AD-A083 577 BOSTON UNIV MA COLL OF ENGINEERING F/G 17/2
A QUANTITATIVE APPROACH TO AGGREGATION IN THE MODELING OF TACTI--ETC(U)
MAR 80 H D'ANGELO AFOSR-79-0101

UNCLASSIFIED AFOSR-TR-80-0258 NL

1 OF 1
AD A
083577

END
DATE
FILMED
5-80
DTIC

AFOSR-TR- 80-0258

ADA083577

March 8, 1980

LEVEL II

FINAL REPORT

Submitted to

Air Force Office of Scientific Research
Building 410
Bolling AFB, D.C. 20332

AFOSR-79-0101

A QUANTITATIVE APPROACH TO AGGREGATION IN THE MODELING OF TACTICAL COMMAND, CONTROL, AND COMMUNICATION SYSTEMS

by

Henry D'Angelo
College Of Engineering
110 Cummington Street
Boston, Massachusetts 02215

411 715

DTIC
ELECTE
APR 23 1980
S
C

Report No. 1

80 4 21 177

...lic release;
...ited.

ABSTRACT

This research effort is directed at determining the feasibility of using structural-sensitivity measures as the basis for simplifying large mathematical models of dynamic systems (e.g., models of tactical command, control, and communication systems). Toward this end, a FORTRAN program was written which can be used to simplify models characterized by sets of linear differential equations. Specifically, the program determines the optimal simplified model (i.e., the set of coefficients characterizing the set of linear differential equations) of specified dimension corresponding to the original linear model of higher dimension.

The algorithm, on which the FORTRAN program is based, minimizes an objective function which is defined in terms of the structural sensitivities of the state variables to be preserved in the simplified model. The program inputs are the set of coefficients which define (1) the linear differential equations representing the original model, (2) the dimension of the simplified model to be determined, and (3) the set of parameters defining the objective function to be minimized. The program outputs are the set of coefficients which define the optimal simplified model and the value of the objective function corresponding to the optimal simplified model.

INTRODUCTION

An approach is proposed to simplifying complex mathematical models of dynamic systems. The approach is based on the concept of structural sensitivity.

The mathematical model to be simplified is represented as a graph in which each link of the graph represents the dependency of one system variable on another. The state variables of the system are selected so as to include all the system variables of interest which are also to be included in the simplified model. The remaining state variables are selected so as to minimize the sensitivity of the interesting state variables to the cutting of all the links between the interesting state variables and the remaining state variables. When this minimization is realized, the optimal simplified model, which includes all the interesting state variables, can be "cut" out of the original system model. This process involves finding that transformation of the original state variables which preserves the interesting state variables while minimizing the structural sensitivity of these state variables with respect to the remaining state variables.

In situations where the best possible simplified model that can be cut out of the original system model is not an acceptable representation of the original system, the dimension of the simplified model can be increased to improve its accuracy. In this case, a transformation must be found which not only preserves the interesting state variables, but which also partitions the remaining state variables into two sets. One set of state variables from this partition is added to the set of interesting state variables to produce an augmented set of interesting state variables; the simplified model will now include this agumented set and thus be of higher dimension, and of improved accuracy. The other set of state variables from the partition is selected so as to minimize the structural sensitivity of the set of interesting state variables with respect to this remaining set.

A

MODEL SIMPLIFICATION: THE GENERAL AGGREGATION PROBLEM

The general aggregation problem is stated here for the case that the system of interest is well modeled by a set of ordinary differential equations. However, the extention of these results to those important systems which are best modeled by discrete-event models is not trivial, particularly with respect to the computation of sensitivities within dynamic systems. Additional research is called for here.

Consider that the system to be studied is well modeled by a set of ordinary differential equations given in canonical state-variable form:

$$\frac{dx}{dt} = f_x(x,u),\ x(t_o) = x_o,\ t \geq t_o \qquad (1)$$

where

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \quad u = \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix}, \quad f_x = \begin{bmatrix} f_{x1} \\ \vdots \\ f_{xn} \end{bmatrix}$$

