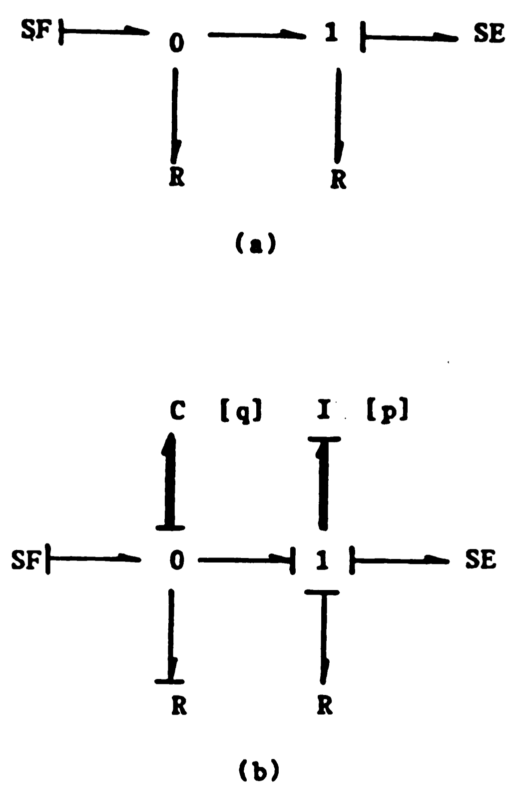


Figure 2-4 A static field and its augmentation

(a) A partitioned static field

(b) The augmented subsystem

$$\sum \text{efforts} = 0$$
$$\sum \text{flows} = 0$$

From this result it is evident that the algebraic character of the static subsystem is preserved at the steady state of a properly augmented dynamic subsystem. At steady state the output e_1 and f_s of the subsystem which are treated as inputs to the adjacent subsystems can be easily determined by

$$e_1 = (1/C)q \tag{2.3a}$$

$$f_s = (1/I)p \tag{2.3b}$$

To understand the nature of dynamic augmentation, let us interpret the bond graph as a physical device as shown in Figure 2-5.

Consider Figure 2-5(a), where A is a massless element being acted upon by a force F, a velocity source V through a damper, and by another resistive force from the ground. This really is a static system. If a mass effect I and an elastic effect C are added into the system , the sum of the forces at any instant is equal to the change in momentum of the mass, and the sum of the velocities at any instant is equal to the change in the length of the spring. At steady state, the rates of change of momentum and length are zero; hence the dynamic system represents the unaugmented static subsystem at equilibrim.

Only adding I elements to 1-junctions and C elements to

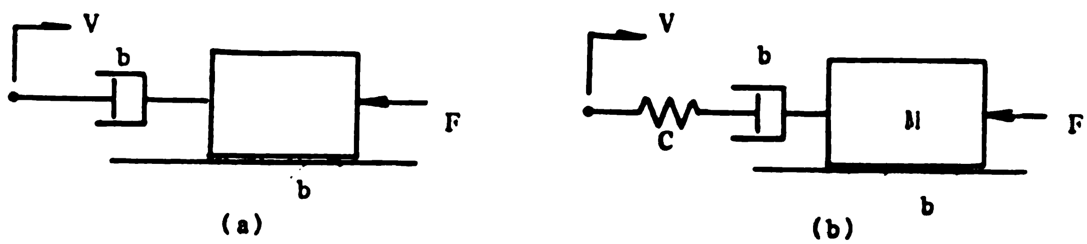


Figure 2-5 Schematic diagrams

(a) Static system

(b) Modified system

0-junctions facilitates the completion of the causality assignment and does not change the original causality orientation at the boundary bonds. So, the principle should be

(1) Add I elements to the 1-junctions

(2) Add C elements to the 0-junctions

The added I and C elements are assumed to be linear, conservative energy storage elements.

Some Definitions

For convenience in the development later on , it is useful to define some terminology which is used frequently in this chapter.

Definition: An R-field or static field , is a collection of dissipative elements (R), source elements (SE and SF) and junctions (0,1,TF and GY), in which the causality assignment is not able to be finished uniquely.

Definition: A dynamic field is a segment of the global bond-graph in which there is at least one dynamic element and the causality can be assigned completely.

The dynamic field may be a collection of some basic elements

(C,I,R,SE,SF,0,1,TF,GY) or it may contain only one single dynamic element (C or I).

Definition: An interacting bond is the connecting bond between an R-field and a dynamic field in the global bond-graph.

Definition: An acting junction is a junction with one or more interacting bonds attached.

Definition: A multi-branch type junction is a junction which possesses more than two internal bonds.

Definition: A single-R type junction is a junction to which only one R element is attached .

Definition: A multi-R type junction is a junction to which more than one R element is attached.

Definition: An imaginary source is a constant effort or flow source whose magnitude equals that of the output of the adjacent field.

An example is shown in Figure 2-6. The global bond-graph model has been partially assigned in causality and then partitioned into two dynamic fields and one R-field. Bond 8 connects dynamic field 1 and

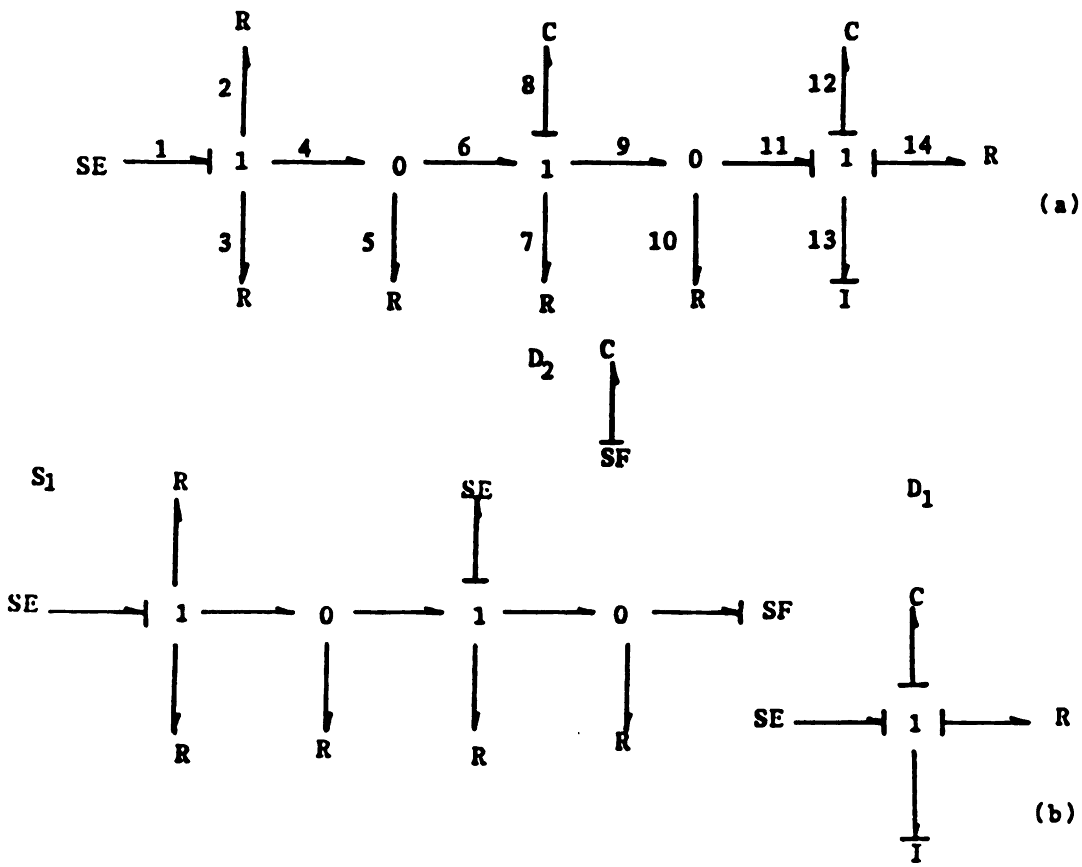


Figure 2-6 Partitioning

(a) Global system

(b) Partitioned subsystems

the R-field, while bond 11 connects the R-field and dynamic field 2. All the interactions among these subsystems occur on these two bonds, so they are the interacting bonds. Nodes (9), (11) and (15) are acting junctions because interacting bonds 8 and 11 attach to them. Nodes (6), (9) and (11) are single-R type junctions, while node (4) is a multi-R type junction. In the partitioned subsystems, SE12, SF21, SE11 and SF11 are introduced to the subgraphs for graph completion and input functions. They are defined as imaginary sources.

Order of the Augmented Subsystem

It is evident that every added energy storage element will introduce a new state variable and a corresponding parameter. Hence it will increase the dynamic order of the subsystem by one.

For a large R-field with a number of non-linear resistive elements, by increasing the number of added dynamic elements, the dynamic order will be increased considerably. This increase may benefit us in one way, in that it increases the flexibility in satisfying the desired causality orientations of all R elements in the subsystem. However, the computational efficiency may be decreased with the increase of the dynamic order. Furthermore, the higher the dynamic order, the more difficult parameter selection will be, because an improper selection of parameters for the added elements may cause a considerable delay in the field reaching its steady state . Consequently augmentation greatly affects the computational efficiency and the accuracy. Parameter selection will be discussed in detail in the next chapter.

Augmentation could be done according to at least two different considerations. One of them is to make the causality orientation satisfy the nature of every particular R element. Reconsider the sub-system containing an algebraic loop represented by the bond-graph in Figure 2-7(a). By adopting the principle mentioned before, we have three possibilities to augment the subsystem by (1) adding only one I element, (2) adding only one C element, or (3) adding one I and one C element to each 1- and 0-junction, respectively. If the constitutive functions of the resistive elements in the system are

$$e_1 = R_1(f_1)$$

$$e_2 = R_2(f_2)$$

choice (1) satisfies the required nature of these R elements. The causality would be assigned as shown in Figure 2-7(b). Other choices are shown in parts (c) and (d).

In this case there are two single-R type junctions; one is the 0-junction and the other one is 1-junction. The maximum number of useful added dynamic elements is two; that is, it equals the sum of the number of 0-junctions and 1-junctions. However, in a general case with more single-R type or multi-R type junctions, it could be visualized that a number of possibilities to introduce the storage elements exist. The number of useful ways depends upon the type of resistive

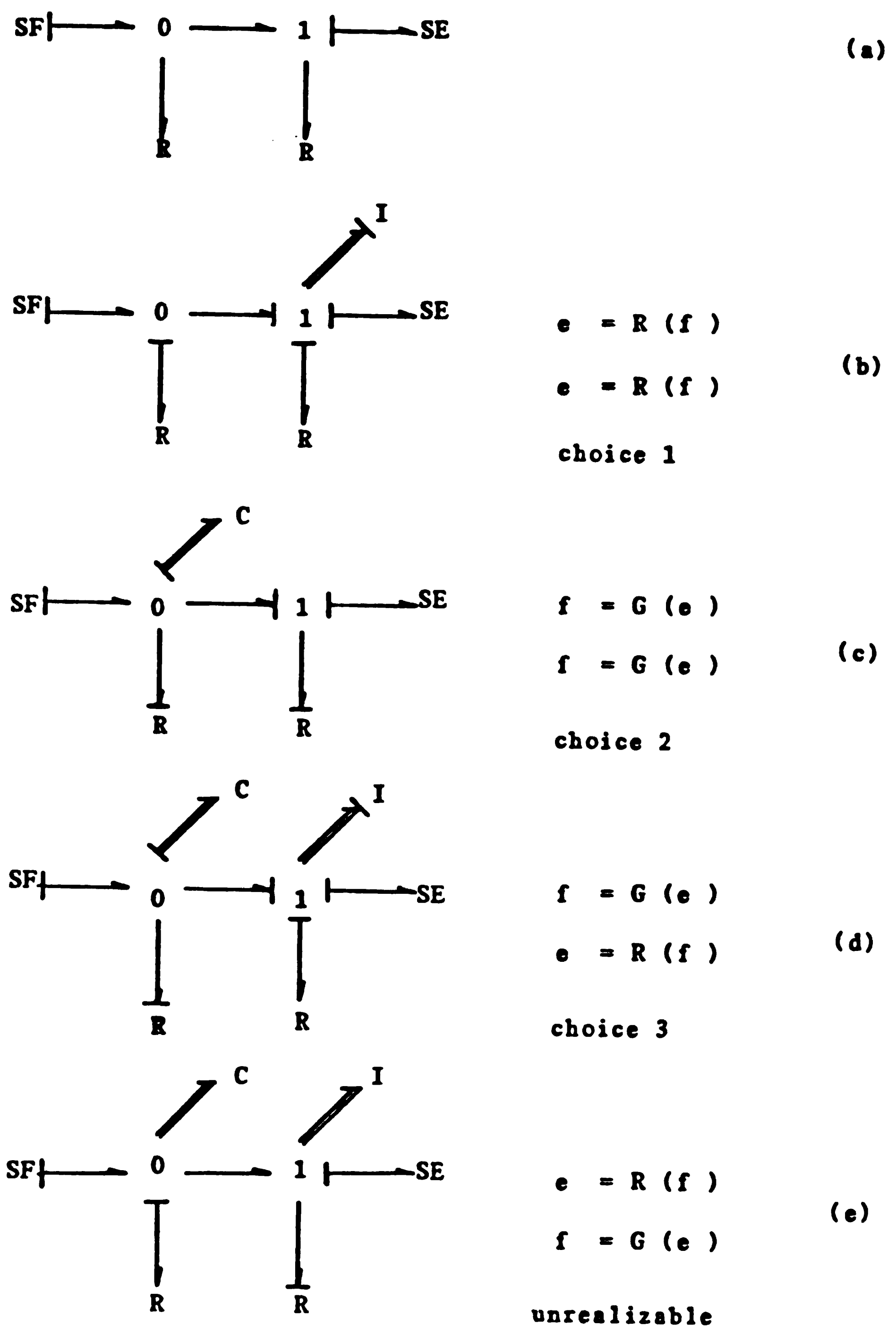


Figure 2-7 Causality orientations

elements in the algebraic loop. If some of them are non-linear and their preferred causality is required, then the number of ways in which the C and I elements are selected is reduced.

Suppose that the causality orientations of all R elements in a a-field are not of concern. That is, the inverse expressions of dissipation functions are easy to get. Then the causality orientations could be any combination. But the maximum order of augmentation would be sum of the numbers of the 0- and 1-junctions.

Carefully studying the example above(Figure 2-7) reveals the fact that only three causal arrangements are available through selective dynamic augmentation. However, the dissipative elements may exhibit four different causal arrangements. For the fourth situation, part(e), the method of dynamic augmentation is inadequate to force the preferred causal orientation on both R elements.

Generally, for translational and rotational dissipative elements and for most fluid resistive effects we have the case in which a dissipation is the function of flow. But some cases, such as semiconductors in electronics and fluid leakage in hydraulics and pneumatics, have the inverse dissipation function. Fortunately, many non-linear dissipation functions are not too complicated and their inverse forms may be found without much effort. But important exceptions exist. For example, in mechanical systems the dry friction is not allowed to be a function of force.

Augmentation Procedure

Assume that every dissipation function can be expressed either as $e = R(f)$ or $f = G(e)$. Then it will be possible for a computer to augment the subsystem automatically without regard to the preferred causality orientations. The procedure is designed as follows.

First we note that every subsystem needs to compute the interacting bond variables as output to the adjacent dynamic subsystems. These bond variables as outputs always are the common forces or flows associated with the 0-junctions or 1-junctions. Therefore, if we add energy elements to the acting junctions, we not only facilitate the causality assignment but also obtain the output vector from the co-energy variables without invoking more complicated algebraic output equations. So adding the I or C elements to the acting junctions possesses the highest priority.

For example, the subgraph in Figure 2-8(a) represents a static field containing a real source SE1 and an imaginary source SE18. Bond 18 is an interacting bond and node (18) is an acting junction. At a certain global time instant, e_{18} is the input from an adjacent dynamic field which holds constant during the integration in the local time scale.When the augmented subsystem reaches its local steady state, f_{18} will be the output to the dynamic field. This output is just the common flow on the acting junction. If we add an I element to the acting junction (18) (Figure 2-8(b)), then the output can be calculated from

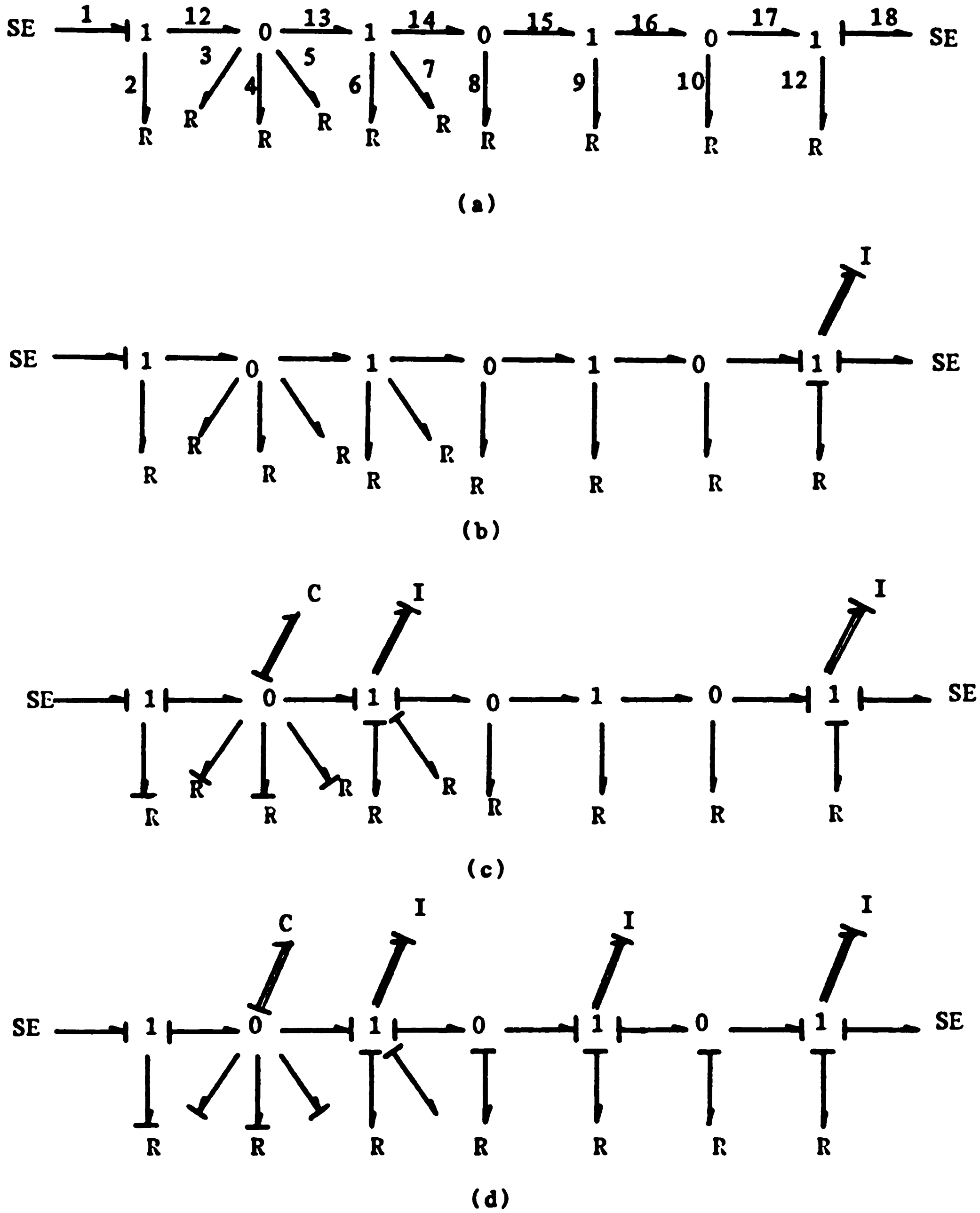


Figure 2-8 An example

(a) A static susystem

(b) After step 1

(c) After step 2

(d) After step 4

$$f_{18} = f_{19} = p_{19} / m_{19}$$

Second, the bonds associated with the multi-R type junction can not be expected to complete a causal assignment by extending from neighbor junctions. We complete the causal assignment by adding a dynamic element directly to this type of junction. In Figure 2-8(c), a C element has been added to node (12) and an I element has been added to node (13). In a computer program it should be done in descending order until no multi-R type junctions exist.

Third, if there exist some multi-branch type junctions in the static field, it is desireable to add dynamic elements to them.

Fourth, add the dynamic elements only to the remaining unassigned single-R type 0- or 1-junctions, whichever is fewer in number. If these two kinds of unassigned junctions have the same number, either kind can be chosen for augmentation.

In this example, there are two single-R type 0-junctions and one single-R type 1-junction unassigned. For the lower order augmentation adding one I element is enough to complete the causality assingment for this subsystem (Figure 2-8(d)).