For the case that n is very large, one is seldom interested in observing all the state variables. In such cases, one observes only a small subset of the state variables: i.e.,

$$q = g(x) \qquad (2)$$

where

$$q = \begin{bmatrix} q_1 \\ \vdots \\ q_p \end{bmatrix}, \quad g = \begin{bmatrix} g_1 \\ \vdots \\ g_p \end{bmatrix}$$

and $p < n$. The function g is called the <u>aggregation function</u>. The problem of aggregation is that of trying to find a simpler model to generate the observed variables q than that provided by equations (1) and (2). Ideally, one seeks an aggregated model characterized by function f_q such that

$$\frac{dq}{dt} = f_q(q,u),\ q(t_o) = g(x_o),\ t \geq t_o \qquad (3)$$

where

$$fq = \begin{bmatrix} f_{q1} \\ \vdots \\ f_{qp} \end{bmatrix}$$

In general there exists no f_q such that an aggregated model can generate the observed variables of a disaggregated model (If the state x is observable through output q, then q cannot, in general, be generated by a system of dimension less than n). In special cases where an f_q can be found such that equation (3) is valid, the aggregated model is said to be <u>dynamically exact</u> to the disaggregated model with respect to q. However, dynamic exactness is so rare in practical situations that, practically, the problem of aggregation is that of finding a function f_q that can be used to generate an approximation q_a to q:

$$\frac{dq_a}{dt} = f_q(q_a,u),\ q_a(to) = g(x_o),\ t \geq t_o \qquad (4)$$

where

$$q_a = \begin{bmatrix} q_{a1} \\ \vdots \\ q_{ap} \end{bmatrix}$$

Often, the variables of interest are so few in number compared to the dimension of the disaggregated model (e.g., one may be interested in only the average of all the state variables in a complex system having, say, 50,000 state variables) that there is little hope of finding any f_q to generate a reasonable approximation to q. In such cases, it is necessary to increase the dimension of the aggregated model. Toward this end the aggregation function is redefined:

$$q = \begin{bmatrix} q_c \\ q_v \end{bmatrix} = \begin{bmatrix} g_c(x) \\ g_v(x) \end{bmatrix}$$

where

$$q_c = \begin{bmatrix} q_{c1} \\ \vdots \\ q_{cr} \end{bmatrix} = \begin{bmatrix} g_{c1}(x) \\ \vdots \\ g_{cr}(x) \end{bmatrix} , \quad q_v = \begin{bmatrix} q_{v,r+1} \\ \vdots \\ q_{vp} \end{bmatrix} = \begin{bmatrix} g_{v,r+1}(x) \\ \vdots \\ g_{vp}(x) \end{bmatrix}$$

where q_c represents the variables of interest and q_v represents the additional variables to be included in the aggregated model to increase model dimension for purposes of improving the approximation. Thus, function g_c is a fixed function defining the variables of interest and function g_v is a function to be selected in the most advantageous manner in designing the aggregated model.

In trading dynamic exactness for model simplicity, by accepting an approximation to q, a difficult problem arises. Namely, one must have a basis for comparing alternate approximations. Clearly, the effectiveness of an approximation is closely tied to the use that the aggregated model is to be put to. Thus, the criteria that might be used in evaluating aggregated models to be used for estimation and prediction could be significantly different from the criteria used when the models are to be used for determining controls. The development of pertinent criteria for evaluating aggregated models is an essential part of the aggregation problem.

A QUANTITATIVE APPROACH TO AGGREGATION: STRUCTURAL SENSITIVITY

The approach taken to the aggregation problem is based on system structure. Specifically, system models are represented by graphs such as link-node structures (1), system diagrams (2), or signal-glow graphs (3, 4) in which certain points on the graphs represent system variables and the influence of one variable on another is denoted by the existence of a path from that variable to the other. Fundamental to this approach is the premise that a proposed aggregated model can be imbedded within a larger system defined by the disaggregated system (i.e., the funciton f_x) and the aggregation function (i.e., the functions g_c and g_v). Importantly, in order that the proposed aggregated model exactly generate the variables of interest q_c, it is necessary that additional variables, say x_Δ, which are functions of the state variables of the disaggregated model, be provided as special inputs to the aggregated model. These relations between the larger-system variables, x_Δ, and the proposed aggregated system variables, g_c and q_v, represent connections in the system graph. Perfect aggregation is achieved when the q_c generated by the aggregated model is totally insensitive to the existence of these connections.

With such insensitivity, all connections from the larger system can be literally cut and the aggregated model can be removed from the larger system. This sensitivity of a system's variables to the cutting of connecting links is called structural sensitivity. By introducing a gain parameter in such connecting links it is possible to relate structural sensitivities to the well-defined parameter sensitivities (e.g., a link gain equal to 1 implies the connection exists, and a link gain equal to 0 implies the connection is broken). The following example illustrates the proposed approach to aggregation.