After assigning and extending the required causalities for all the sources and the preferred causalities for all the storage elementes (C or I), the program PART will begin to identify the dynamic fields and the static fields, then partition them into subsystems by using the imaginary sources to preserve the boundary conditions. All the information will be stored in matrix form where every column or row contains a dynamic or static subsystem. The program AUGMEN finds the most appropriate 0- or 1-junction to add the corresponding C or I element according to the procedure discussed before. They will be incorporated into the ENPORT-6 program. The calling tree is shown in Appendix A and the program listings are given in Appendix B.

2.3 Two-time-scale Integration

The solution for the steady-state vector X in the case of linear dissipative fields can be achieved by simple linear algebra, provided the local [A] matrix is nonsingular.

$$X = -[A]^{-1}[B]U$$

However, in the case of non-linear dissipative fields we can not use this approach to get a solution. Although we can employ a linearization method to get a constant [R] matrix at a certain time instant, then use the formula above to calculate steady state, it will reduce the accuracy of computation. For example, we can calculate X at time t through

$$X_t = -[A]^{-1}_{t-\Delta t}[B]U_t$$

The [A] matrix is not the present value. If we can use non-linear integration to get the steady-state in the local time scale, the computation result will be more accurate and so will the output vector.

The augmented subsystems need to reach steady state by integration under a set of constant inputs which are the outputs from adjacent dynamic subsystems at global time T . It seems that when the dynamically augmented subsystem is under the process of integration in local time , t , the global time is 'frozen'. After the steady state has been reached, the output vector of augmented subsystem will be sent to the adjacent dynamic subsystems and the global time will warm up and make a step (ΔT) forward for the global integration. A new output vector from the one-step integration refreshes the input vector of the augmented subsystems. These subsystems will integrate again in their local time scales to get a new steady-state . This process is repeated for the total global time interval and the desired simulation results. Since there are two time scales in the whole simulation process , one for the local subsystems and the other one for the global system, we call this scheme two-time-scale integration. A schematic diagram is shown in Figure 2-9.

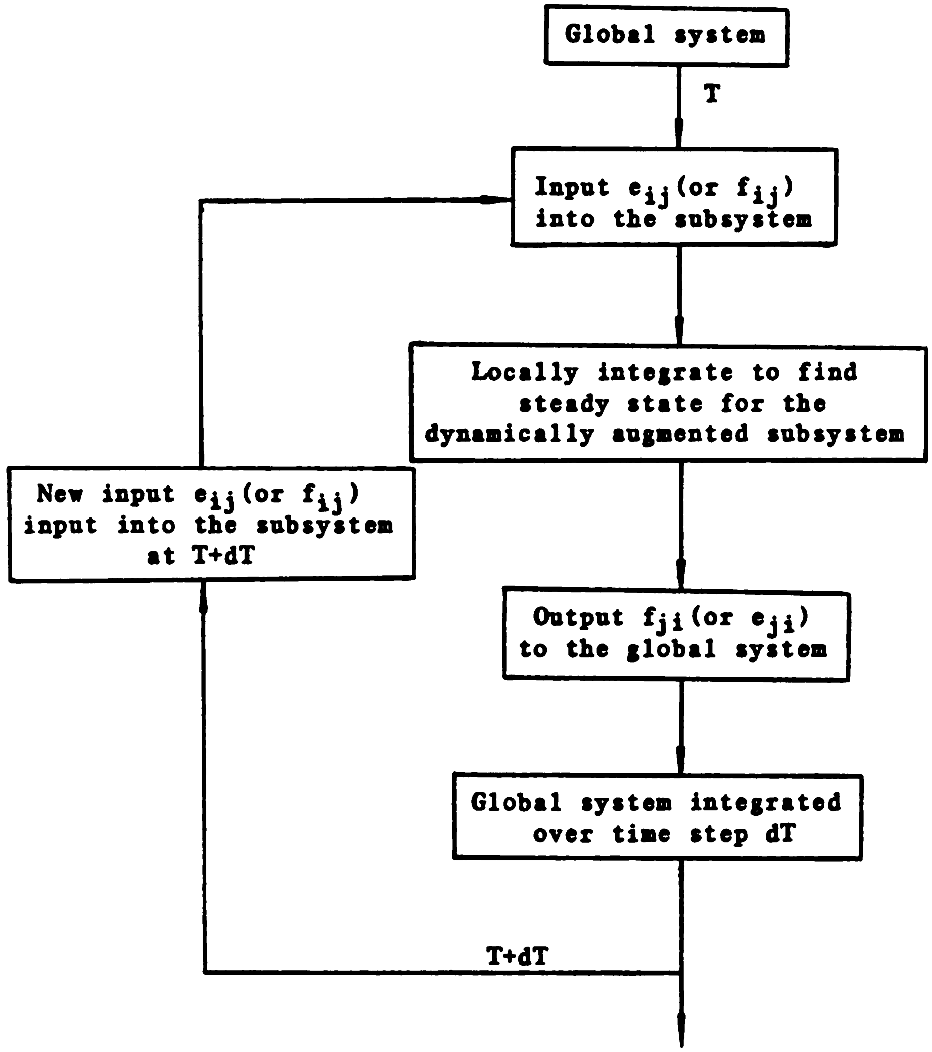


Figure 2-9 Schematic diagram of two-time-scale integration

3 SOME COMPUTATIONAL CONSIDERATIONS

3.1 Effect of Eigenvalues on Computation Efficiency

Our ultimate goal is to simulate the dynamic response of non-linear systems, but the following discussion on linear systems may benefit this.

For a dynamically augmented subsystem, the system representation is readily arranged into an explicit state-space form

$$\dot{X} = [A]X + [B]U \tag{3.1}$$

Structurally, the [A] matrix may be resolved further into the following form since no dependent energy variables exist (i.e., all integral causality):

$$[A] = -\Big[\ R\ \Big]\Big[\ M^{-1}\ \Big] \tag{3.2a}$$

for the augmented subsystems by adding only I elements,

or
$$[A] = -\Big[\ G\ \Big]\Big[\ K\ \Big] \tag{3.2b}$$

for the augmented subsystems by adding only C elements.

The [R] and [G] matrices are derived from the bond-graph topology and dissipative elements, while the $[M^{-1}]$ and [K] matrices consist of the free parameters introduced through the dynamic augmentation.

The $[M^{-1}]$ or $[K]$ matrix corresponding to the energy-storage fields is diagonal and positive definite for integral causality. all non-zero terms in the $[M^{-1}]$ or $[K]$ matrix are positive. For the R-fields to dissipate energy for any possible port condition, the resistance parameters should be positive . This is true because only the 1-port resistors can dissipate power and the junction elements conserve power. From the preceding statements, it follows that the [R] matrix will be positive definite.

The question that arises is "How should the free parameters be selected to provide computationally efficient convergence to the steady state?"

As mentioned before, during the local time integration the input vector is a constant vector. We want the augmented state variables of each subfield to follow a path like below to reach their steady state

$$X = \lim_{t \to \infty} e^{At} \int_{t_0}^{t} e^{-A\tau} BU \, d\tau \tag{3.3}$$

where A is an n n matrix with real negative eigenvalues and e^{At} is the fundamental matrix of the system, B is a n*m matrix and U is the input vector with dimension m.

Let us call this form "diffusion"-type convergence. Graphically, the "diffusion" convergence has the exponential feature as in Figure 3-1(a) and 3-1(b).The n*n matrix [A] has n eigenvalues. Suppose that the augmented subsystem is a "diffusion"-type, that is all the eigen-

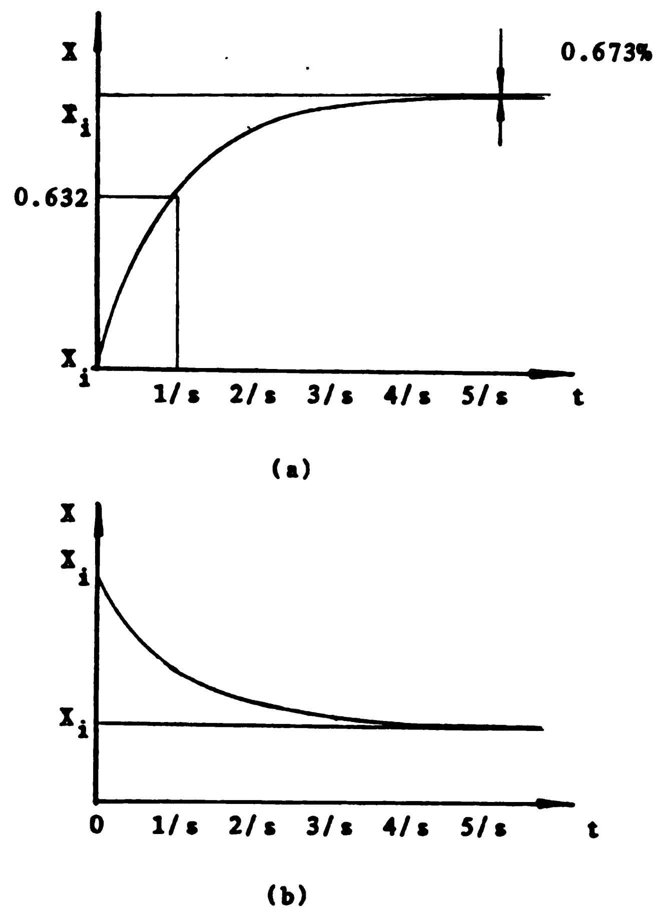


Figure 3-1 "Diffusion" convergence

values are negative real numbers. They are s_1, s_2, ..., s_n. To achieve reasonable accuracy the integration duration should be set to at least five times ($1/|s|_{min}$) and the time-step should be set less than one third of $1/|s|_{max}$ (Figure 3-1). From these considerations, we hope all eigenvalues are clustered. This means that all the eigenvalues lie within a small region in the s-plane, so that the number of integration steps can be minimized .

Mathematically, we can state the objective as trying to choose k_{ii} such that the normalized "spread" of the eigenvalues, ρ, be minimized, namely

$$\rho = (|s|_{max} - |s|_{min}) / (|s|_{max} + |s|_{min}) \tag{3.4}$$

where s_{min} and s_{max} are the minimum and maximum among the all eigenvalues,respectively. The value of the "spread" varies between zero and one .

The eigenvalues of the [A] matrix are determined by the characteristics of the [R] matrix, which is intrinsic to the field , and the [K] matrix, in which every diagonal term is the parameter of an added dynamic element. Although the minimum of the "spread" ρ is determined by the [R] matrix itself, getting to the minimum depends upon how we select the parameters of the added elements.

3.2 Parameter Selection in Linear Dissipative Fields

Now let us discuss the issue of how to select the parameters of the added elements starting from the simplest first-order case to get some feeling about the general rule for the n-th order case.

First-order Augmented Subsystems

The first-order augmentation has one free parameter to be selected. It is the simplest case in R-field augmentation. Consider the R-field shown in Figure 3-2(a) and (b). The state equations for the augmented R-fields (a) and (b) are given by Eq.(3.5a) and Eq.(3.5b), respectively:

$$\dot{p}_6 = -(R_2 + R_4)p_6/m_6 + R_2V_1 - F_3 \qquad (3.5a)$$

$$\dot{p}_5 = -(R_2 + R_4)p_5/m_5 + F_1 - F_3 \qquad (3.5b)$$

where the [A] matrix is

$$[A] = -[R_2 + R_4](1/m) \qquad (3.6)$$

The eigenvalues of both equations have the same form

$$s = -(R_2 + R_4)/m \qquad (3.7)$$

If we set s to be an arbitrary constant, n , for example, s = -1, the parameter m may be determined by

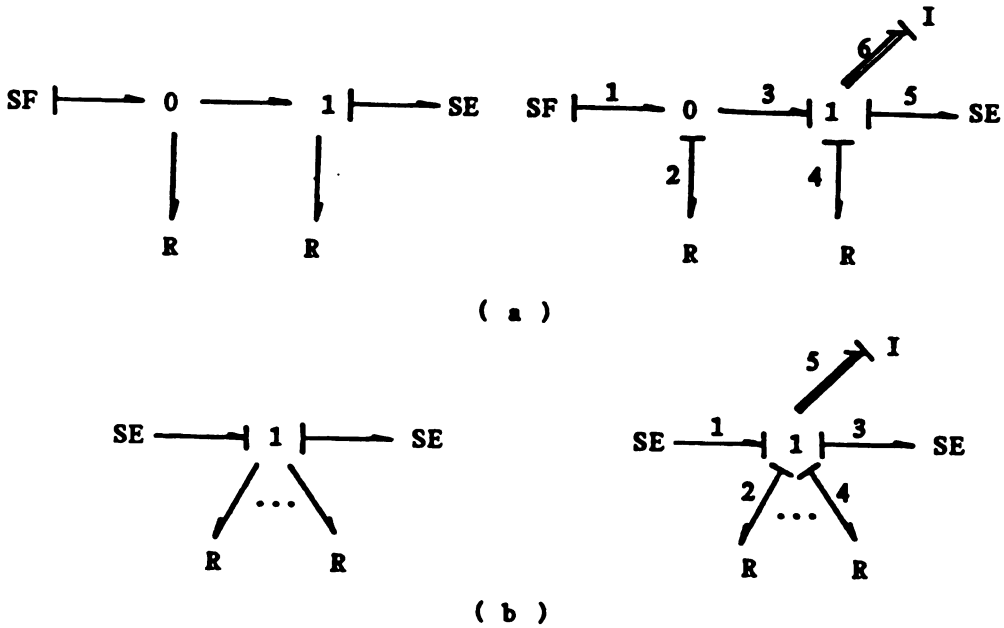
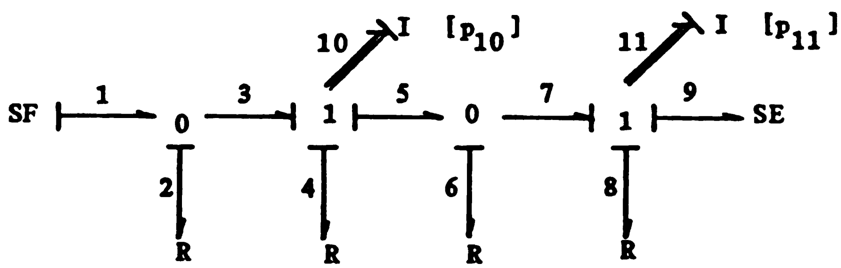


Figure 3-2 The first-order augmentation

(a) Single-R type

(b) Multi-R type



$$X = \begin{bmatrix} p_{10} \\ p_{11} \end{bmatrix} \qquad U = \begin{bmatrix} f_1 \\ e_9 \end{bmatrix}$$

Figure 3-3 The second-order augmentation

$$m = R_2 + R_4 \tag{3.8}$$

Using the eigenvalue $s = -1$, we can choose the time interval and the duration for the local time integration

$$\Delta t = (1/3)*(1/s\ ) = 1/3 \tag{3.9a}$$

and $$t_f = 5*(1/s\ ) = 5 \tag{3.9b}$$

which requires 15 steps for "reasonable" accuracy.

Second-order Augmented Subsystems

A typical second-order dynamic augmented subsystem is shown in Figure 3-3. The state equations in matrix form are

$$\begin{bmatrix} \dot{p}_{10} \\ \dot{p}_{11} \end{bmatrix} = \begin{bmatrix} -(R_2 + R_4 + R_6) & R_6 \\ R_6 & -(R_6 + R_8\ ) \end{bmatrix} \begin{bmatrix} 1/m_{10} & 0 \\ 0 & 1/m_{11} \end{bmatrix} \begin{bmatrix} p_{10} \\ p_{11} \end{bmatrix} + \begin{bmatrix} R_2 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} V_1 \\ F_9 \end{bmatrix} \tag{3.10}$$

where the [A] matrix is

$$[A] = \begin{bmatrix} -(R_2 + R_4 + R_6)/m_{10} & R_6/m_{11} \\ R_6/m_{10} & -(R_6 + R_8)/m_{11} \end{bmatrix}$$

At this point it is the appropriate time to apply the Gershgorin circle theorem to study the eigenvalue distribution.

The Gershgorin circle theorem[11] states that:

Let [A] be an n*n matrix, and let C_i, i = 1, 2, ..., n be the discs with centers a_{ii} and radii $Ri = \sum_{k=1, k\neq i}^{n} a_{ik}$. Let D denote the union of the discs C_i. Then all the eigenvalues of [A] lie within D.

If we inspect the [R] matrix, which is generated from the second-order linear system, we may find the off-diagonal terms are less than the corresponding diagonal terms. This implies that the locations of the circle centers are more important than the radii in estimating the distribution of eigenvalues for this type matrix. Generally we expect that if all the circle centers are located at the same point in the s-plane and all the radii are as similar as possible, then the distribution region would be reduced. Therefore, the "spread" of eigenvalues may be the minimum.

For simplicity, let all centers be at -1.0. The conditions are

$$R_2 + R_4 + R_6 = m_{10} \tag{3.11a}$$

$$R_6 + R_8 = m_{11} \tag{3.11b}$$

then the [A] matrix becomes

$$[A] = \begin{bmatrix} -1 & R_6/(R_6 + R_8) \\ R_6/(R_2 + R_4 + R_6) & -1 \end{bmatrix}$$

The characteristic polynomial p(s) would be

$$p(s) = \det(sI - A)$$

$$= s^2 + 2s + 1 - R_6^2/[(R_2+R_4+R_6)(R_6+R_8)] = 0 \qquad (3.12)$$

The roots of p(s) are

$$s_{1,2} = -1 \pm \left[R_6^2/(R_2+R_4+R_6)(R_6+R_8)\right]^{1/2} \qquad (3.13)$$

Because the radical term is always greater than zero and less than one, two negative real eigenvalues are produced. Therefore "diffusion" dynamics could be realized.

We also can analytically verify the correctness of applying the Gershgorin circle theorem using a general second-order augmented case in which the [A] matrix has the following form

$$[A] = -\begin{bmatrix} g_{11} & g_{12} \\ g_{12} & g_{22} \end{bmatrix} \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix}$$

where the [G] matrix is symmetric, positive definite and the [K] matrix is positive diagonal because of the topology characteristics of R-field. Another property of matrix [G] is that the diagonal terms g_{ii} are dominant.

The characteristic polynomial p(s) would be

$$p(s) = \det(sI - A)$$

$$= s^2 + s(g_{11}k_1 + g_{22}k_2) + (g_{11}k_1g_{22}k_2 - g_{12}^2k_1k_2) \quad (3.14)$$

The roots of p(s) are

$$s_{1,2} = -(g_{11}k_1 + g_{22}k_2)/2 \pm [(g_{11}k_1 + g_{22}k_2)^2 - 4(g_{11}k_1g_{22}k_2 - g_{12}^2k_1k_2)]^{1/2}/2 \quad (3.15a)$$

or

$$s_{1,2} = -(g_{11}k_1 + g_{22}k_2)/2 \pm [(g_{11}k_1 - g_{22}k_2)^2 - 4g_{12}^2k_1k_2]^{1/2}/2 \quad (3.15b)$$

Eq.(3.15) shows that the roots will be real and negative. We simplify the notation of $s_{1,2}$ as follows

$$s_{1,2} = -B/2 \pm (B^2 - 4C)^{1/2}/2 \quad (3.16)$$

where

$$B = g_{11}k_1 + g_{22}k_2 \quad (3.17)$$

$$C = g_{11}k_1g_{22}k_2 - g_{12}^2k_1k_2 \quad (3.18)$$

Then the "spread" is

$$\rho = (1 - 4C/B^2)^{1/2} \quad (3.19)$$

Since $\rho > 0$, we can minimize ρ^2 with respect to k_1 and k_2, namely

$$d(\rho^2)/dk_1 = -4(g_{11}g_{22}k_2 - g_{12}^2k_2 - 2Cg_{11}/B)B^2 \quad (3.20a)$$

$$d(\rho^2)/dk_2 = -4(g_{11}g_{22}k_1 - g_{12}^2k_1 - 2Cg_{22}/B)B^2 \quad (3.20b)$$

These equations yield

$$2Cg_{11} - BC/k_1 = 0 \qquad (3.21a)$$

and

$$2Cg_{22} - BC/k_2 = 0 \qquad (3.21b)$$

Finally we have

$$k_{11}g_{11} = k_2 g_{22} \qquad (3.22)$$

To ensure a minimum we also can show that $d^2(\rho^2)/dk_1^2 > 0$ and $d^2(\rho^2)/dk_2^2 > 0$. So if we choose $k_1 = 1/g_{11}$, then k_2 would be $1/g_{22}$ from the above relation, i.e., one of the optimal selection will be

$$k_1 = 1/g_{11} \qquad (3.23a)$$

$$k_2 = 1/g_{22} \qquad (3.23b)$$

This coincides with the result obtained by using the Gershgorin circle theorem. Let $k_1 = 1/g_{11}$ and $k_2 = 1/g_{22}$, then "spread" ρ will be

$$\rho = g_{12}(1/g_{11}g_{22})^{1/2} \qquad (3.24)$$

Since $g_{12}^2 < g_{11}g_{22}$, so $\rho^2 = g_{12}^2/g_{11}g_{22} < 1$; hence $\rho < 1$.

The above observation has been illustrated by an arbitrary numerical example in TABLE 1. The [A] matrix is

TABLE 3-1 EIGENVALUES OF A SECOND ORDER AUGMENTATION

		CASE 1	CASE 2	CASE 3
Parameters Of R elements	R2		100	
	R4		100	
	R6		10	
	R8		100	
Parameters of added elements	M12	R2+R4+R6= 210	R2+R4+R6= 210	R2+R4+R6= 210
	M11	R6+R8= 110	1/10(R6+R8)= 11	2(R6+R8)= 220
Eigenvalues	S1	-1.066	-10.005	-1.004
	S2	-0.904	-0.995	-0.496
Spread of eigenvalues		0.0810	0.8191	0.3387

$$[A] = \begin{bmatrix} -210/m_{10} & 10/m_{11} \\ 10/m_{10} & -110/m_{11} \end{bmatrix}$$

Three different parameter sets of m_{10} and m_{11} have been used. Their eigenvalue distribution regions are shown in Figure 3-4. Obviously the parameter pair in case 1 e the optimal selection among the choices shown.

General n th-order Augmented Subsystems

Experience indicates that many physical systems containing algebraic loops may be adequately handled with only first or second order augmentation. But it is still possible for higher order augmentations to emerge in some complicated systems. A useful rule may be generalized from the discussion above on the first and the second-order cases to the general nth-order case. In general, the augmentation by adding only I or C elements will produce the resolvent [R] and $[M^{-1}]$ or [G] and [K] matrices as follows

$$[A] = -[R][M^{-1}] = -\begin{bmatrix} r_{11} & r_{12} & \cdots\cdots & r_{1n} \\ r_{21} & r_{22} & \cdots\cdots & . \\ . & . & & . \\ . & & . & . \\ . & & \;\;. & . \\ r_{n1} & \cdots\cdots & & r_{nn} \end{bmatrix} \begin{bmatrix} 1/m_1 & \cdots\cdots & 0 \\ . & 1/m_2 & \\ . & . & \\ . & \;\;. & \\ . & & . \\ 0 & \cdots\cdots & 1/m_n \end{bmatrix}$$

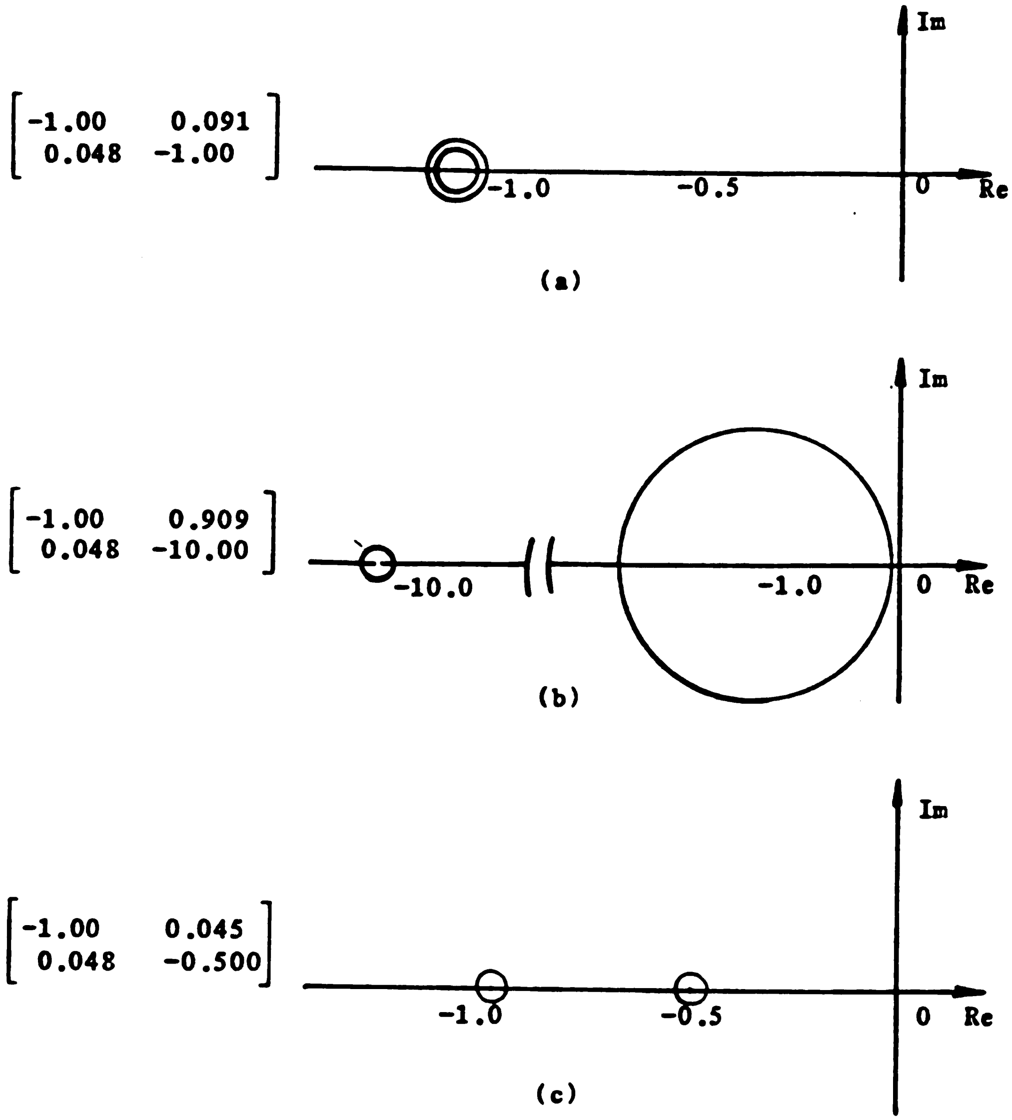


Figure 3-4 The eigenvalue distribution

(a) Case 1

(b) Case 2

(c) Case 3

or

$$[A] = -[G][K] = -\begin{bmatrix} g_{11} & g_{12} & \cdots\cdots & g_{1n} \\ g_{21} & g_{22} & \cdots\cdots & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & & \cdot & \cdot \\ \cdot & & \cdot & \cdot \\ g_{n1} & \cdots\cdots & & g_{nn} \end{bmatrix} \begin{bmatrix} k_1 & \cdots\cdots & & 0 \\ \cdot & k_2 & & \\ \cdot & \cdot & & \\ \cdot & & \cdot & \\ \cdot & & & \cdot \\ 0 & \cdots\cdots & & k_n \end{bmatrix}$$

The general [R] or [G] matrix has following properties as mentioned before:

1) The [R] or [G] matrix is symmetric and positive definite;

2) The diagonal terms r_{ii} or g_{ii} are dominant among the terms in the same row.

The idea that the free parameters should make the diagonal terms a_{11} and a_{22} in the two dimensional product matrix [A] be minus one so that the "spread" of eigenvalues is the minimum, may be extended to a general n-th order augmentation. So the general rule for optimal parameter selection may be stated as that:

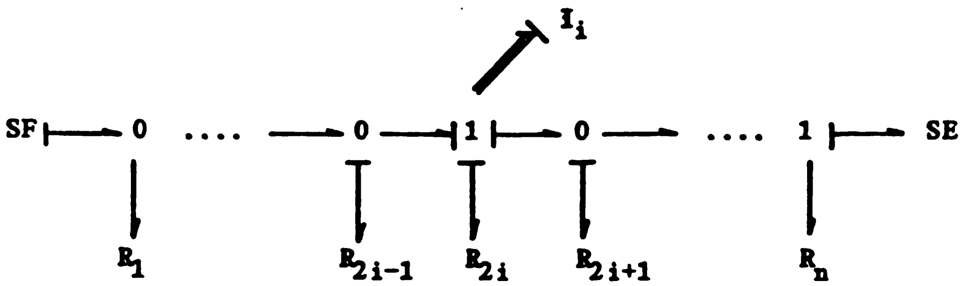
The optimal parameters should make all the diagonal terms a_{ii} in the product matrix [A] be an identical number, more precisely, $m_i = r_{ii}$ for adding only I elements or $k_i = 1/g_{ii}$ for adding only C elements.

The most common junction structure of R-fields is so-called "chain" type 0/1 junction structure (Figure 3-5) in which the junction elements 0 and 1 are arranged in alternation. It may be worth paying a little more attention to it. Obviously every added dynamic element increases the number of eigenvalues of the [A] matrix by one. The additional eigenvalue will be determined principally by the parameter of the dynamic element itself and the values of the dissipative elements in the neighborhood.

The [R] matrix produced from a general n-th-order dynamic augmented static field with 'chain' junction structure by adding I or C elements possesses a banded form:

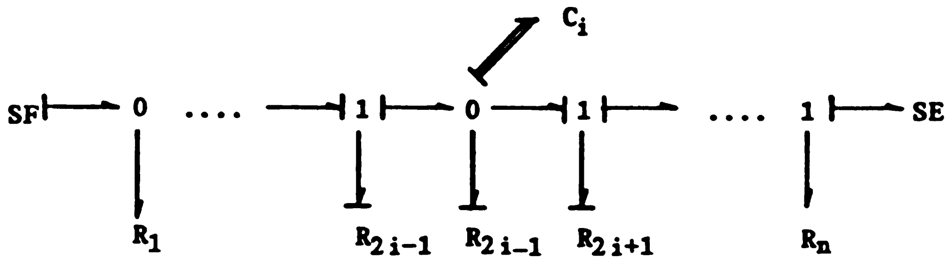
$$
-[R] = \begin{bmatrix}
-(R_1+R_2+R_3) & R_3 & 0 & 0 & 0 & 0 & 0 \\
R_3 & -(R_3+R_4+R_5) & R_5 & 0 & 0 & 0 & 0 \\
0 & . & . & . & . & . & . \\
. & . & . & . & . & . & . \\
. & . & . & R_{i-1} & -(R_{i-1}+R_i+R_{i+1}) & R_{i+1} & . \\
. & . & . & . & . & . & . \\
. & . & . & . & . & . & . \\
0 & . & . & . & . & R_{n-1} & -(R_{n-1}+R_n)
\end{bmatrix}
$$

An explicit rule of parameter selection for "chain" type junction structure may be stated as



$$I_i = R_{2i-1} + R_{2i} + R_{2i+1}$$

(a)



$$C_i = 1/R_{2i-1} + 1/R_{2i} + 1/R_{2i+1}$$

(b)

Figure 3-5 General n-th-order augmentation

(a) By adding I-elements

(b) By adding C-elements

1) for augmentation by adding only I elements, use

$$m_i = R_{2i-1} + R_{2i} + R_{2i+1} \qquad (3.25a)$$

2) for augmentation by adding only C elements, use

$$1/k_i = 1/R_{2i-1} + 1/R_{2i} + 1/R_{2i+1} \qquad (3.25b)$$

where m_i ---- the m parameter of the i-th introduced I element;

k_i ---- the k parameter of the i-th introduced C element;

R_i ---- linear resistance in the field.

An example for general case is shown in Figure 3-6. The [G] and [K] can be derived from the topology of the bond graph model as below

$$-[G][K] = \begin{bmatrix} -(1/R_1+1/R_2+1/R_3+1/R_5) & 1/R_3 & 1/R_5 \\ 1/R_3 & -(1/R_3+1/R_4) & 0 \\ 1/R_5 & 0 & -(1/R_5+1/R_6) \end{bmatrix} \begin{bmatrix} k_{15} & 0 & 0 \\ 0 & k_{16} & 0 \\ 0 & 0 & k_{17} \end{bmatrix}$$

Some numerical results are tabulated in Table 2. It shows that the optimal parameter set is obtained by using the suggested rule.

It has been noticed that for those [R] matrices in which some of the r_{ij} terms ($i = j$) are greater than zero, the eigenvalue "spread" obtained by the general rule is not the minimum. However, employing the general rule still makes the parameter set result in a satisfactary eigenvalue distribution in the s-plane. Although the mathematical proof has not been given, the general rule still can be applied to such [R] matrices. An example is given in Figure 3-7(notice that this case is not of the least augmentation). The [A]

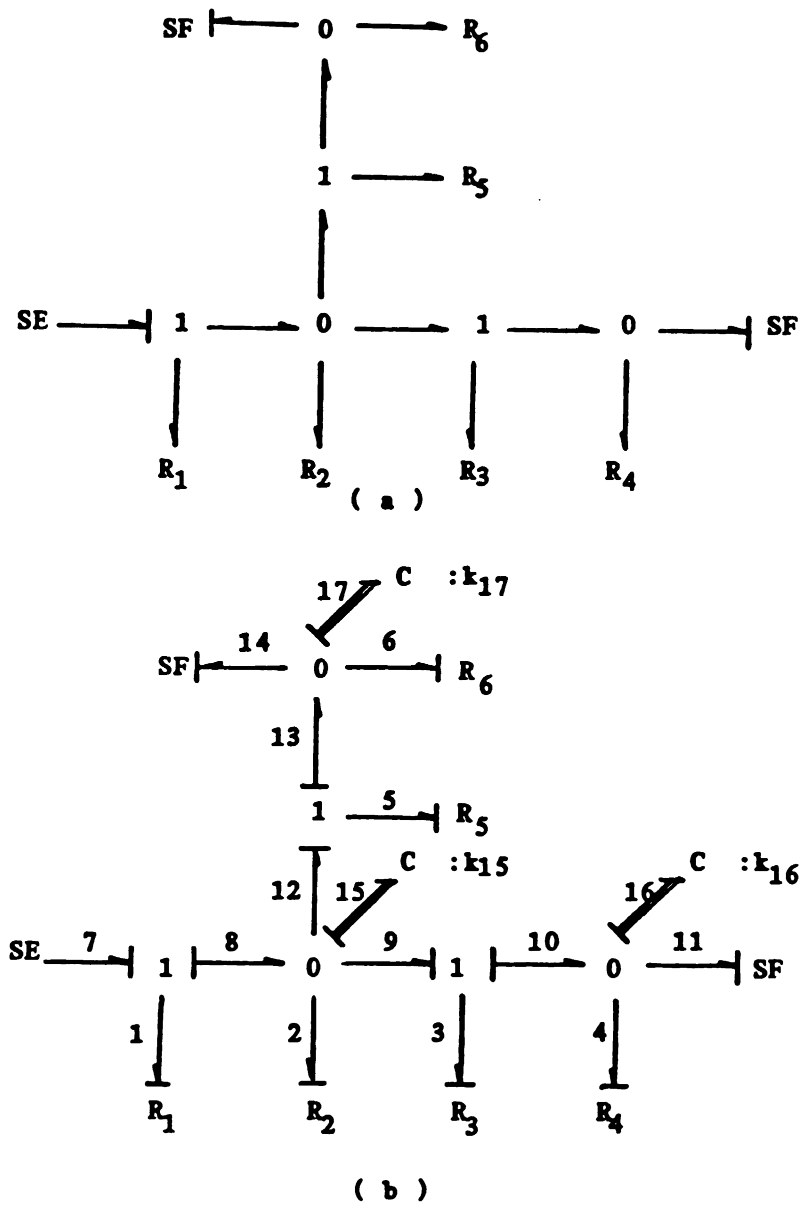


Figure 3-6 An R-field: example 1

(a) Before augmentation

(b) After augmentation

TABLE 3-2 EIGENVALUES OF A THIRD ORDER AUGMENTATION

Parameters of R elements						Free Parameters			Eigenvalues			spread
R1	R2	R3	R4	R5	R6	K15	K16	K17	S1	S2	S3	
						1.0	1.0	1.0	-0.4732	-0.2000	-0.1268	0.5865
						2.5	5.0	5.0	-0.5000	-1.0000	-1.5000	0.5000 *
10.0	10	10	10	10	10	10.0	5.0	5.0	-4.3028	-1.0000	-0.6972	0.7211
						2.5	10.0	5.0	-0.5499	-1.2271	-2.2230	0.6034
						2.5	1.0	5.0	-1.3645	-0.1636	-0.6719	0.7862

* ---- The optimal parameter set.

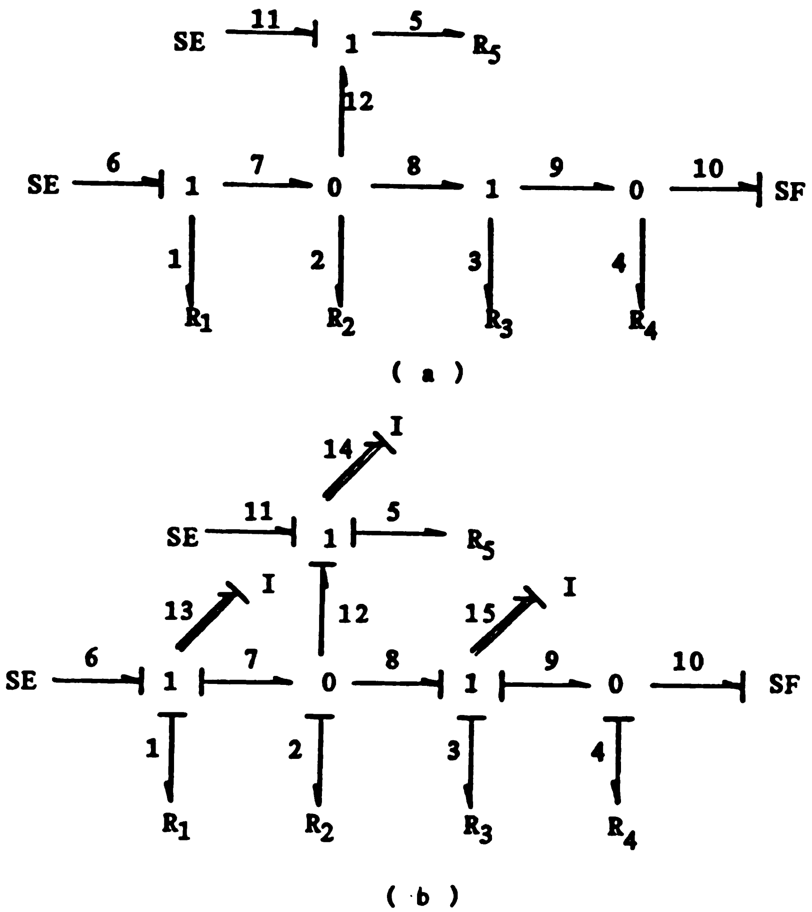


Figure 3-7 An R-field: examlpe 2

a) Before augmentation

b) After augmentation

TSBLE 3-3 EIGENVALUES OF AN ARBITRARY SUBSYSTEM

Case	Parameters of R elements					Free parameters			Eigenvalues			Spread
	R1	R2	R3	R4	R5	M13	M14	M15	S1	S2	S3	
I	10	10	10	10	10	1.0	1.0	1.0	-44.142	-10.000	-15.858	0.6306
						30.0	20.0	20.0	-0.500	-0.621	-1.879	0.5796 *
						40.0	20.0	20.0	-0.500	-0.500	-1.750	0.5556
						50.0	20.0	20.0	-0.416	-0.500	-1.684	0.6042
						30.0	25.0	20.0	-0.438	-0.607	-1.755	0.6005
II	10	50	10	10	10	1.0	1.0	1.0	-16.52	-10.00	-163.5	0.8847
						70.0	60.0	60.0	-0.248	-0.167	-2.584	0.8788 *
						100.0	60.0	60.0	-0.193	-0.167	-2.341	0.8567
						200.0	60.0	60.0	-0.109	-0.167	-2.075	0.9006
						70.0	100.0	60.0	-2.242	-0.121	-0.237	0.8970
III	10	2	10	10	10	1.0	1.0	1.0	-22.90	-10.00	-13.10	0.3920
						21.0	11.0	11.0	-1.150	-0.909	-0.942	0.1167 *
						42.0	11.0	11.0	-.491	-0.909	-1.100	0.3824
						21.0	22.0	11.0	-1.04	-2.03	-0.934	0.3692
IV	10	10	10	10	50	1.0	1.0	1.0	-3.00	-6.646	-1.354	0.6614
						30.0	60.0	20.0	-1.628	-0.786	-0.586	0.4705 *
						30.0	30.0	20.0	-0.589	-2.308	-1.103	0.5936
						30.0	60.0	40.0	-1.376	-0.354	-0.771	0.5909

matrix is derived as below

$$[A] = \begin{bmatrix} -(R_2+R_3+R_4) & -R_2 & R_2 \\ -R_2 & -(R_2+R_5) & R_2 \\ R_2 & R_2 & -(R_1+R_2) \end{bmatrix} \begin{bmatrix} 1/m_{13} & 0 & 0 \\ 0 & 1/m_{14} & 0 \\ 0 & 0 & 1/m_{15} \end{bmatrix}$$

Some numerical computational results on the eigenvalue "spread" are tabulated in Table 3. It shows the parameter selection by using the general rule is rather close to the possible best one; therefore it is acceptable.