Consider a continuous autonomous system that is well modeled by

$$\frac{dx}{dt} = f_x(x)$$

(x is an n vector). We would like to design an aggregated model of this system and we demand that the aggregated model generate a specified set of outputs q_c defined by a fixed aggregation function:

$$q_c = g_c(x)$$

(q_c is an r vector). However, although we wish to design an r-th order aggregated model in which q_c is the state, we are willing to increase the dimension of the aggregated model by adding variables q_v to the state in the hope that the inclusion of important dynamic modes in the aggregated model will lead to a better approximation of q. The variables q_v are selected by the designer as a function of the state variables:

$$q_v = g_v(x)$$

(q_v is a p vector). Thus, here we seek an aggregated model of the form:

$$\frac{dq_c}{dt} = f_{qc}(q_c, q_v)$$

$$\frac{dq_v}{dt} = f_{qv}(q_c, q_v)$$

With proper care in selecting the aggregation functions g_c and g_v, so as to avoid algebraic dependencies among the elements of q_c and q_v, one can think of q_c and q_v as being a set or r+p state variables in a new state description. Denoting the new state vector by $\hat{x}$, we have

$$\hat{x} = \begin{bmatrix} q_c \\ q_v \\ x_\Delta \end{bmatrix}$$

where x_Δ is an n-p-r vector that is augmented to q_c and q_v to complete the state description. x_Δ is not unique and can be obtained by an appropriate transformation from the original state description.

$$x_\Delta = g_\Delta(x)$$

Thus, the transformation from x to $\hat{x}$ is given by

$$\hat{x} = \begin{bmatrix} g_c(x) \\ g_v(x) \\ g_\Delta(x) \end{bmatrix} = g_t(x)$$

where g_t is a transformation function and, as such, has an inverse: i.e.

$$x = g_t^{-1}(\hat{x})$$

Differentiating $\hat{x}$ with respect to time gives

$$\frac{d\hat{x}}{dt} = \begin{bmatrix} \nabla_x(g_c(x)) \\ \nabla_x(g_v(x)) \\ \nabla_x(g_\Delta(x)) \end{bmatrix} \frac{dx}{dt}$$

where $\nabla_x(g_c(x))$ is an rxn matrix such that each row of $\nabla_x(g_c(x))$ is the gradient, in the x space, of the corresponding element of $g_c(x)$. Thus for example, the i-th row of $\nabla_x(g_c(x))$ is

$$\nabla_x(g_{ci}(x)) = \left[\frac{\partial g_c}{\partial x_1}\ \frac{\partial g_c}{\partial x_2}\ \cdots\ \frac{\partial g_c}{\partial x_n}\right]$$

Similarly, $\nabla_x(g_v(x))$ is a pxn matrix and $\nabla_x(g_\nabla(x))$ is an (n-p-r)xn matrix. *Since* $\frac{dx}{dt} = f_x(x)$, *we may write*

$$\frac{d\hat{x}}{dt} = \begin{bmatrix} \nabla_x(g_c(x)) \\ \nabla_x(g_v(x)) \\ \nabla_x(g_\Delta(x)) \end{bmatrix} f_x(x)$$

And since $\hat{x} = g_t^{-1}(x)$, we obtain a set of transformed state equations for the original system:

$$\frac{d\hat{x}}{dt} = \begin{bmatrix} \nabla_x(g_c(g_t^{-1}(\hat{x}))) \\ \nabla_x(g_v(g_t^{-1}(\hat{x}))) \\ \nabla_x(g_\Delta(g_t^{-1}(\hat{x}))) \end{bmatrix} f_x(g_t^{-1}(\hat{x}))$$

or, equivalently

$$\begin{aligned} \frac{dq_c}{dt} &= f_{qc}(q_c, q_v, x_\Delta) \\ \frac{dq_v}{dt} &= f_{qv}(q_c, q_v, x_\Delta) \qquad (5) \\ \frac{dx_\Delta}{dt} &= f_{x\Delta}(q_c, q_v, x_\Delta) \end{aligned}$$

where

$$\begin{aligned} f_{qc}(q_c, q_v, x_\Delta) &= \nabla_x(g_c(g_t^{-1}(\hat{x})))f_x(g_t^{-1}(\hat{x})) \\ f_{qv}(q_c, q_v, x_\Delta) &= \nabla_x(g_v(g_t^{-1}(\hat{x})))f_x(g_t^{-1}(\hat{x})) \\ f_{x\Delta}(q_c, q_v, x_\Delta) &= \nabla_x(g_\Delta(g_t^{-1}(\hat{x})))f_x(g_t^{-1}(\hat{x})) \end{aligned}$$