The general rule for a "chain" type structure has been applied to a number of arbitrary [R] matrices with different dimensions and numerical conditions. Figure 3-8 represents a third-order augmented R-field by adding three I elements which introduce three free parameters: m_1, m_2 and m_3. Three different cases have been investigated(Table 4). For the first case the parameter values of all R elements are equal or close each other; we call this the normal case. For the second case the parameter values of R elements are different in such a way that the R elements attached to the junctions to which the dynamicizers are added have relatively large parameter values compared with their neighbors. The third case is just the opposite of case 2.

In every case, if we choose parameter values for m_1, m_2 and m_3 according to the general rule obtained before, then the "spread" ρ

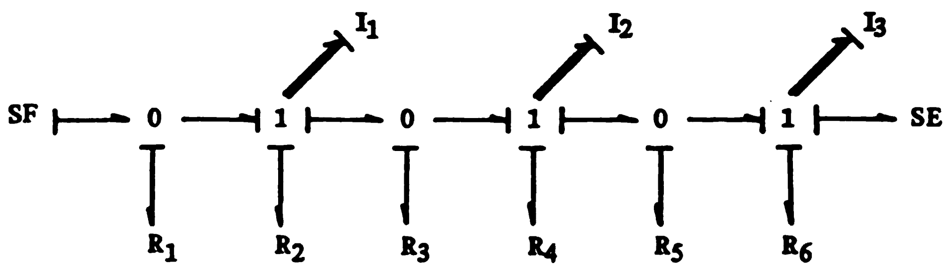


Figure 3-8 A third-order augmentation

TABLE 3-4 EIGENVALUES OF A SUBSYSTEM WITH CHAIN STRUCTURE

Conditions	Parameters of elements						Param. of added elem.			Eigenvalues			Spread
	R1	R2	R3	R4	R5	R6	M1	M2	M3	S1	S2	S3	
Equal	10	10	10	10	10	10	30.0	30.0	20.0	-1.000	-0.473	-1.527	0.5267 *
							60.0	30.0	20.0	-0.356	-1.439	-0.705	0.6033
							30.0	60.0	20.0	-1.000	-0.301	-1.199	0.5983
							30.0	30.0	40.0	-1.383	-0.334	-0.783	0.6115
Favorable	10	100	20	50	5	60	130.0	75.0	65.0	-1.000	-0.785	-1.215	0.2148 *
							260.0	75.0	65.0	-0.948	-0.461	-1.091	0.4057
							130.0	150.0	65.0	-1.042	-0.458	-1.000	0.3900
							130.0	75.0	130.0	-1.204	-0.494	-0.802	0.4183
Adverse	10	5	20	5	40	5	35.0	65.0	45.0	-1.000	-1.850	-0.150	0.8502 *
							70.0	65.0	45.0	-0.500	-1.401	-0.099	0.8681
							35.0	130.0	45.0	-0.564	-1.345	-0.091	0.8727
							35.0	65.0	90.0	-2.305	-0.111	-1.084	0.9082

* --- M1 = R1 + R2 + R3
M2 = R3 + R4 + R5
M3 = R5 + R6

will be the smallest. Furthermore, among the three cases the smallest one is achieved from case 2. Case 2 is called one with favorable condition. Generally we hope the R-fields are in such favorable condition which will make the local integration be more efficient. The opposite case shows a much worse result, so it is called the case with adverse condition.

3.3 Parameter Selection in Nonlinear R-fields

Linearization of R Elements

In a nonlinear dissipative problem the "parameters" of nonlinear R elements vary. The eigenvalues of the associated [A] matrix change during simulation. Consequently the parameters of the added dynamic elements may need adjustment during simulation. Otherwise they may cause integration in the local time scale to be unstable or lead to a considerable error.

One technique for applying previous results to nonlinear system problems is linearization. At selected times and states, the equivalent R-field parameters can be found by linearizing the dissipative element characteristics. That is, we find the local tangent

$$R = de/df \tag{3.27}$$

where

$$e = g_r(f). \tag{3.28}$$

Alternatively a chordal approximation based on the global e and f can be used, namely,

$$R = e/f \tag{3.29}$$

For example, a nonlinear dissipasive R element possesses the following constitutive function

$$e = 0.15f^{1.2} \tag{3.30}$$

By using the chordal definition for the parameter of a linear R element, the equivalent parameter of R would be calculated by

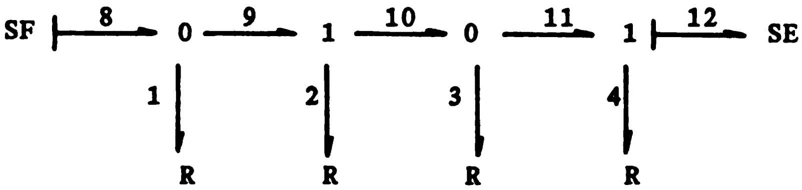
$$R_c = e \,/\, f = 0.15f^{1.2}/f = 0.15f^{0.2} \tag{3.31}$$

whereas the tangent approximation yields

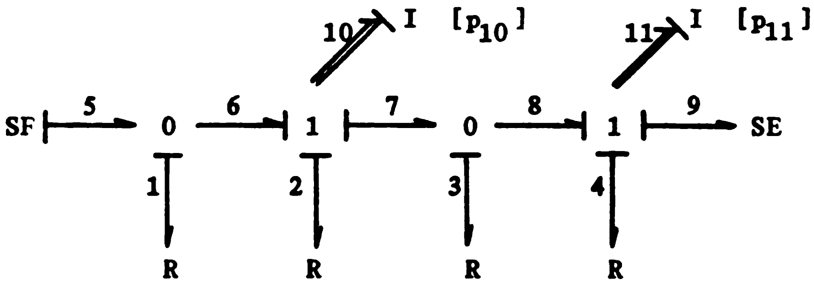
$$R_t = de/df = 0.18f^{0.2} \;. \tag{3.32}$$

Reference R-Matrix for Integration Control

Partitioned static and dynamic fields are shown in Figure 2-2. One of the possible augmented subsystem is proposed in Figure 3-9, with the state vector, X, the input vector, U, and the output vector, V. The constitutive dissipation functions are



(a)



$$X = \begin{bmatrix} p_{10} \\ p_{11} \end{bmatrix} \qquad U = \begin{bmatrix} f_5 \\ e_9 \end{bmatrix} \qquad V = \begin{bmatrix} f_9 \end{bmatrix}$$

(b)

Figure 3-9 A nonlinear R-field

(a) Before augmentation

(b) After augmentation

$$e_1 = 2.0\mathrm{SIGN}(f_1)f_1^{1.2} \tag{3.33a}$$

$$e_2 = \mathrm{SIGN}(f_2)f_2^2 \tag{3.33b}$$

$$e_3 = 3\mathrm{TANH}(f_3) \tag{3.33c}$$

$$e_4 = 2.0f_4 \tag{3.33d}$$

The equivalent parameters of the R elements linearized by the chordal approximation would be

$$R_1 = 2.0\mathrm{SIGN}(f_1)f_1^{0.2} \tag{3.34a}$$

$$R_2 = \mathrm{SIGN}(f_2)f_2 \tag{3.34b}$$

$$R_3 = 3\mathrm{TANH}(f_3)/f_3 \tag{3.34c}$$

$$R_4 = 2.0 \tag{3.34d}$$

The state equations for the nonlinear subsystem can be derived from the bond graph model as usual. Remember that the added I elements are linear with constant parameters m_{10} and m_{11}.

$$\dot{p}_{10} = 2.0\mathrm{SIGN}(f_s-p_{10}/m_{10})(f_s-p_{10}/m_{10})^{1.2} - \mathrm{SIGN}(p_{10}/m_{10})(p_{10}/m_{10})^2 - 3\ \mathrm{TANH}(p_{10}/m_{10} - p_{11}/m_{11}) \tag{3.35a}$$

$$\dot{p}_{11} = 3\ \mathrm{TANH}(p_{10}/m_{10} - p_{11}/m_{11}) - 2.0p_{11}/m_{11} - e_9 \tag{3.35b}$$

If we arrange the above equations as follows

$$\dot{p}_{10} = \frac{2.0(f_s - p_{10}/m_{10})^{1.2}}{(f_s - p_{10}/m_{10})}(f_s - p_{10}/m_{10}) - \frac{(p_{10}/m_{10})^2}{(p_{10}/m_{10})}(p_{10}/m_{10})$$

$$- \frac{3\mathrm{TANH}(p_{10}/m_{10} - p_{11}/m_{11})}{(p_{10}/m_{10} - p_{11}/m_{11})}(p_{10}/m_{10} - p_{11}/m_{11}) \tag{3.36a}$$

$$\dot{p}_{11} = \frac{3\,\mathrm{TANH}(p_{10}/m_{10} - p_{11}/m_{11})}{(p_{10}/m_{10} - p_{11}/m_{11})}(p_{10}/m_{10} - p_{11}/m_{11}) - 2.0p_{11}/m_{11} - e_g \tag{3.36b}$$

Further manipulating yields

$$\dot{p}_{10} = -\left[2.0(f_s - p_{10}/m_{10})^{0.2} + p_{10}/m_{10} + E\right]p_{10}/m_{10}$$

$$+ E\,p_{11}/m_{11} + 2.0(f_s - p_{10}/m_{10})^{0.2}f_s \tag{3.37a}$$

$$\dot{p}_{11} = E\,p_{10}/m_{10} - [\,E + 2.0\,]p_{11}/m_{11} - e_g \tag{3.37b}$$

where $E = 3\mathrm{TANH}(p_{10}/m_{10} - p_{11}/m_{11})/(p_{10}/m_{10} - p_{11}/m_{11})$

Comparing the terms in the brackets with the equivalent parameters of these R elements defined by the chordal approximation it turns out that they are identical. At a certain global time the state equations in matrix form would be

$$\begin{bmatrix}\dot{p}_{10}\\ \dot{p}_{11}\end{bmatrix} = \underbrace{\begin{bmatrix}-(R_1+R_2+R_3) & R_3\\ R_3 & -(R_3+R_4)\end{bmatrix}\begin{bmatrix}1/m_{10} & 0\\ 0 & 1/m_{11}\end{bmatrix}}_{[A]}\begin{bmatrix}p_{10}\\ p_{11}\end{bmatrix} + \begin{bmatrix}(f_s - p_{11}/m_{11}{}^{0.2}) & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}f_s\\ e_g\end{bmatrix} \begin{matrix}(3.38a)\\ (3.38b)\end{matrix}$$

where R_1, R_2, R_3, R_4 are defined by Eqs.(3.34a,b,c,d).

With the equivalent linear resistances, an estimate of the instantaneous dynamics of the nonlinear system can be obtained by the extraction of the eigenvalues from the above [A] matrix. The [R] matrix is analogous to the one of linear case described earlier. Although this analogous [R] matrix can not be used for direct simulation, it may serve to predict the local dynamics in the two-time-scale integration technique. It will help in parameter selection for the dynamicizers and the local integration controls. Therefore, such an [R] matrix is named the reference [R] matrix.

By using the instantaneous values for the bond variables or state variables available from the previous global time step, a set of instantaneous equivalent linear resistances can be computed. In this example, at a certain global time step the state variables p_{10} and p_{11} are known as $\tilde{p}_{10}$ and $\tilde{p}_{11}$. The reference [R] matrix would be calculated as follows

$$[R] = - \begin{bmatrix} [(\tilde{f}_s - \tilde{p}_{11}/\tilde{m}_{11})^{0.2} + \tilde{p}_{11}/\tilde{m}_{11} + E\,] & E \\ E & -[\,E + 2.0\,] \end{bmatrix}$$

where $\tilde{f}_s$, $\tilde{p}_{10}$, $\tilde{p}_{11}$, $\tilde{m}_{10}$ and $\tilde{m}_{11}$ are the values at the previous global time step, and E is as in Eq.(3.37).

In many cases a set of R parameters could be used for several global time steps before re-evaluation would become necessary. This must be judged from the rate at which the local field input variables are changing.

At steady state the equivalent resistance parameters virtually may be the functions of only the input vector to the R-field; namely

$$R_i = \gamma_i(U) \tag{3.39}$$

If the change of the input vector is not too large the equivalent resistance R_i will not change dramatically, provided the nonlinear dissipative element is well-behaved; for instance, the effort is a low order power of flow. For many physical systems, the nonlinear dissipative elements possess fairly moderate characteristics. The changes in equivalent parameters over the dynamic range of interest are not very large. In the mechanical systems such nonlinear dissipation effects come from static friction, columb friction and other nonlinear frictions which have small degree of nonlinearity. Therefore, the reference [R] matrix will not change the local dynamics too much and will not require the freqent re-selection of added dynamicizers.

The process described in the preceding paragraphs can be repeated as the global time variable increases; that is, the free parameters could be reselected intermittently throughout the global simulation in prescribed fashion.

The technique of parameter selection for the n-th-order nonlinear subsystem is summarized below:

1) At $t=t_0$, set all free parameters to be an arbitrary constant, say, unity, set arbitrary initial conditions for the added dynamic elements and choose a small time step and a long duration for local time scale, then integrate the subsystem to steady state.

2) Compute the instantaneous equivalent linear resistances for each nonlinear dissipative element in the augmented subsystem.

3) Form the reference [R] or [G] matrix.

4) Select the optimal free parameters following the general rule.

5) Compute the eigenvalues of $-[G][K]$ or $-[R][M^{-1}]$ matrix, reset the local time scale.

7) Integrate the subsystem to the steady state in the new local time scale.

8) Compute the output vector of the subsystem.

9) Continue with the global integration forward one step.

10) Repeat the parameter selection process according to the error control criteria.

4 NUMERICAL EXAMPLE

To illustrate the effectiveness of the bond graph augmentation method discussed in the preceding chapters, a demonstration example is presented in Figure 4-1. The subroutine PART identified the system and partitioned it into one dynamic field and one static field. Then subroutine AUGMEN automatically determined which nodes should be augmented and what kind of dynamicizers should be added. After the augmentation process was finished, the augmentation information was printed out for inspection (Figure 4-2). The augmented subsystem and the dynamic subsystem are shown in Figure 4-3. The bonds and the nodes in both subsystems have been renumbered regarding the initial descending numbering sequence.

For different purposes a linear case and a nonlinear case have been simulated. The existing program for linear systems can be used as an examiner.

4.1 Linear Case

If the all dissipative elements in the R-field are linear, then the state equations of the augmented subsystem are

$$\begin{bmatrix} \dot{p}_{10} \\ \dot{p}_{11} \end{bmatrix} = \begin{bmatrix} -(R_3+R_4) & R_3 \\ R_3 & -(R_1+R_2+R_3) \end{bmatrix} \begin{bmatrix} 1/m_{10} & 0 \\ 0 & 1/m_{11} \end{bmatrix} \begin{bmatrix} p_{10} \\ p_{11} \end{bmatrix} + \begin{bmatrix} 0 & -1 \\ R_1 & 0 \end{bmatrix} \begin{bmatrix} f_8 \\ e_9 \end{bmatrix} \qquad (4.1)$$

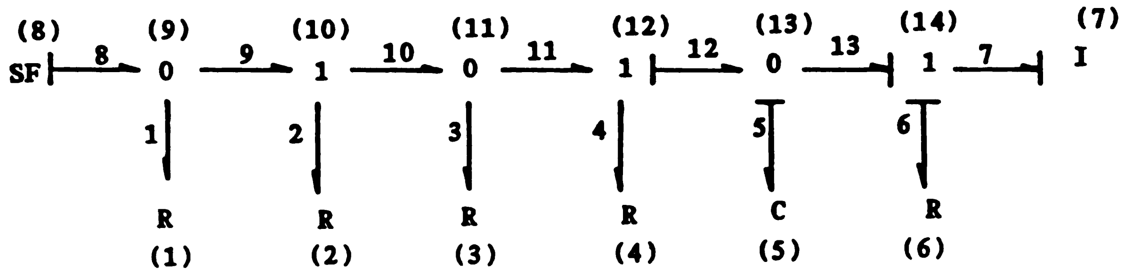
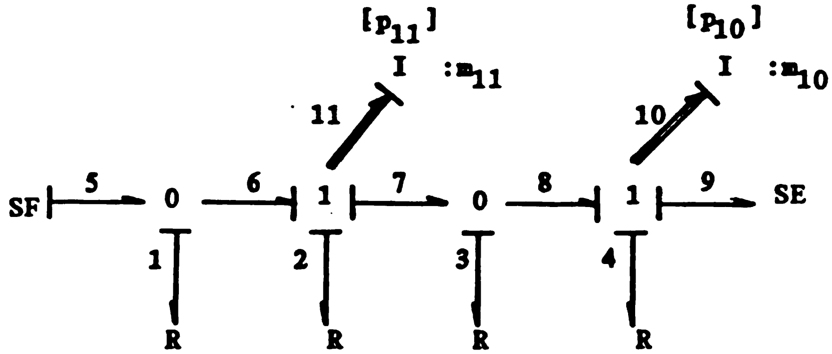


Figure 4-1 The global bond graph model for the example

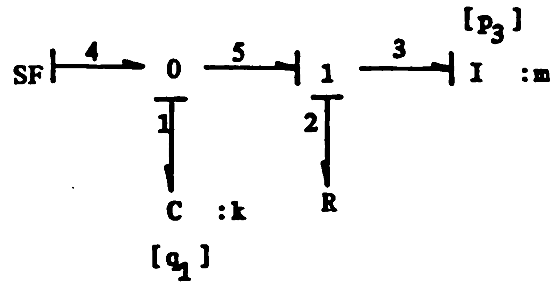
```
THERE ARE  1  R-FIELDS IN THE GIVEN SYSTEM
R FIELD NUMBER      1 :
NODES
             9  O          8   1   9
             8  SF         8
             1  R          1
            10  1          9   2  10
             2  R          2
            11  O         10   3  11
             3  R          3
            12  1         11   4  12
             4  R          4
            13  SE        12
PORTS :
            12
THERE ARE  O  JUNCTION STRUCTURE COMPLEXES
THERE ARE  1 DYNAMIC FIELDS IN THE GIVEN SYSTEM
DYNAMIC FIELD NUMBER      1 :
NODES :
            13  O         12   5  13
            12  SF        12
             5  C          5
            14  1         13   6   7
             6  R          6
             7  I          7
PORTS :
            12
ADD C/I ELEMENT TO THE ACTING JUNCTIONS
   ADD  I   ELEMENT TO NEW NODE  8  (THE OLD NODE 12  )
ADD C/I ELEMENT TO THE SINGLE-TYPE JUNCTINS
   ADD  I   ELEMENT TO NEW NODE  4  (THE OLD NODE 10  )

 THERE ARE   2 STATE VARIABLES
       AND   2 INPUT VARIABLES.
 THE STATE VECTOR...
   X  1 = P 10
   X  2 = P 11

 THE INPUT VECTOR...
   U  1 = F  5
   U  2 = E  9
```

Figure 4-2 The Augmentation Result for the Example



$$X^s = \begin{matrix} p_{10} \\ p_{11} \end{matrix} \qquad U^s = \begin{matrix} f_5 \\ e_9 \end{matrix} \qquad V^s = f_9$$

(a)



(b)

Figure 4-3 The subsystems

a) Augmented subsystem

b) Dynamic subsystem

If all the elements in the dynamic field are linear, the state equations of the dynamic sybsystem are

$$\dot{q}_1 = f_4 - p_3/m_3 \tag{4.2a}$$

$$\dot{p}_3 = k_1 q_1 - R_2 p_3/m_3 \tag{4.2b}$$

Assume the parameters of the elements are

$R_1 = R_1^s = 10.0$ $\quad$ $R_2 = R_2^s = 20.0$

$R_3 = R_3^s = 5.0$ $\quad$ $R_4 = R_4^s = 10.0$

$k_5 = k_1^d = 20.0$ $\quad$ $R_6 = R_1^d = 5.0$

$m_7 = m_3^d = 2.0$

where the superscripts s and d denote the augmented subsystem and the dynamic subsystem, respectively. The [R] matrix of the augmented subsystem is

$$[R] = \begin{bmatrix} -15.0 & 5.0 \\ 5.0 & -35.0 \end{bmatrix}$$

Acoording to the general rule we choose 15 and 35 for parameters m_{10} and m_{11}, respectively, so the [A] matrix appears

$$[A] = \begin{bmatrix} -1.0 & 0.1429 \\ 0.3333 & -1.0 \end{bmatrix}$$

The eigenvalues of the [A] matrix are

$$s_1 = -0.7818$$

$$s_2 = -1.2182$$

We may set the local time scale as follows

$\Delta t = 1/(3s_{min}) = 1/(3*1.2182) = 0.273$ sec

$t_f = 5/s_{max} = 5/0.7818 = 6.395$ sec

If you want to change the scale by a factor n, you may change the parameters m_{10} and m_{11} by the factor. For instance, let m_{10} and m_{11} be 1.5 and 3.5, respectively, then

$s_1 = -7.818$

$s_2 = -12.182$

then the time interval and integral duration will be

$\Delta t = 0.0275$ sec

$t_f = 0.6395$ sec

In the execution we set $\Delta t = 0.025$ sec and $t_f = 0.65$ sec.

The global time scale may also be determined from the eigenstructure, $s_{1,2} = -1.95 \pm 3.114j$, as follows

$\Delta T = 0.05 - 0.10$ sec

$T_f = 5.0$ sec

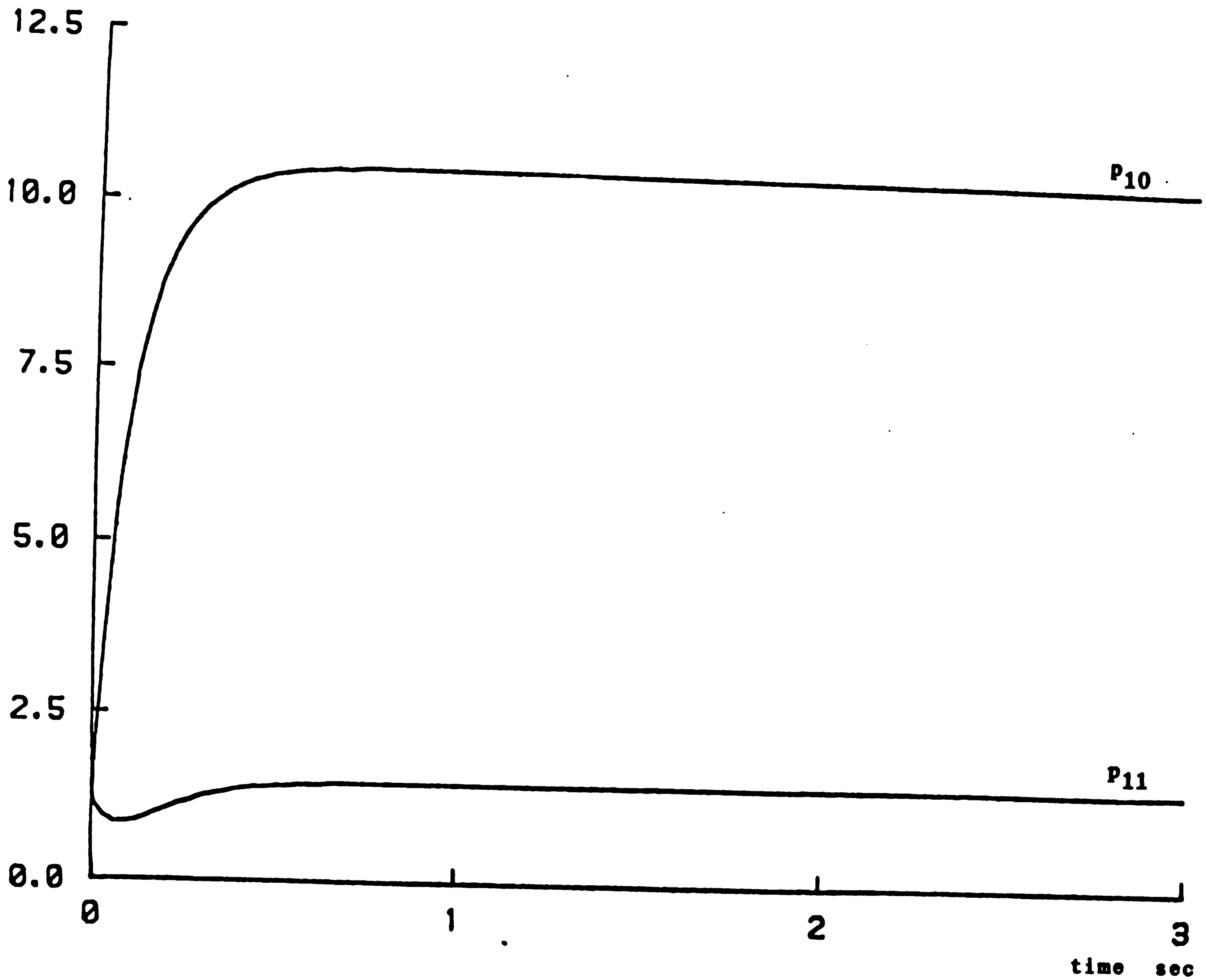
Suppose that all the initial conditions are equal to zero at T = 0, and the flow input from source SF8 is a constant, say, $f_8^s = 10.0$.

Since T = 0, $q_1 = 0$, and $p_3 = 0$, therefore $e_9^s = e_4^d = 0$. These two subsystems interact through bond 12 in the initial system model,DOhat is the output of the augmented subsystem, f_9^s, is the input to the dynamic subsystem, f_4^d, while the output of the latter, e_4^d, is the input to the former, e_9^s. Besides the input to a augmented subsystem at present global time step from the adjacent dynamic subsystem would be the output of the latter at the just previous step. So, at global time T = 0.05 sec, the input $e_9^s = 0$. Integrating in the local time scale that has been set earlier yields the output f_9^s at T = 0.05,

$$f_9^s = \tilde{p}_{10}/m_{10}. \tag{4.3}$$

It will be the input to the dynamic subsystem at T = 0.05. Figure 4-4 shows the local time scale integration process at the global time T = 0.05, and Figure 4-5 displays the next local integration. The dynamic subsystem then integrates one time step in the global time scale. The steady states of the augmented subsystem serve as initial conditions for next local-time-scale integration and also the states of dynamic subsystems become the initial condition for the next global-time-scale integration. These two subsystems interact in this manner throughout the entire global time so that the system simulation can be realized.

To demonstrate that this procedure can lead to correct results, we have used the DIFFEQ[12] package that can solve either linear and nonlinear differential equations. The numerical results and a plot are shown in Figure 4-6 and Figure 4-7, respectively. Comparing these with the results obtained by the ENPORT-5[13] package (Figure 4-8 and Figure 4-9) shows that they are almost identical. The small differ-



Figuer 4-4 Local-time integration over the first global-time step

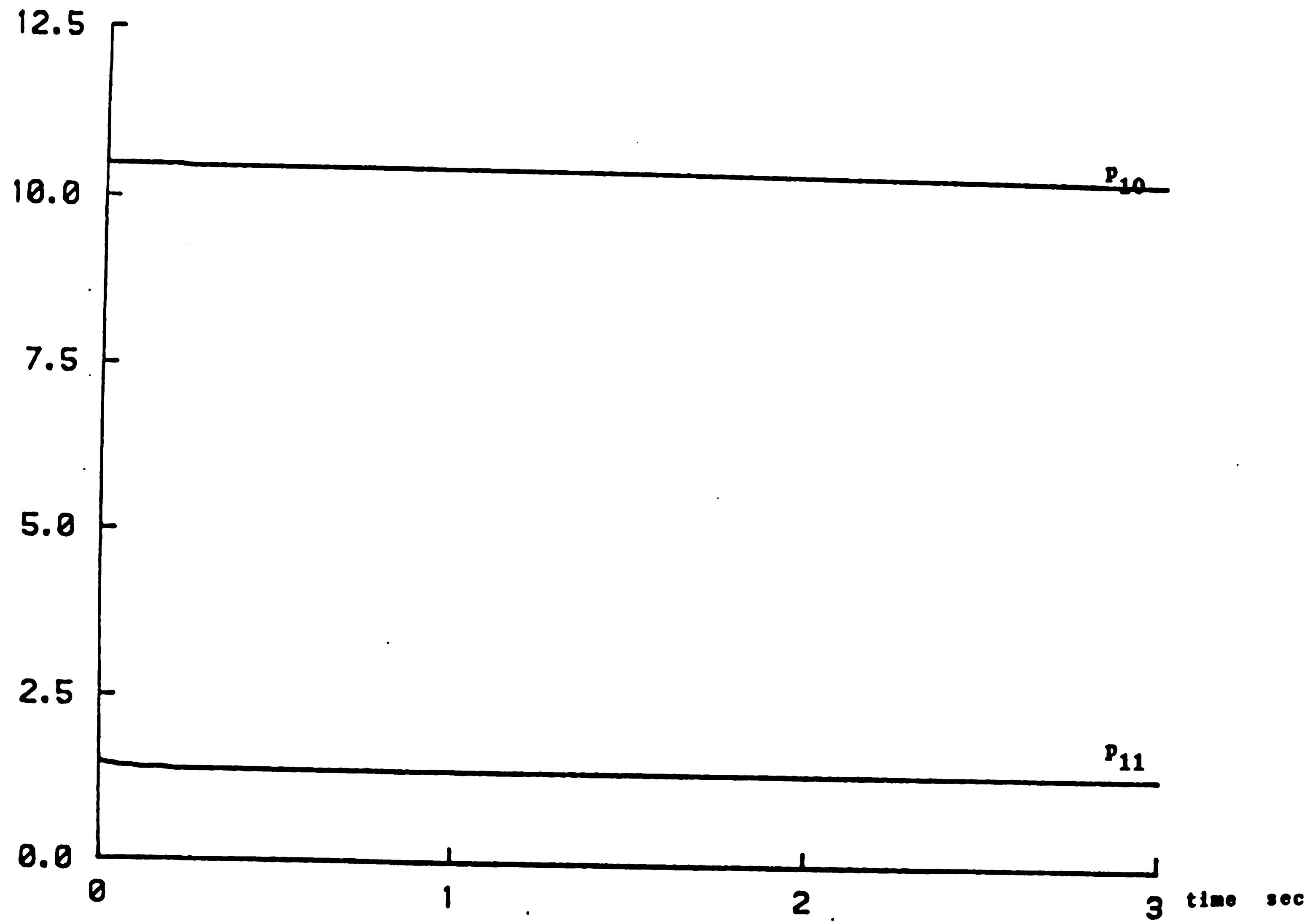


Figure 4-5 local-time integration over the second global-time step

TIME	P10	P11	E9
0.5000E-01	0.4980E-01	0.2394E-01	0.9960E+00
0.1000E+00	0.9497E-01	0.8974E-01	0.1899E+01
0.1500E+00	0.1349E+00	0.1878E+00	0.2698E+01
0.2000E+00	0.1693E+00	0.3093E+00	0.3386E+01
0.2500E+00	0.1980E+00	0.4462E+00	0.3961E+01
0.3000E+00	0.2212E+00	0.5911E+00	0.4424E+01
0.3500E+00	0.2391E+00	0.7384E+00	0.4782E+01
0.4000E+00	0.2521E+00	0.8829E+00	0.5042E+01
0.4500E+00	0.2606E+00	0.1020E+01	0.5213E+01
0.5000E+00	0.2652E+00	0.1148E+01	0.5305E+01
0.5500E+00	0.2665E+00	0.1263E+01	0.5329E+01
0.6000E+00	0.2650E+00	0.1365E+01	0.5299E+01
0.6500E+00	0.2612E+00	0.1452E+01	0.5224E+01
0.7000E+00	0.2554E+00	0.1524E+01	0.5108E+01
0.7500E+00	0.2489E+00	0.1582E+01	0.4979E+01
0.8000E+00	0.2413E+00	0.1627E+01	0.4827E+01
0.8500E+00	0.2333E+00	0.1659E+01	0.4666E+01
0.9000E+00	0.2252E+00	0.1680E+01	0.4504E+01
0.9500E+00	0.2173E+00	0.1690E+01	0.4346E+01
0.1000E+01	0.2098E+00	0.1692E+01	0.4196E+01
0.1050E+01	0.2029E+00	0.1687E+01	0.4057E+01
0.1100E+01	0.1966E+00	0.1676E+01	0.3933E+01
0.1150E+01	0.1911E+00	0.1661E+01	0.3822E+01
0.1200E+01	0.1864E+00	0.1643E+01	0.3728E+01
0.1250E+01	0.1825E+00	0.1623E+01	0.3650E+01
0.1300E+01	0.1794E+00	0.1602E+01	0.3588E+01
0.1400E+01	0.1752E+00	0.1561E+01	0.3504E+01
0.1500E+01	0.1736E+00	0.1524E+01	0.3471E+01
0.1600E+01	0.1739E+00	0.1494E+01	0.3478E+01
0.1700E+01	0.1754E+00	0.1473E+01	0.3508E+01
0.1800E+01	0.1776E+00	0.1459E+01	0.3551E+01
0.1900E+01	0.1800E+00	0.1453E+01	0.3599E+01
0.2000E+01	0.1822E+00	0.1452E+01	0.3644E+01

Figure 4-6 Numerical result of linear case obtained by the two-time-scale integration

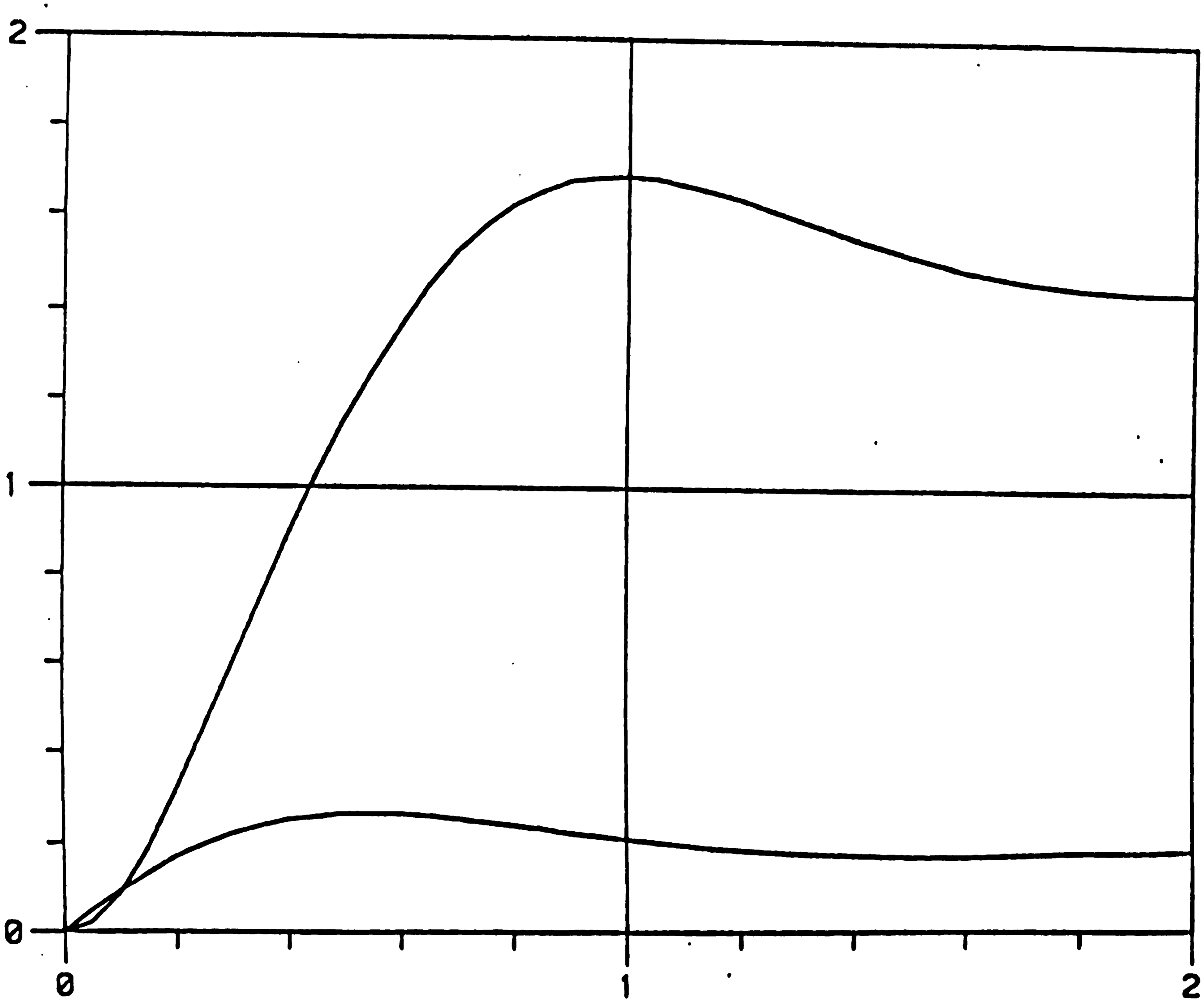


Figure 4-7 Dynamic response obtained by the two-time-scale integration

TIME	P10	P11
0.000E 00	0.00000E 00	0.00000E 00
0.500E-01	0.48095E-01	0.23386E-01
0.100E 00	0.91862E-01	0.87213E-01
0.150E 00	0.13073E 00	0.18237E 00
0.200E 00	0.16437E 00	0.30038E 00
0.250E 00	0.19269E 00	0.43361E 00
0.300E 00	0.21577E 00	0.57527E 00
0.350E 00	0.23382E 00	0.71954E 00
0.400E 00	0.24718E 00	0.86155E 00
0.450E 00	0.25629E 00	0.99736E 00
0.500E 00	0.26160E 00	0.11239E 01
0.550E 00	0.26364E 00	0.12390E 01
0.600E 00	0.26292E 00	0.13411E 01
0.650E 00	0.25994E 00	0.14294E 01
0.700E 00	0.25521E 00	0.15036E 01
0.750E 00	0.24918E 00	0.15640E 01
0.800E 00	0.24226E 00	0.16112E 01
0.850E 00	0.23482E 00	0.16461E 01
0.900E 00	0.22718E 00	0.16697E 01
0.950E 00	0.21961E 00	0.16834E 01
0.100E 01	0.21232E 00	0.16885E 01
0.105E 01	0.20550E 00	0.16864E 01
0.110E 01	0.19927E 00	0.16784E 01
0.115E 01	0.19371E 00	0.16658E 01
0.120E 01	0.18887E 00	0.16497E 01
0.125E 01	0.18478E 00	0.16314E 01
0.130E 01	0.18143E 00	0.16117E 01
0.135E 01	0.17878E 00	0.15916E 01
0.140E 01	0.17680E 00	0.15716E 01
0.145E 01	0.17543E 00	0.15524E 01
0.150E 01	0.17459E 00	0.15345E 01
0.155E 01	0.17423E 00	0.15181E 01
0.160E 01	0.17427E 00	0.15035E 01
0.165E 01	0.17464E 00	0.14908E 01
0.170E 01	0.17526E 00	0.14801E 01
0.175E 01	0.17608E 00	0.14713E 01
0.180E 01	0.17703E 00	0.14644E 01
0.185E 01	0.17806E 00	0.14592E 01
0.190E 01	0.17912E 00	0.14556E 01
0.195E 01	0.18019E 00	0.14535E 01
0.200E 01	0.18121E 00	0.14526E 01

Figure 4-8 Numerical result obtained by the ENPORT-5

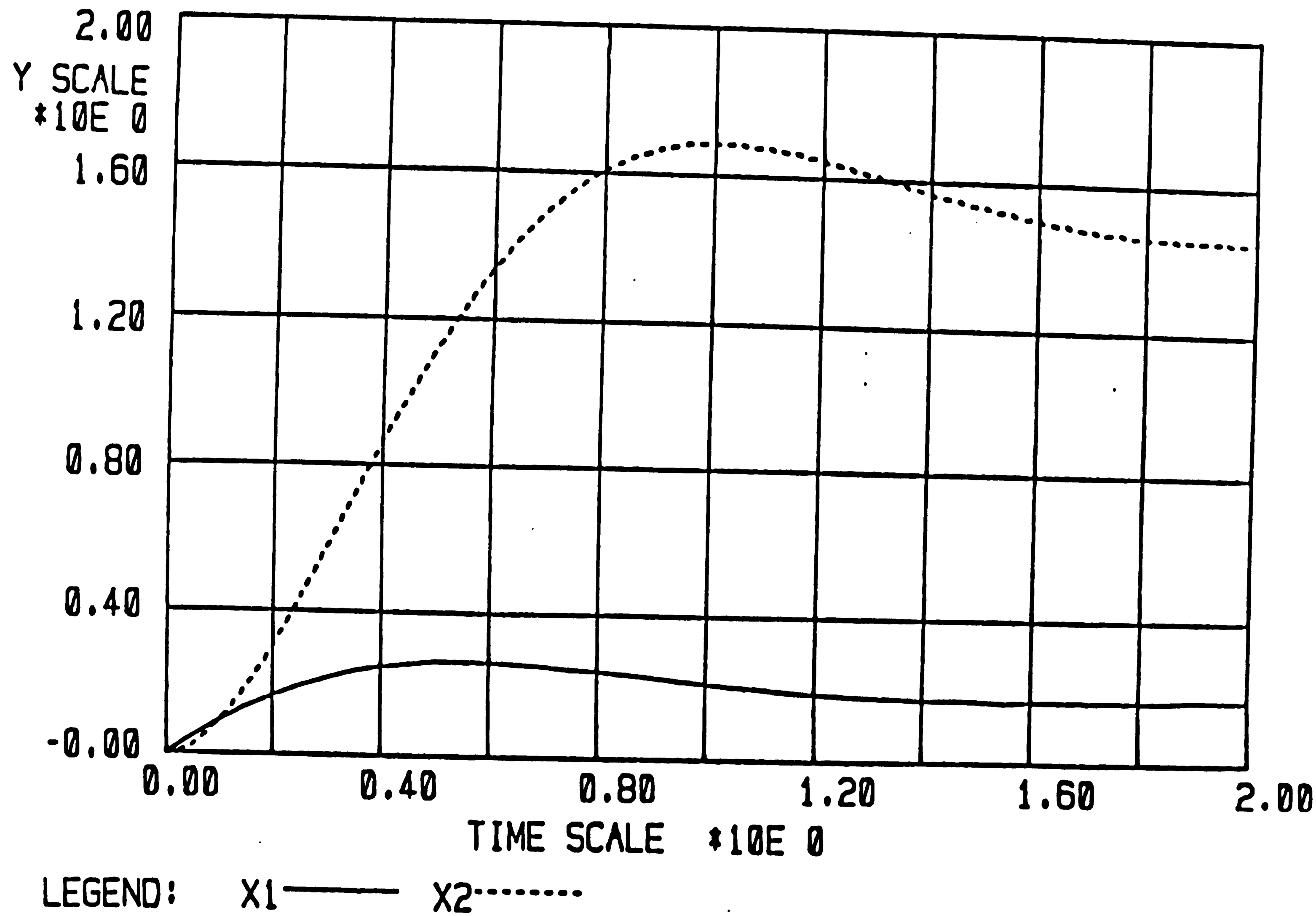


Figure 4-9 Dynamic response obtained by the ENPORT-5

ence (about 1.0 - 2.0 %) comes from the different algorithms employed in these two programs. It, however, has clearly illustrated how the two-time-scale integration technique works and how well.