Figure 1 shows the system diagram for this transformed system. Clearly, if g_v and g_t can be selected so that q_c is completely independent of x_Δ, then perfect aggregation is achieved. This independence is equivalent to being able to cut the connection marked with "X" without affecting q_c. Note that perfect aggregation is achieved if $f_{qc}(q_c,q_v,q_\Delta)$ is insensitive to x_Δ. However, in terms of designing an aggregated model, this condition may be much too strong. For example, suppose the system's operation is such that $\frac{dx_\Delta}{dt} \cong 0$ (i.e., $f_{x\Delta}(q_c,q_v,x_\Delta) \cong 0$) and $f_{x\Delta}$ is such that $f_{x\Delta}(q_c,q_v,x_\Delta) = 0$ can be solved for x_Δ in terms of q_c and q_v. In this case although $f_{qv}(q_c,q_v,x_\Delta)$ is a function of x_Δ, the additional relation relating x_Δ to q_c and q_v makes perfect aggregation possible.

In situations where no functions g_v and g_Δ can be found that desensitize q_c to cutting the connections that bring x_Δ to the f_{qc} and f_{qv} blocks, an approximation procedure is suggested which is based on introducing a cutting parameter α. Figure 2 shows the system diagram of equations (5) with such a cutting parameter: $\alpha = 1$ gives the original system: $\alpha = 0$ cuts the connections. In this system, one may use the sensitivity of q_v to the cutting parameter α as an indicator of the effectiveness of aggregation and proceed to look for the functions g_v and g_Δ that minimize this sensitivity. It should be noted that the sensitivity of q_c to α not only depends on the aggregation function g_v but also on the set of state variables selected for x_Δ : i.e., on the selection function g_Δ.

The approach to the computation of the structural sensitivities is straightforward. The outputs q_c are computed with $\alpha = 1$. For large systems, this computation could strain the capacity of the computer being used and, perhaps, prove to be impractical in certain cases. In such situations where the limits of the computer are being tested, it is essential that efficient computational algorithms be used (4, 5). By setting $\alpha = 0$, the smaller proposed aggregated system is separated from the larger system and the outputs of this smaller system, q_{ca}, are computed. The structural sensitivities are simply the differences:

$$\frac{\Delta q_c}{\Delta \alpha} = q_{ca} - q_c$$

Perfect aggregation (i.e., dynamic exactness) is achieved when the structural sensitivities are zero over the time interval of interest. However, given that dynamic exactness is generally not possible, one then seeks the functions g_v and g_Δ that in some way minimize the structural sensitivities. For example, a figure of merit can be defined in terms of the structural sensitivities. There are many possibilities for defining a figure of merit. Some examples are:

$$M_1 = \left| \frac{\Delta q_c}{\Delta \alpha}(1) \right|$$

$$M_2 = \int_0^1 \left(\frac{\Delta q_c}{\Delta \alpha}(\tau)\right)^2 w(\tau)\, d\tau$$

$$M_3 = \int_0^1 \left| \frac{\Delta q_c}{\Delta \alpha}(\tau) \right| w(\tau)\, d\tau$$

$$M_4 = \int_0^1 \left| \frac{\Delta q_c}{\Delta \alpha}(\tau) \cdot \frac{\alpha}{q_c} \right| w(\tau)\, d\tau$$

Figure of merit M_1 stresses the importance of the final value of q_c; M_2 and M_3 consider the accuracy of q_c to be important over the entire time interval (the weighing function w allows the emphasis to vary over the time interval); and M_4 is in terms a normalized sensitivity for situations in which one is concerned with percent errors. Clearly, the figure of merit to be used in any instance depends on the system being modeled.

LINEAR SYSTEMS

Although it is quite unreasonable to expect that any tactical C^3 system could be realistically represented by a linear time-invariant model, it is nevertheless useful to look at the aggregation problem for this special case. Importantly, many large subsystems of tactical C^3 systems are well modeled by linear differential equations and some progress toward obtaining useful aggregated models can be made by aggregating individual subsystems separately. Further, some of the ideas set forth here are rather simply illustrated using linear systems as examples. However, it must be noted that the assumption of linearity gives rise to significant simplifications that do not exist for any other class of systems.