4.2 Nonlinear Case

If some of the dissipative elements in the augmented subsystem are nonlinear, the simulation process will follow the steps summarized in the preceding chapter. Suppose the dissipation functions are the same as Eqs.(3.34a,b,c,d). The state equations are rewritten as below

$$\dot{p}_{10} = 3\mathrm{TANH}(p_{11}/m_{11}-p_{10}/m_{10}) - 2.0p_{10}/m_{10} - e_9 \qquad (4.5a)$$

$$\dot{p}_{11} = 2.0\mathrm{SIGN}(f_s-p_{11}/m_{11})(f_s-p_{11}/m_{11})^{1.2} - \mathrm{SIGN}(p_{10}/m_{10})(p_{10}/m_{10})^2$$
$$- 3\mathrm{TANH}(p_{11}/m_{11}-p_{10}/m_{10}) \qquad (4.5b)$$

The reference [R] matrix may be

$$\begin{array}{cc} & \begin{array}{cc} -(R_3+R_4) & R_3 \end{array} \\ [R] = & \begin{bmatrix} -[\,E + 2.0\,] & E \\ E & -[(f_s-p_{10}/m_{10})^{0.2}+p_{10}/m_{10}+ E\,] \end{bmatrix} \\ & \begin{array}{cc} R_3 & -(R_1+R_2+R_3) \end{array} \end{array}$$

where $E = 3\mathrm{TANH}(p_{11}/m_{11} - p_{10}/m_{10})/(p_{11}/m_{11} - p_{10}/m_{10})$

The initial parameters m_{10}, m_{11} and the initial conditions p_{10}, p_{11} may be any reasonable numbers as long as there is no zero divide implied, say, m_{10} = 1.0, m_{11} = 1.0, p_{10} = 1.0 and p_{11} = 2.0.

Choose $\Delta t = 0.01$ and $t_f = 3.0$ as the local time scale. Integrating in the local time scale produced the steady state values $\tilde{p}_{10}$ and $\tilde{p}_{11}$, and then all the bond variables can be calculated from the constraint equations. They are used to compute the instantaneous equivalent linear resistances for R_1, R_2, R_3 and R_4. The referece [R] matrix, then is formed as below

$$[R] = \begin{bmatrix} -23.351 & 0.7223 \\ 0.7223 & -2.7223 \end{bmatrix}$$

For a smaller time scale, we set

$$m_{10} = 2.3351$$

$$m_{11} = 0.27223$$

Then the [A] matrix is

$$[A] = \begin{bmatrix} -10.0 & 2.653 \\ 0.3093 & -10.0 \end{bmatrix}$$

The eigenvalues of [A] are

$$s_1 = -9.0941$$

$$s_2 = -10.9659$$

Reset the local time scale by

$$\Delta t = 1/(3*10.9059) = 0.03056 \qquad => 0.030 \quad \text{sec}$$

$$t_f = 5/9.0941 = 0.5498 \qquad => 0.60 \quad \text{sec}$$

For comparison the eigenvalues and the local-time-scale have been computed when parameters m_{10} and m_{11} are arbitrarily set to be one. they are listed below:

$$s_1 = -2.697$$

$$s_2 = -23.376$$

and $\Delta t = 0.01425$ sec

$t_f = 1.854$ sec

Figure 4-10 displays the local integration process when the free parameters are selected arbitrarily, while Figure 4-11 shows the local integration at the same global time after the parameters have been optimized. From these plots we can see that the optimal parameters make the "diffusion" process quicker and the local integration steps fewer. The remaining process would be the same as that in the linear case discribed in preceding section. The only difference is that the nonlinear case needs to control the local time scale by detecting the change rate of the subsystem input vector or the equivalent resistances. Fortunately, the equivalent resistances, R_1, R_2, R_3 and R_4 in the particular example change only a little, so the local time scale needs not to be modified in the later 20 global time steps. The numerical output and a plot of this nonlinear case are shown in Figure 4-12 and Figure 4-13, respectively.

Comparing Figure 4-13 with Figure 4-9 obtained in linear case shows that they have a similar appearance aside from the different y-axis scales. It means that both cases have almost the same dynamic characteristics. This may be explained by the fact that the nonlinear dissipative field changes are very small in the particular system because of the constant input SF8.

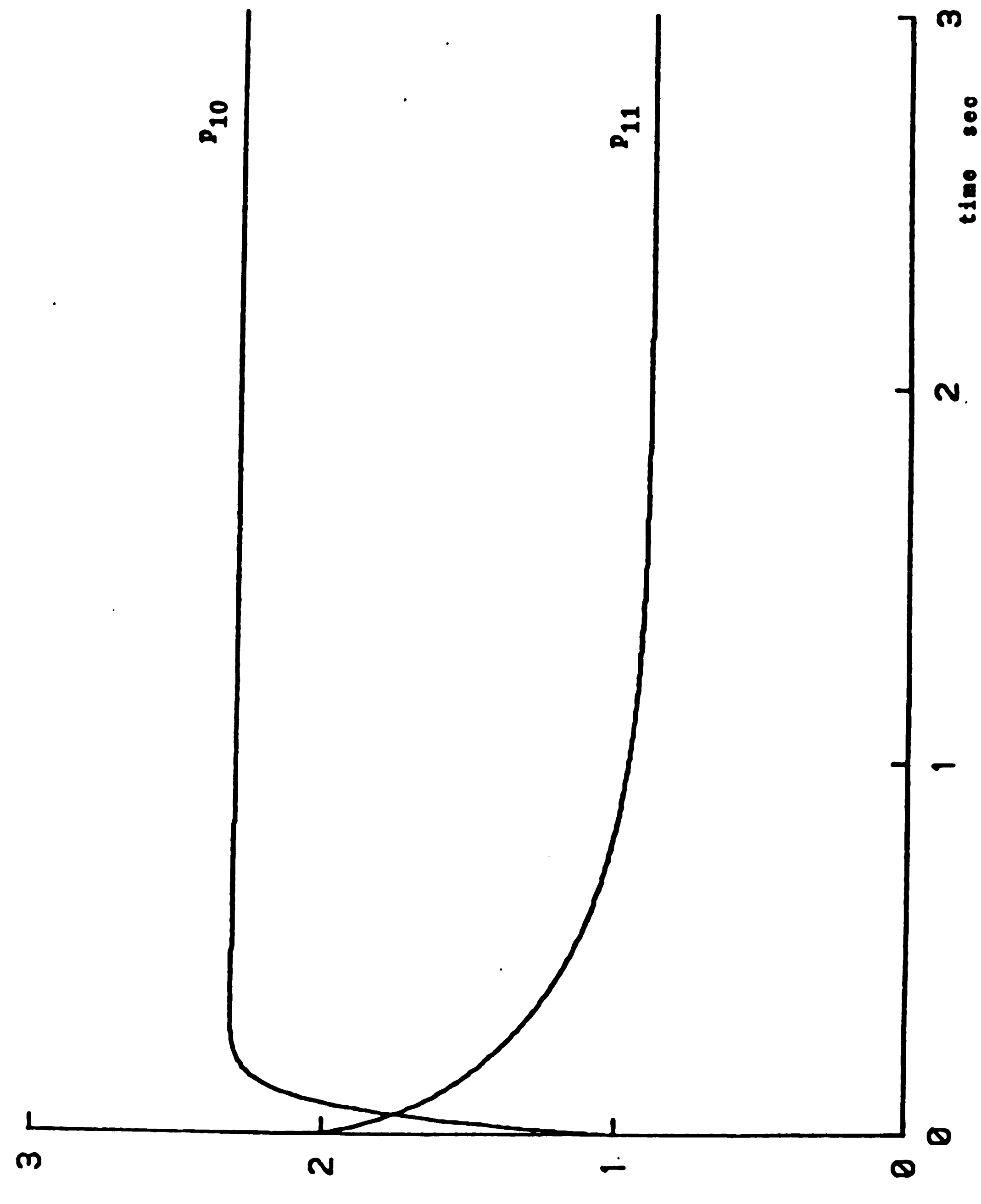


Figure 4-10 Local integration with arbitrary parameters

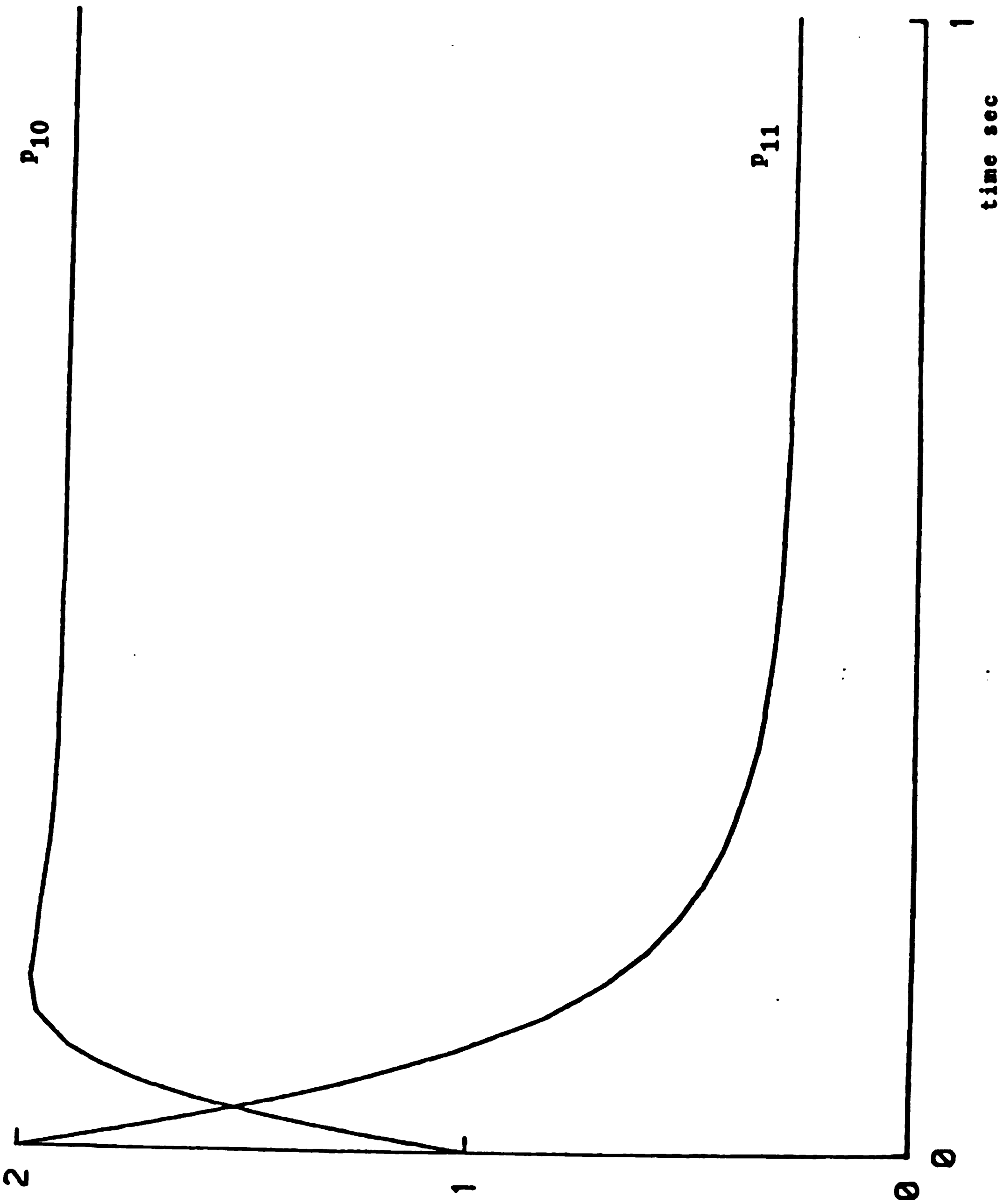


Figure 4-11 Local integration with optimal parameters

TIME	P10	P11	E9
0. 1000E+00	0. 8745E-01	0. 8121E-01	0. 1749E+01
0. 2000E+00	0. 8993E-01	0. 2213E+00	0. 1799E+01
0. 3000E+00	0. 8375E-01	0. 3266E+00	0. 1675E+01
0. 4000E+00	0. 7908E-01	0. 3988E+00	0. 1582E+01
0. 5000E+00	0. 7582E-01	0. 4479E+00	0. 1516E+01
0. 6000E+00	0. 7211E-01	0. 5038E+00	0. 1442E+01
0. 7000E+00	0. 7364E-01	0. 4812E+00	0. 1473E+01
0. 8000E+00	0. 7110E-01	0. 5191E+00	0. 1422E+01
0. 9000E+00	0. 7040E-01	0. 5295E+00	0. 1408E+01
0. 1000E+01	0. 6993E-01	0. 5366E+00	0. 1399E+01
0. 1100E+01	0. 6958E-01	0. 5414E+00	0. 1392E+01
0. 1200E+01	0. 6936E-01	0. 5446E+00	0. 1387E+01
0. 1300E+01	0. 6924E-01	0. 5468E+00	0. 1385E+01
0. 1400E+01	0. 6909E-01	0. 5482E+00	0. 1382E+01
0. 1500E+01	0. 6906E-01	0. 5492E+00	0. 1381E+01
0. 1600E+01	0. 6903E-01	0. 5499E+00	0. 1381E+01
0. 1700E+01	0. 6897E-01	0. 5504E+00	0. 1379E+01
0. 1800E+01	0. 6899E-01	0. 5507E+00	0. 1380E+01
0. 1900E+01	0. 6895E-01	0. 5509E+00	0. 1379E+01
0. 2000E+01	0. 6895E-01	0. 5509E+00	0. 1379E+01

Figure 4-12 Numerical result obtained by the two-time-scale integration

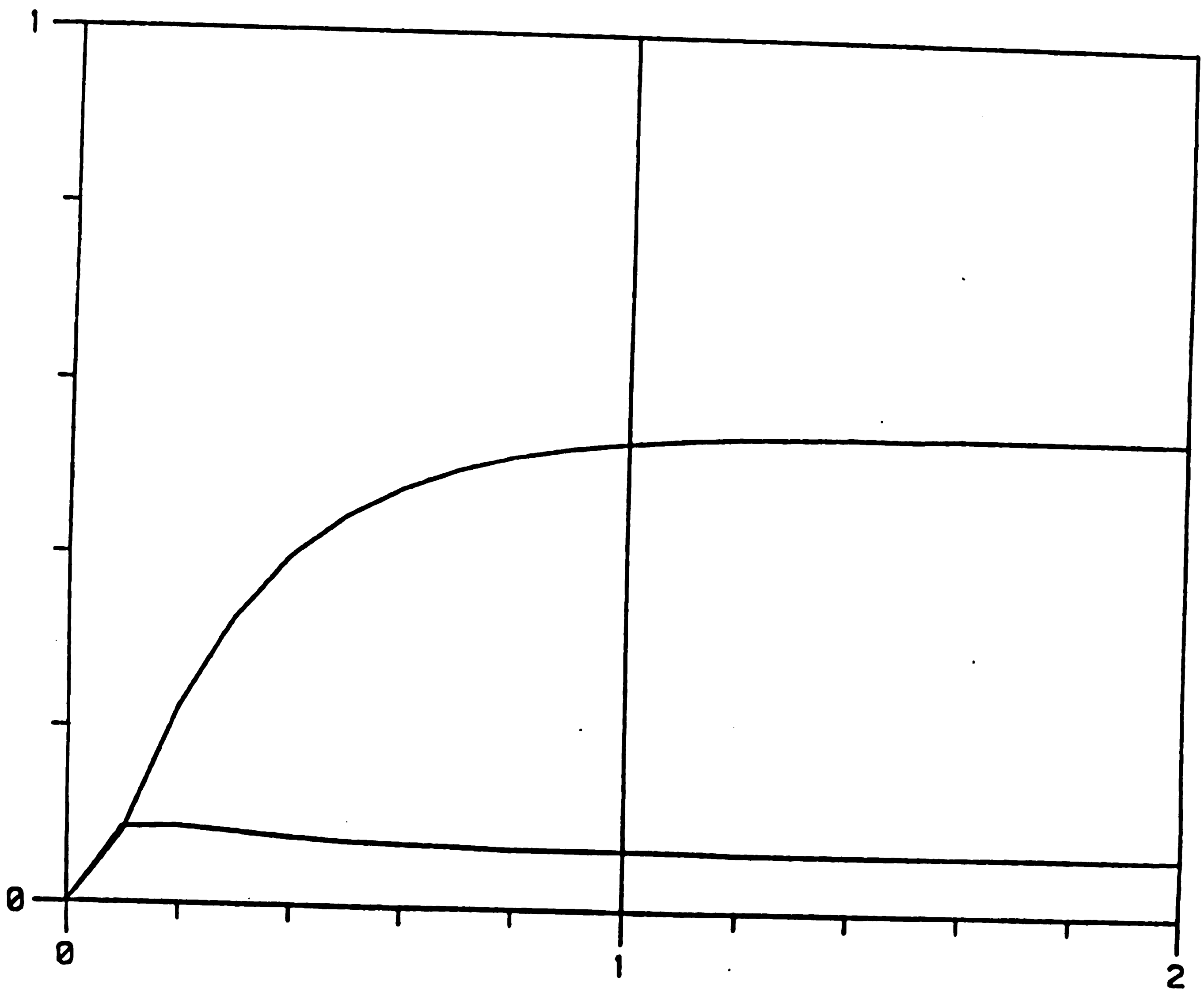


Figure 4-13 Dynamic response of the nonlinear case

5 CONCLUSION

5.1 Summary

The bond graph augmentation method for simulation of a dynamic system containing coupled nonlinear dissipative fields has been developed.

The causality assignment process is very useful for identifying R-fields. The order of augmented subsystems has been briefly discussed and it has been shown that to get the least augmentation order and to make the parameter selection as easy as possible, adding only one type of dynamic element (C or I) is desirable. An augmentation sequence has been suggested. By adding only one type of dynamic element the augmented subsystem will reach its steady state in a "diffusion" manner. To obtain the best computation efficiency the selection of parameters for the dynamicizers is the key. The discussion on parameter selection in linear subsystems has been extended to nonlinear subsystems through the concepts of instantaneous equivalent linear resistance and the reference [R] matrix. A general rule for parameter selection for arbitrary n-th-order subsystems has been suggested and numerically verified by several cases with different orders and numerical conditions.

The partition of dynamic fields and dissipative fields and the augmentation process are automatically accomplished by the subroutines

PART and AUGMET.

An example with linear and nonlinear cases has demonstrated the effectiveness of the new approach. The results obtained by this approach and by the existing ENPORT-5 program from linear case have been compared. It shows that the two-time-scale integration technique is valid for dynamic study. As the dissipative effects in the same system are reset to be nonlinear, the simulation result has shown the expected system dynamic behavior.

The approach to simulate a system with coupled nonlinear dissipative fields proposed above has several advantages. Among them are:

1) The parallelism in this approach is well-suited for parallel computers.

2) The manner in which we break up the whole problem into parts may be used to simulate large systems in a small capacity computer.

3) The coupled algebraic loop in system equations will be solved and the system dynamics can be investigated without necessarily approximating the nonlinear dissipative effect as a linear one.

On the other hand, the use of this approach based on only C or only I augmentation caries with it certain possible limitations:

1) For the adverse numerical condition of a dissipative field(its "spread" of eigenvalues is close to one), the computational efficiency would deteriorate.

2) The general rule for parameter selection may not apply to certain nonlinear static fields containing junction loops or GY elements because sometimes it is impossible to complete the causality by adding only C or I elements.

5.2 Future Development

The complete implementation of the approach is not possible until the program NONLIN, which can simulate uncoupled nonlinear systems, is available for use. This program is currently under development. The remaining work mainly is to design the control of the data flow among subsystems and the data assembly.

Basically the mixed C and I augmentation is not preferred becuse it might cause complex eigenvalues. But it still remains for futher investigation.

Some interesting problems are the sensitivity of the eigenstructure of an augmented subsystem to the change of its input vector and the influence of the augmentation order on the sensitivity. Investigation into these problems may benefit the control of the local time scales and therefore computational efficiency.

Until this stage, it is still too early to give a definite conclusion on the computational efficiency of this approach. The efficiency is commonly defined to be inversely proportional to the CPU time consummed during the execution of the computer program. One needs to compare the CPU time that is used to obtain the solution for different class problems by this approach with those used by other traditional methods, such as mentioned in chapter 1. It would be possible when the program NONLIN is available.

REFERENCES

REFERENCES

1). CSMP III , White Plains, New Yoyk, IBM, 1975.

2). R. N. Nilsen, CSSL-IV , Chatsworth, CA., Simulation Service, 1977.

3). G. A. Korn, J. V. Wait, Digital Continuous System Simulation , Englewood Cliff, New Jersey, Prentice-Hall, 1978.

4). J. C. Bowers and S. R. Sedore, SCEPTRE , Englewood Cliff, Prentice-Hall, 1971.

5). G. D. Byrne and C. A. Hall, Numerical Solutions of Systems of Nonlinear Algebraic Equations , Academic Press, New York, 1973.

6). C. Wolfe , Phillip, The Secant Method for Simultaneous Nonlinear Equations , Communications of the ACM, Vol. 2, 1959.

7). K. M. Brown, A Quadratically Convergent Newton-like Method Based Upon Gaussian Elimination , SIAM Journal on Numerical Analysis, Vol.6, 4, 1969.

8). The IMSL Library , International Mathematical & Statistical Libraries, Houston, Texas, 1982.

9). R. C. Rosenberg, D. C. Karnopp, Introduction to Physical System Dynamics , New York, McGraw-Hill, 1983.

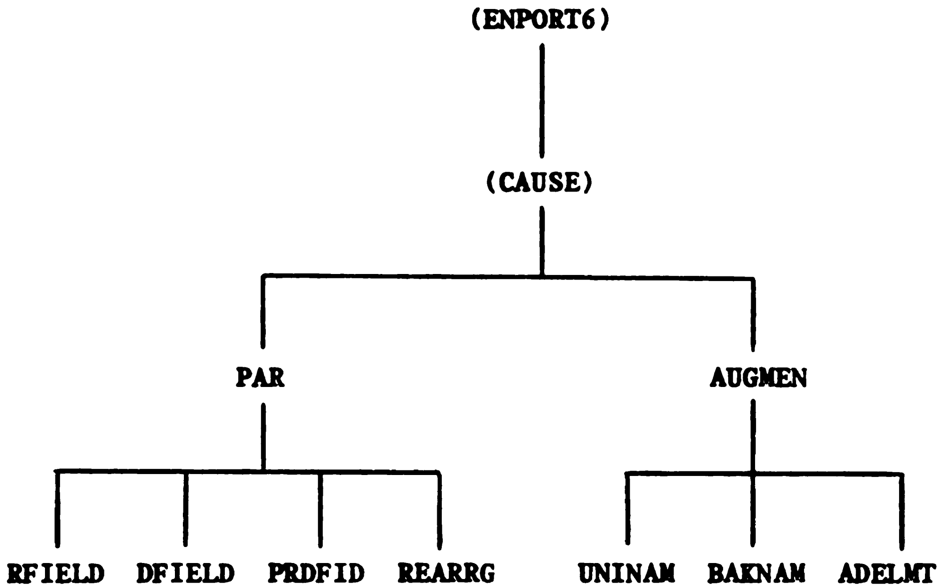
10). P. L. Graf, Dynamic Augmentation of Dissipation Algebraic Loops , M.S. Thesis, Michigan State University, 1981.

11). A. R. Gourlay and G. A. Watson, Computational Methods for Matrix Eigenproblems , New York, John Wiley & Sons, 1973.

12). Case Center User's Guide , A. H. Case Center for CAD/CAM, Michigan State University, 1982.

13). R. C. Rosenberg, ENPORT-5 User's Manual , Michigan State University, 1983.

APPENDIX A



CALLING TREE

APPENDIX B

```
C*PART*****************************************************************
C
      SUBROUTINE PART
C
C --- PROGRAMMER -- TONG ZHOU, AUG. 1983
C
C --- PART FINDS AND SEPARATES THE R-FIELDS AND THE JUNCTION
C     STRUCTURE COMPLEXES FROM THE ORIGINAL BOND-GRAPH.
C
C      MR   --- R-FIELD INDEX
C      MD   --- DYNAMIC FIELD INDEX.
C      NJ   --- CAUSAL BRANCH INDEX
C      JX   --- JUCTION STRUCTURE COMPLEX INDEX
C
C*** DECLARATIONS
C
$INSERT SYBGBK
$INSERT CAUSBK
$INSERT UTILBK
$INSERT PARTBK
C
C*********************************************************************
C
      MR=0
      MD=0
      NJ=0
      IBD1=0
      IBR1=0
      NELR=0
      NELD=0
      INN=0
C
      NBD2=NBD*2
      DO 2 I=1,NBD2
      ICMXT(I)=ICMX(I)
2     CONTINUE
C
C --- FIND THE FIRST JUNCTION
C
5     DO 10 N1=1,NEL
      IF(IELLST(N1).GE.6) GOTO 15
10    CONTINUE
C
15    NP1=NPTR(N1)
      NP2=NPTR(N1+1)-1
C
C --- DETERMINE IF ALL THE BONDS INCIDENT TO THE JUNCTION
C     ARE CAUSAL ASSIGNED
C
      DO 25 I=NP1,NP2
      IF(ICMXT(I).EQ.1) GOTO 30
25    CONTINUE
C
```

```
C --- GET THE ALL IMFORMATION FOR THE PRESENT DYNAMIC FIELD
C
      IF (MD.EQ.0) MD=1
      CALL DFIELD(MD,MR,N1,NP1,NP2)
      GOTO 35
C
C --- GET THE ALL INFORMATION FOR THE PRESENT R-FIELD
C
30    MR=MR+1
      CALL RFIELD (MD,MR,N1,NP1,NP2)
C
C
C --- EXTEND TO THE NEXT ADJUCENT FIELD
C
35    IF(IXBD.EQ.0) GOTO 40
      N1=JBD(IXBD)
      IXBD=IXBD-1
      GOTO 15
C
40    IF (IXBR.EQ.0) GOTO 90
      N1=JBR(IXBR)
      IXBR=IXBR-1
      GOTO 15
C
C --- FIND THE JUNCTION STRUCTURE COMPLEX FROM R-FIELDS
C
90    JX=0
      DO 110 I=1,MR
      L=LR(I)
      DO 100 J=1,L
      IF (IRLST(I,J).EQ.3) GOTO 110
100   CONTINUE
      JX=JX+1
      JCOMX(JX)=I
110   CONTINUE
      INO=INN
C
C----REARANGE THE DATA ARRAYS
C
      DO 120 I=1,MR
      MD1=0
      IBR1=IBR(I)
      NELR=LR(I)
      CALL REARRG ( I,MD1,IBR1,NELR,IBMXR,NBIMXR,IBMXRN,
     +               NBIXRN,INDXRE)
120   CONTINUE
C
      DO 130 I=1,MD
      MR1=0
      IBD1=IBD(I)
      NELD=LD(I)
      CALL REARRG ( MR1,I,IBD1,NELD,IBMXD,NBIMXD,IBMXDN,
     +               NBIXDN,INDXDE)
```

```
130   CONTINUE
C
C --- PRINT R-FIELDS AND JUNCTION COMPLEXES
C
       CALL PRRFLD(MR,MD)
C
      MRT=MR
      MDT=MD
      RETURN
      END
C
C
C
CRFIELD*************************************************************
C
      SUBROUTINE RFIELD (MD,MR,N1,NP1,NP2)
C
C --- RFIELD GETS ALL INFORMATION FOR THE R-FIELDS( zUNCTION
C    STRUCTURE COMPLEX IS TREATED AS A R-FIELD ) FROM THE ORIGINAL
C    BOND-GRAPH.
C
C      INPUT  --- N1,NP1,NP2
C                 OF THE BEGEINING JUNCTION OF THE R-FIELD
C      OUTPUT --- N1,NP1,NP2
C                 OF THE END JUNCTION OF THE R-FIELD
C
C      L      --- NODE INDEX
C      IB     --- BOND INDEX
C      IP     --- PORT INDEX
C      NUMJ   --- BRANCH INDEX
C      INDXRE --- NODE SERIEL NUMBER
C      IRLST  --- ELEMENT TYPE LIST
C      IRNAM  --- ELEMENT NAM LIST
C      NPTRR  --- POINTER LIST FOR NBIMXR(START OF BOND GROUP)
C      NBIMXR --- LIST OF BONDS INCIDENT TO EACH ELEMENT
C      JUNCTR --- LIST OF BRANCHES IN THE R-FIELD
C      LR     --- NODE NUMBER STORRAGE ARRAY FOR EACH R-FIELD
C      IBR    --- BOND NUMBER STORAGE ARRAY
C      IPR    --- PORT NUMBER STORAGE ARRAY
C
C*** DECLARATIONS
C
$INSERT SYBGBK
$INSERT CAUSBK
$INSERT UTILBK
$INSERT PARTBK
C
C
C*****************************************************************
C
C --- INITIALIZATION
C
      L=0
```

```
      IB=0
      IP=0
      NUMJ=0
      NN2=0
      NBD2=NBD*2
C
C --- GET JUNCTION DATA
C
10    L=L+1
      NPTRR(MR,L)=IB+1
      II=IB
      DO 20 I=NP1,NP2
      IF (ICMXT(I).EQ.0) RETURN
      IB=II+I-NP1+1
      NBIMXR(MR,IB)=NBIMX(I)
      ICMXR(MR,IB)=ICMX(I)
      IBOND=NBIMX(I)
      IBMXR(MR,IBOND,1)=IBMX(IBOND,1)
      IBMXR(MR,IBOND,2)=IBMX(IBOND,2)
20    CONTINUE
C
      INDXRE(MR,L)=N1
      IRLST(MR,L)=IELLST(N1)
      IRLNAM(MR,L)=IELNAM(N1)
      IF(L.EQ.1) GOTO 50
C
      DO 30 I=1,L-1
      IF(INDXRE(MR,I).EQ.INDXRE(MR,L)) GOTO 40
30    CONTINUE
      GOTO 50
40    IB=IB-NP2+NP1
      L=L-1
      GOTO 185
C
C --- GET ADJOINING NODE DATA
C
50    DO 180 I=NP1,NP2
      IBOND=NBIMX(I)
      NA=IBMX(IBOND,1)
      NB=IBMX(IBOND,2)
      N2=NA
      IF(N1.EQ.NA) N2=NB
C
      NUM=0
      DO 55 J=1,60
      IF(NBIMXR(MR,J).NE.IBOND) GOTO 55
      NUM=NUM+1
      IF(NUM.NE.2) GOTO 55
      ICMXT(I)=0
      GOTO 180
55    CONTINUE
C
C --- SAVE BOUNDARY INFORMATION
```

```
C
      IF (IELLST(N2).EQ.3) GOTO 95
      IF (IELLST(N2).EQ.4) GOTO 95
      IF (IELLST(N2).EQ.5) GOTO 95
      IF (ICMXT(I).EQ.1) GOTO 95
      NN2=NN2+1
      MD=MD+1
      DO 60 J=1,NBD2
      IF(J.EQ.I) GOTO 60
      IF(NBIMX(J).EQ.IBOND) GOTO 65
60     CONTINUE
65    IF(IELLST(N2).EQ.1.OR.IELLST(N2).EQ.2) ICMXT(J)=0
      IF(ICMXT(I).EQ.6) GOTO 70
      IF(ICMXT(I).EQ.3) GOTO 75
C
C---- SINGLE C- ELEMENT
C
70    ICMXT(I)=0
      ISOURL(NN2,1)=5
      ISOURL(NN2,2)=4
      ISOURN(NN2,1)='SF'
      ISOURN(NN2,2)='SE'
      ICMXD(MD,1)=6
      ICMXD(MD,2)=3
      IF(IELLST(N2).GT.5) GOTO 85
      GOTO 80
C
C--- SINGLE I- ELEMENT
C
75    ICMXT(I)=0
      ISOURL(NN2,1)=4
      ISOURL(NN2,2)=5
      ISOURN(NN2,1)='SE'
      ISOURN(NN2,2)='SF'
      ICMXD(MD,1)=3
      ICMXD(MD,2)=6
      IF(IELLST(N2).GT.5) GOTO 85
C
C --- GET THE DATA FOR THE SINGLE DYNAMIC SUBSYSTEM
C
80    NPTRD(MD,1)=1
      NPTRD(MD,2)=2
      NPTRD(MD,3)=3
      INDXDE(MD,1)=N1
      INDXDE(MD,2)=N2
      IDLST(MD,2)=IELLST(N2)
      IDNAM(MD,2)=IELNAM(N2)
      IDLST(MD,1)=ISOURL(NN2,2)
      IDNAM(MD,1)=ISOURN(NN2,2)
      NBIMXD(MD,1)=NBIMX(I)
      NBIMXD(MD,2)=NBIMX(I)
      IBMXD(MD,1,1)=N1
      IBMXD(MD,1,2)=N2
```

```
C
      IBD(MD)=2
      IBD1=2
      LD(MD)=2
      NELD=2
C
C --- CHECK IF THE JUNCTION HAS BEEN TAKEN
C
85    IF(IELLST(N2).EQ.1.OR.IELLST(N2).EQ.2) GOTO 90
      IF(ICMXT(J).EQ.0) MD=MD-1
      NUMJ=NUMJ+1
      JUNCTR(NUMJ)=N2
90    IP=IP+1
      IPORTR(MR,IP)=IBOND
      L=L+1
      IB=IB+1
      NPTRR(MR,L)=IB
      INDXRE(MR,L)=N2
      IRLST(MR,L)=ISOURL(NN2,1)
      IRLNAM(MR,L)=ISOURN(NN2,1)
      NBIMXR(MR,IB)=NBIMX(I)
      ICMXR(MR,IB)=ICMX(J)
      INN=INN+1
      INTER(INN,1)=IBOND
      INTER(INN,2)=IBOND
      INTER(INN,3)=MR
      INTER(INN,4)=MD
      INTER(INN,5)=ISOURL(NN2,1)
      GOTO 180
C
95    ICMXT(I)=0
      IF(IELLST(N2).GT.5) GOTO 110
      IF(IELLST(N2).EQ.1.OR.IELLST(N2).EQ.2) GOTO 180
C
      L=L+1
      IB=IB+1
      NPTRR(MR,L)=IB
      INDXRE(MR,L)=N2
      IRLST(MR,L)=IELLST(N2)
      IRLNAM(MR,L)=IELNAM(N2)
      NBIMXR(MR,IB)=NBIMX(I)
      DO 100 J=1,NBD2
      IF (J.EQ.I) GOTO 100
      IF (NBIMX(J).EQ.IBOND) GOTO 105
100   CONTINUE
105   ICMXR(MR,IB)=ICMX(J)
      ICMXT(J)=0
      IF(IELLST(N2).LT.6) GOTO 180
C
110    DO 115 K=1,L
      IF (INDXRE(MR,K).EQ.N2) GOTO 180
115    CONTINUE
      NUMJ=NUMJ+1
```

```
      JUNCTR(NUMJ)=N2
180    CONTINUE
      NT=N1
C
C --- NEXT JUNCTION
C
200    IF(NUMJ.EQ.0) GOTO 210
185    N1=JUNCTR(NUMJ)
      NUMJ=NUMJ-1
      NP1=NPTR(N1)
      NP2=NPTR(N1+1)-1
      DO 205 I=NP1,NP2
      IF(ICMXT(I).EQ.1) GOTO 10
      IF(ICMXT(I).EQ.0) GOTO 200
205    CONTINUE
      IXBD=IXBD+1
      JBD(IXBD)=N1
      GOTO 200
C
C --- STORE THE SIZE DATA
C
210   IPR(MR)=IP
      LR(MR)=L
      IBR(MR)=IB
      IBR1=IB
      NELR=L
      NPTRR(MR,L+1)=IB+1
C
C
      RETURN
      END
C
C
C
C*PRRFLD*********************************************************
C
      SUBROUTINE PRRFLD(MR,MD)
C
C --- PRRFLD PRINTS OUT THE INFIRMATION ABOUT THE R-FIELDS AND
C     JUNCTION STRUCTURE COMPLEXES.
C
C --- DECLARATION
C
$INSERT PARTBK
C
C***************************************************************
C
C --- PRINT R-FIELD
C
      OPEN(UNIT=5,FILE='SHOW')
C
      IR=MR-JX
      WRITE(*,1000) IR
```

```
1000  FORMAT(/'THERE ARE ',I2,'  R-FIELDS IN THE GIVEN SYSTEM')
      IF(IR.EQ.0) GOTO 300
      DO 100 I=1,MR
      IF(JX.NE.0) GOTO 950
      N=0
      GOTO 960
950   DO 900 N=1,JX
      IF(I.NE.JCOMX(N)) GOTO 900
      GOTO 100
900   CONTINUE
960   IN=I-N
      WRITE(*,1010) IN
1010  FORMAT(/'R-FIELD NUMBER ',I4,' :')
      WRITE(*,1020)
1020  FORMAT(/'NODES :')
      DO 150 K=1,LR(IN)
      I1=NPTRR(I,K)
      I2=NPTRR(I,K+1)-1
150   WRITE(*,5030) INDXRE(I,K),IRLNAM(I,K),(NBIMXR(I,J),J=I1,I2)
5030  FORMAT(7X,I3,2X,A4,2X,5I4)
      WRITE(*,1050)
1050  FORMAT(/,'PORTS :',/)
      WRITE(*,1060) (IPORTR(I,J),J=1,IPR(I))
1060  FORMAT(7X,I3)
100   CONTINUE
C
C --- PRINT JUNCTION STRUCTURE COMPLEXES
C
300   CONTINUE
      WRITE(*,1100)JX
1100  FORMAT(/'THERE ARE ',I2,'  JUNCTION STRUCTURE COMPLEXES')
      IF(JX.EQ.0) GOTO 260
      DO 200 I=1,JX
      JXN=JCOMX(I)
      WRITE(*,1210) I
1210  FORMAT(/'THE JUNCTION STRUCTURE COMPLEX NUMBER ',I2,' :')
      WRITE(*,1020)
      DO 250 K=1,LR(JXN)
      I1=NPTRR(I,K)
      I2=NPTRR(I,K+1)-1
250   WRITE(*,5030) INDXRE(I,K),IRLNAM(I,K),(NBIMXR(I,J),J=I1,I2)
      WRITE(*,1050)
      WRITE(*,1060)(IPORTR(JXN,J),J=1,IPR(I))
200   CONTINUE
C
C---- PRINT DYNAMIC FIELDS
C
260   WRITE(*,1300) MD
1300  FORMAT(/'THERE ARE ',I2,' DYNAMIC FIELDS IN THE GIVEN SYSTEM')
      DO 400 I=1,MD
      WRITE(*,1310) I
1310  FORMAT(/'DYNAMIC FIELD NUMBER ',I4,' :')
      WRITE(*,1020)
```

```
      DO 310 K=1,LD(I)
      I1=NPTRD(I,K)
      I2=NPTRD(I,K+1)-1
310   WRITE(*,5030) INDXDE(I,K),IDNAM(I,K),(NBIMXD(I,J),J=I1,I2)
      WRITE(*,1050)
      WRITE(*,1060)(IPORTD(I,J),J=1,IPD(I))
400   CONTINUE
C
      CLOSE(UNIT=5,STATUS='KEEP')
      RETURN
      END
C
C
C
C*DFIELD********************************************************
C
      SUBROUTINE DFIELD(MD,MR,N1,NP1,NP2)
C
C --- DFIELD GETS ALL INFORMATION FOR THE DYNAMIC ORSOURCE FIELDS
C      AJDASENT TO THE R-FIELDS FROM THE ORIGINAL SYSTEM.
C
C      INPUT   --- MD, MR ,N1,NP1,NP2 .
C      OUTPUT  --- MD,MR,N1,NP1,NP2 .
C                 OF THE END JUNCTION OF THE D-FIELD.
C
C         L      --- NODE INDEX
C         IB     --- BOND INDEX
C         IP     --- PORT INDEX
C         NUMJ   --- BRANCH INDEX
C         INDXDE --- NODE SERIEL NUMBER
C         IDLSTD  --- ELEMENT TYPE LIST
C         IDNAM  --- ELEMENT NAME LIST
C         NPTRD  --- POINTER LIST FOR NBIMXD(START OF BOND GROUP)
C         NBIMXD --- LIST OF BONDS INCIDENT TO EACH ELEMENT
C         JUNCTD  --- LIST OF BRANCHES IN THE D-FIELD
C         LD     --- NODE NUMBER STORAGE ARRAY FOR EACH D-FIELD
C         IBD   ---- BOND NUMBER STORAGE ARRAY
C         IPD    --- PORT NUMBER STORAGE ARRAY
C
C*** DECLARATIONS
C
$INSERT SYBGBK
$INSERT CAUSBK
$INSERT UTILBK
$INSERT PARTBK
C
C***************************************************************
C
C --- INITIALIZATION
C
      L=0
      IB=0
      IP=0
```

```
      NUMJ=0
C
C --- GET JUNCTION DATA
C
10    L=L+1
      NPTRD(MD,L)=IB+1
      II=IB
      DO 30 I=NP1,NP2
      IF (ICMXT(I).EQ.0) RETURN
      IB=II+I-NP1+1
      NBIMXD(MD,IB)=NBIMX(I)
      ICMXD(MD,IB)=ICMX(I)
      IBOND=NBIMX(I)
      IBMXD(MD,IBOND,1)=IBMX(IBOND,1)
      IBMXD(MD,IBOND,2)=IBMX(IBOND,2)
30    CONTINUE
C
      INDXDE(MD,L)=N1
      IDLST(MD,L)=IELLST(N1)
      IDNAM(MD,L)=IELNAM(N1)
      IF(L.EQ.1) GOTO 50
C
      DO 40 I=1,L-1
      IF(INDXDE(MD,I).EQ.INDXDE(MD,L)) GOTO 45
40    CONTINUE
      GOTO 50
45    IB=IB+NP1-NP2
      L=L+1
      GOTO 85
C
C --- GET ADJOINING NODE DATA
C
50    DO 80 I=NP1,NP2
      IBOND=NBIMX(I)
      NA= IBMX(IBOND,1)
      NB= IBMX(IBOND,2)
      N2=NA
      IF(N1.EQ.NA) N2=NB
C
C--- FIND THE NODE WHICH HAS BEEN INCLUDED.
C
      NUM=0
      DO 55 J=1,20
      IF(NBIMXD(MD,J).NE.IBOND) GOTO 55
      NUM=NUM+1
      IF(NUM.NE.2) GOTO 55
      ICMXT(I)=0
       GOTO 80
55    CONTINUE
C
      DO 56 J=1,NBD2
      IF (J.EQ.I) GOTO 56
      IF(NBIMX(J).EQ.IBOND) GOTO 57
```

```
56     CONTINUE
C
57     IF(ICMXT(J).EQ.0) GOTO 65
       IF(IELLST(N2).GT.5) GOTO 59
       ICMXT(I)=0
       ICMXT(J)=0
       L=L+1
       IB=IB+1
       NPTRD(MD,L)=IB
       INDXDE(MD,L)=N2
       NBIMXD(MD,IB)=NBIMX(I)
       ICMXD(MD,IB)=ICMX(J)
       IDLST(MD,L)= IELLST(N2)
       IDNAM(MD,L)=IELNAM(N2)
       GOTO 80
C
59     DO 75 K=1,L
       IF(INDXDE(MD,K).EQ.N2) GOTO 80
75     CONTINUE
       ICMXT(I)=0
       NUMJ=NUMJ+1
       JUNCED(NUMJ)=N2
       JUNCB2(NUMJ)=I
       JUNCB(NUMJ)=IBOND
       JUNCOL(NUMJ)=N1
       GOTO 80
C
65     ICMXT(I)=0
       L=L+1
       IB=IB+1
       NPTRD(MD,L)=IB
       INDXDE(MD,L)=N2
       NBIMXD(MD,IB)=NBIMX(I)
       IBOND=NBIMX(I)
       DO 651 J=1,NBD2
       IF (J.EQ.I) GOTO 651
       IF (NBIMX(J).EQ.IBOND) GOTO 652
651    CONTINUE
652    ICMXD(MD,IB)=ICMX(J)
       IP=IP+1
       IPORTD(MD,IP)=IBOND
       IF(IELLST(N2).EQ.6) GOTO 67
       IF (IELLST(N2).EQ.7) GOTO 68
       GOTO 80
67     IDLST(MD,L)=4
       IDNAM(MD,L)='SE'
       GOTO 80
68     IDLST(MD,L)=5
       IDNAM(MD,L)='SF'
80     CONTINUE
C
C --- NEXT JUNCTION
C
```

```
      NT=N1
85    IF(NUMJ.EQ.0) GOTO 120
      N1=JUNCED(NUMJ)
      NUMJ=NUMJ-1
      NP1=NPTR(N1)
      NP2=NPTR(N1+1)-1
      DO 90 I=NP1,NP2
      IF(ICMXT(I).EQ.1) GOTO 100
90    CONTINUE
      GOTO 10
C
100   IXBR=IXBR+1
      JBR(IXBR)=N1
      N2=N1
      L=L+1
      IB=IB+1
      NPTRD(MD,L)=IB
      INDXDE(MD,L)=N2
      NBIMXD(MD,IB)=NBIMX(JUNCB2(NUMJ+1))
      IBOND=JUNCB(NUMJ+1)
      DO 653 J=1,NBD2
      IF (J.EQ.JUNCB2(NUMJ+1)) GOTO 653
      IF (NBIMX(J).EQ.IBOND) GOTO 654
653   CONTINUE
654   ICMXD(MD,IB)=ICMX(J)
      IP=IP+1
      IPORTD(MD,IP)=NBIMX(JUNCB2(NUMJ+1))
      IF(IELLST(N2).EQ.6) GOTO 110
      IF(IELLST(N2).EQ.7) GOTO 115
      GOTO 120
110   IDLST(MD,L)=4
      IDNAM(MD,L)='SE'
      GOTO 120
115   IDLST(MD,L)=5
      IDNAM(MD,L)='SF'
120   IF(NUMJ.NE.0) GOTO 85
      IPD(MD)=IP
      LD(MD)=L
      IBD(MD)=IB
      NELD=L
      IBD1=IB
      NPTRD(MD,L+1)=IB+1
      N1=NT
C
C
      RETURN
      END
C
C
C
C*REARRG**************************************************************
C
      SUBROUTINE REARRG(MRR,MDD,IB,NELS,IBMXS,NBIMXS,IBMXSN,
```

```
     +       NBIXSN,INDXES)
C ---- REARRG REARRANGE ALL THE ARRAY IN SEQUENCE.
C
C*** DECLARATIONS
C
      INTEGER IA(60),IBB(60),NBIMXS(5,60),IBM(5,30,2),
     +     IBMXS(5,30,2),NBIXSN(5,60),IBMXSN(5,30,2),INDXES(5,30)
C
$INSERT PARTBK
$INSERT SYBGBK
C
C
C*********************************************************************
C
      IF(MRR.EQ.0.AND.MDD.EQ.0) RETURN
      IF(MRR.EQ.0) GOTO 2
      MS=MRR
      KK=1
      GOTO 5
2     MS=MDD
      KK=2
C
5     DO 10 I=1 50
      IA(I)=0
      IBB(I)=0
10    CONTINUE
C
C
      NBDS2=IB
      NBDS=NBDS2/2
      DO 100 I=1,NBDS2
      IA(I)=NBIMXS(MS,I)
100   CONTINUE
C
      DO 120 I=1,NBDS2-1
      L=I+1
      DO 120 J=L,NBDS2
      IF(IA(J)-IA(I)) 110,120,120
110   ITEMP=IA(I)
      IA(I)=IA(J)
      IA(J)=ITEMP
120   CONTINUE
C
      DO 130 I=1,NBDS2
      X=I/2.0
      K=I/2
      Y=X-K
      IF(Y.EQ.0) IBB(K)=IA(I)
130   CONTINUE
C
      DO 160 I=1,NBDS2
      DO 150 J=1,NBDS
      IF(NBIMXS(MS,I).NE.IBB(J)) GOTO 150
```

```
      NBIXSN(MS,I)=J
      GOTO 160
150   CONTINUE
160   CONTINUE
C
      DO 170 I=1,NBD
      IBM(MS,I,1)=IBMXS(MS,I,1)
      IBM(MS,I,2)=IBMXS(MS,I,2)
170   CONTINUE
C
      DO 190 I=1,NBD
      DO 180 J=1,NBDS
      IF(IBM(MS,I,1).EQ.0) GOTO 190
      IF(I.NE.IBB(J)) GOTO 180
      IBMXSN(MS,J,1)=IBM(MS,I,1)
      IBMXSN(MS,J,2)=IBM(MS,I,2)
180   CONTINUE
190   CONTINUE
C
      DO 191 J=1,NBDS
      DO 192 K=1,2
      DO 193 I=1,NELS
      IF (IBMXSN(MS,J,K).EQ.INDXES(MS,I)) GOTO 194
193   CONTINUE
194   IBMXSN(MS,J,K)=I
192   CONTINUE
191   CONTINUE
C
      DO 220 N=1,5
      IBOND=INTER(N,KK)
      IF (IBOND.EQ.0) GOTO 220
      K1=4
      IF (KK.EQ.1) K1=3
      IF (INTER(N,K1).NE.MS) GOTO 220
      DO 200 M=1,NBDS
      IF(IBB(M).EQ.IBOND) GOTO 210
200   CONTINUE
210   INTER(N,KK)=M
220   CONTINUE
C
C
      RETURN
      END
```

```
C
C-AUGMEN-***************************************************************
C
      SUBROUTINE AUGMEN(IR)
C
C --- PROGRAMMER -- TONG ZHOU, AUG. 1983
C
C --- AUGMEN CONVERTS A STATIC SUBSYSTEM TO A DYNAMIC SUBSYSTEM BY
C     INTRODUCING C OR I ELEMENTS FOLLOWING A GENERAL RULE.
C
C***DECLARATIONS
C
      DIMENSION NR(35),JZERO(10),JONE(10),IN10(35)
C
$INSERT SYBGBK
$INSERT PARTBK
$INSERT CAUSBK
$INSERT UTILBK
C
C***************************************************************
C
C---- RESTORE THE SUBSYSTEM DATA FOR PROCESSING.
C
      NEL=LR(IR)
      NBD2=IBR(IR)
      NBD=NBD2/2
C
      CALL UNINAM(IR,IRLST,IRLNAM,NPTRR,IBMXRN,ICMXR,NBIXRN)
C
C---- ADD DYNAMIC ELEMENTS TO ACTING JUNCTIONS.
C
      WRITE(*,5)
      DO 10 J=1,INN
      IF(INTER(J,3).NE.IR) GOTO 10
      N1=IBMX(INTER(J,1),1)
      N2=IBMX(INTER(J,1),2)
       IF (IELLST(N2).EQ.6.OR.IELLST(N2).EQ.7) JNOD=N2
      IF(IELLST(N1).EQ.6.OR.IELLST(N1).EQ.7) JNOD=N1
      IF (JNOLD.EQ.JNOD) GOTO 10
      GOTO 15
C
5     FORMAT(/,'ADD C/I ELEMENT TO THE ACTING JUNCTIONS')
15    CALL ADELMT(JNOD,IR)
      JNOLD=JNOD
C
10    CONTINUE
C
C
C---- TEST FOR COMPLETED CAUSALITY ON THE GRAPH.
C
400   DO 20 J=1,NBD2
      IF (ICMX(J).EQ.1) GOTO 30
20    CONTINUE
```

```
      GOTO 800
C
C --- ADD C- OR I- ELEMENTS TO THE MULTI-R TYPE JUNCTIONS.
C
30    WRITE(*,40)
40    FORMAT(/,'ADD C/I ELEMENT TO THE MULTI-R TYPE JUNCTIONS')
200   DO 50 I=1,NEL
50    NR(I)=0
C
      DO 60 I=1,NEL
      NP1=NPTR(I)
      NP2=NPTR(I+1)-1
      IF (NP1.EQ.NP2) GOTO 60
      DO 55 K=NP1,NP2
      IBOND=NBIMX(K)
      IF (ICMX(K).NE.1) GOTO 55
      NA=IBMX(IBOND,1)
      NB=IBMX(IBOND,2)
      N2=NA
      IF (I.EQ.NA) N2=NB
      IF (IELLST(N2).EQ.3) NR(I)=NR(I)+1
55    CONTINUE
60    CONTINUE
C
      JOLD=1
      DO 70 J=1,NEL
      IF (NR(J).LE.NR(JOLD)) GOTO 70
      JOLD=J
70    CONTINUE
C
      IF(NR(JOLD).EQ.1) GOTO 85
      IF (NR(JOLD).EQ.0) GOTO 800
      JNOD=JOLD
      CALL ADELMT(JNOD,IR)
      GOTO 200
C
C
C --- ADD C- OR I- ELEMENTS TO THE 0- OR 1- JUNCTIONS WHICH
C       POSESSES THE MOST INTERNAL BONDS.
C
85    WRITE(*,90)
90    FORMAT(/,'ADD C/I ELEMENT TO THE MULTI-BRANCH JUNCTIONS')
95    DO 100 I=1,NEL
      IN10(I)=0
100   CONTINUE
C
      DO 120 N1=1,NEL
      NP1=NPTR(N1)
      NP2=NPTR(N1+1)-1
      IF(NP1.EQ.NP2) GOTO 120
      DO 105 K=NP1,NP2
      IF (ICMX(K).EQ.1) GOTO 110
105   CONTINUE
```

```
      GOTO 120
110   DO 115 K=NP1,NP2
      IBOND=NBIMX(K)
      NA=IBMX(IBOND,1)
      NB=IBMX(IBOND,2)
      N2=NA
      IF(N1.EQ.NA) N2=NB
      IF(IELLST(N2).EQ.6.OR.IELLST(N2).EQ.7) IN10(N1)=IN10(N1)+1
115   CONTINUE
120   CONTINUE
C
      JOLD=0
      DO 130 I=1,NEL
      IF (IN10(I).LE.2) GOTO 130
      IF (IN10(I).LE.IN10(JOLD)) GOTO 130
      JOLD=I
130   CONTINUE
C
      IF (JOLD.EQ.0) GOTO 300
      JNOD=JOLD
      CALL ADELMT(JNOD,IR)
      GOTO 95
C
C
C --- ADD C- OR I- ELEMENTS TO THE SINGLE-R TYPE JUNCTIONS.
C
300   J0=0
      J1=0
      WRITE(*,301)
301   FORMAT(/,'ADD C/I ELEMENT TO THE SINGLE-R TYPE JUNCTINS')
      DO 310 N=1,NEL
      IF (IELLST(N).EQ.6.OR.IELLST(N).EQ.7) GOTO 320
      GOTO 310
320   NP1=NPTR(N)
      NP2=NPTR(N+1)-1
      DO 330 K=NP1,NP2
      IF (ICMX(K).EQ.1) GOTO 340
330   CONTINUE
      GOTO 310
340   IF(IELLST(N).EQ.7) GOTO 345
      J0=J0+1
      JZERO(J0)=N
      GOTO 310
345   J1=J1+1
      JONE(J1)=N
310   CONTINUE
C
      J10=J1
      IF(J1.GT.J0) J10=J0
C
      DO 350 J=1,J10
      JONE(J)=JONE(J)
      IF (J1.GT.J0) JONE(J)=JZERO(J)
```

```
      NP1=NPTR(JONE(J))
      NP2=NPTR(JONE(J)+1)-1
      DO 355 K=NP1,NP2
      IF (ICMX(K).EQ.1) GOTO 360
355   CONTINUE
      GOTO 350
360   JNOD=JONE(J)
      CALL ADELMT(JNOD,IR)
350   CONTINUE
      GOTO 800
C
C
800   DO 850 J=1,NBD2
      IF (ICMX(J).EQ.1) GOTO 900
850   CONTINUE
C
C
      CALL BAKNAM(IR,IRLST,IRLNAM,NPTRR,IBMXRN,ICMXR,NBIXRN)
C
C
900   RETURN
      END
C
C
C-UNINAM-*********************************************************
C
      SUBROUTINE UNINAM(IRD,ILST,INAM,NPT,IBM,ICM,NNBIM)
C
C --- UNINAM STORES THE DATA OF A SUBSYSTEM INTE THE MATRICES
C     WHICH ARE COMPATIBLE WITH THE PROGRAM ENPORT5(GLOBAL FORM).
C
      DIMENSION ILST(5,30),NPT(5,31),IBM(5,30,2),
     +          ICM(5,60),NNBIM(5,60)
      CHARACTER*32 INAM(5,30)
$INSERT SYBGBK
$INSERT PARTBK
$INSERT CAUSBK
$INSERT UTILBK
C
C*****************************************************************
C
C
C
      DO 10 I=1,NEL
      IELLST(I)=ILST(IRD,I)
      IELNAM(I)=INAM(IRD,I)
10    CONTINUE
      DO 15 I=1,NEL+1
      NPTR(I)=NPT(IRD,I)
15    CONTINUE
C
      DO 20 I=1,NBD
      IBMX(I,1)=IBM(IRD,I,1)
```

```
      IBMX(I,2)=IBM(IRD,I,2)
20    CONTINUE
C
      DO 25 I=1,NBD*2
      ICMX(I)=ICM(IRD,I)
      NBIMX(I)=NNBIM(IRD,I)
25    CONTINUE
C
      RETURN
      END
C
C
C
C-BAKNAM-*************************************************************
C
      SUBROUTINE BAKNAM (IRD,ILST,INAM,NPT,IBM,ICM,NBIM)
C
C --- BAKNAM RESTORES THE DATA OF THE SYBSYTEM BACK TO ITS LOCAL
C     MATICES.
C
      DIMENSION ILST(5,30),NPT(5,31),IBM(5,30,2),ICM(5,60),
     +          NBIM(5,30)
      CHARACTER*32 INAM(5,30)
$INSERT SYBGBK
$INSERT PARTBK
$INSERT CAUSBK
$INSERT UTILBK
C
C********************************************************************
C
C
C
      NBD2=NBD*2
      DO 10 I=1,NEL
      ILST(IRD,I)=IELLST(I)
10    INAM(IRD,I)=IELNAM(I)
C
      DO 15 I=1,NEL+1
15    NPT(IRD,I)=NPTR(I)
C
      DO 20 I=1,NBD
      IBM(IRD,I,1)=IBMX(I,1)
20    IBM(IRD,I,2)=IBMX(I,2)
C
      DO 25 I=1,NBD2
      ICM(IRD,I)=ICMX(I)
25    NBIM(IRD,I)=NBIMX(I)
C
      RETURN
      END
C
C
C-ADELMT-*************************************************************
```

```
C
      SUBROUTINE ADELMT(JNOD,IR)
C
C --- ADELMT ADDS THE I OR C ELEMENT TO AN APPROPRIAT JUNCTION.
C
$INSERT SYBGBK
$$NSERT PARTBK
$INSERT CAUSBK
C
C**********************************************************************
C
C
      IF(IELLST(JNOD).EQ.6) GOTO 110
      NENAME='I'
      IBTD=1
      GOTO 120
110   NENAME='C'
      IBTD=-1
120   NEL=NEL+1
      NBD=NBD+1
      WRITE(*,1000) NENAME,JNOD,INDXRE(IR,JNOD)
1000  FORMAT(/,'   ADD  ',A2,'  ELEMENT TO NEW NODE ',I2,
     +            '  (THE OLD NODE ',I2,'  )')
      IF(IBTD.EQ.-1) IELLST(NEL)=1
      IF(IBTD.EQ.1) IELLST(NEL)=2
      IELNAM(NEL)=NENAME
C
      IBMX(NBD,1)=JNOD
      IBMX(NBD,2)=NEL
      N1=NPTR(JNOD)
      N2=NPTR(NEL)-1
      J=N2
      DO 50 N=N1,N2
      NBIMX(J+1)=NBIMX(J)
      ICMX(J+1)=ICMX(J)
      J=J-1
50    CONTINUE
C
      NBIMX(N1)=NBD
      NBIMX(N2+2)=NBD
      NBIT=6
      IF(IBTD.EQ.1) NBIT=3
      ICMX(N1)=9-NBIT
      ICMX(N2+2)=NBIT
      N1=JNOD+1
      DO 60 N=N1,NEL
      NPTR(N)=NPTR(N)+1
60    CONTINUE
      NPTR(NEL+1)=NPTR(NEL)+1
C
C
      MNSTKN=40
      MNSTKM=40
```

```
      LEVEL=2
       IF(IELLST(NEL).EQ.1) THEN
         IT=6
       ELSEIF (IELLST(NEL).EQ.2) THEN
        IT=3
       ELSE
         GOTO 280
       ENDIF
      NP1=NPTR(NEL)
      NP2=NPTR(NEL+1)-1
C
      DO 270 K=NP1,NP2
      ICMX(K)=1
      CALL TWPASS(NEL,K,IT)
270   CONTINUE
280   CONTINUE
C
      RETURN
      END
C
```