Consider the case that the system of interest is well modeled by tha set of linear differential equations, written in matrix form:

$$\frac{dx}{dt} = Ax \quad (= f_x(x))$$

where x is an n vector and A is an nxn matrix of constants with the element in the i-th row and the j-th column represented by a_{ij}. The variables of interest, which are to be outputs of the aggregated model, are linear combinations of the original set of state variables:

$$q_c = G_c x \quad (= g_c(x))$$

where q_c is an r vector ($r<n$) and G_c is an rxn matrix. Variables q_c are to be state variables of the aggregated model. Additional state variables q_v, where q_v is a p vector ($p<n-r$),

may be allowed in the aggregated model to improve accuracy:

$$q_v = G_v x \quad (= g_v(x))$$

where G_v is a pxn matrix. If q_c and q_v are considered to be state variables in a transformed coordinate system, an additional n-p-r state variables must be selected, also as a linear combinations of the original state variables, to complete the transformed state description:

$$x_\Delta = G_\Delta x \quad (= g_\Delta(x))$$

where x_Δ is an n-p-r vector and G_Δ is an (n-p-r)xn matrix. Thus, representing the new state description with the n vector $\hat{x}$, we have

$$\hat{x} = \begin{bmatrix} G_c \\ G_v \\ G_\Delta \end{bmatrix} x = Gx \quad (= g_t(x))$$

where G is an nxn nonsingular transformation matrix appropriately constructed from submatrices G_c, G_v, and G_Δ. Differentiating the transformation equation $\hat{x}$=Gx with respect to t gives

$$\frac{d\hat{x}}{dt} = GAG^{-1}x = G_{qv\Delta}x$$

where

$$G_{qv\Delta} = GAG^{-1}$$

This can be written in expanded form as

$$\frac{dq_c}{dt} = G_{qq}q_c + G_{qv}q_v + G_{q\Delta}x_\Delta \quad (= f_{qc}(q_c, q_v, x_\Delta))$$

$$\frac{dq_v}{dt} = G_{vq}q_c + G_{vv}q_v + G_{v\Delta}x_\Delta \quad (= f_{qv}(q_c, q_v, x_\Delta)) \qquad (6)$$

$$\frac{dx_\Delta}{dt} = G_{\Delta q}q_c + G_{\Delta v}q_v + G_{\Delta\Delta}x_\Delta \quad (= f_{x\Delta}(q_c, q_v, x_\Delta))$$

where the matrix coefficients of q_c, q_v, x_Δ are the appropriate partitions of $G_{cv\Delta}$. Figure 3 shows the system diagram corresponding to equations (6) with the cutting parameter α included. The objective, of course, is to find the transformation submatrices G_v and G_Δ that minimize the sensitivity of q_c to the cutting parameter α.

A DIGITAL COMPUTER PROGRAM

A FORTRAN program was written which can be used in simplifying linear systems. The program determines the optimal simplified model (i.e., the coefficient matrices G_{cc} and G_{cv}) of specified dimension r+p from the original linear model of dimension n (n>r+p). The program minimizes a figure of merit which is defined in terms of the structural sensitivities and the time interval of interest. The program inputs are the original n-dimensional model (i.e., the coefficient matrix A) and the dimension of the simplified model to be developed. Powell's algorithm (6) is used to execute the minimization.

The program consists of two major parts (see Figure 4). One part consists of the algorithms required to compute the value of the figure of merit from the original model and the proposed aggregated model; the other part consists of the optimization algorithm. Figure 5 details the algorithm for the linear case. Note that the A and G_c matrices are specified as input and G_v and G_Δ are determined from the optimization process. In the program written, Powell's method (6) was used as the basis for the optimization algorithm; Appendix 1 details the optimizatic.. algorithm used to determine the optimal aggregated system (i.e., the matrices G_v and G_Δ) and the corresponding figure of merit. Appendix 2 gives the definitions of the important FORTRAN variables. A listing of the resulting FORTRAN program is given in the Appendix 3. The results of a sample run are given in Appendix 4.

CONCLUSIONS

The research done thus far on using structural sensitivities as a basis for a measure of effectiveness of aggregation seems promising. The FORTRAN program written, which determines aggregated models for linear systems, establishes the feasibility of the approach, at least for an important class of systems. Especially important, insofar as competitive systems is concerned, is that by using structural sensitivities in the design of an aggregated model, attention must be given to all excluded dynamic modes; it is simply not sufficient that the variables of interest appear to be resonably approximated. In minimizing structural sensitivities it is virtually not possible to accidently overlook important system dynamics in the aggregated model.

Use of this FORTRAN program on a variety of examples makes it clear that additional research is required to answer important questions concerning both the effectiveness and the applicability of the approach. As a first step for future research, I suggest a complete rewriting of the aggregation program either in a language such as APL so as to enormously simplify the program or, if again in FORTRAN, on a system with efficient linear algebra software packages. Certainly, having an experienced programmer write the program in a well-documented modular form would be advisable. In addition, the optimization algorithm should be carefully studied for the purpose of improving its covergence properties. Other optimization algorithm should also be explored to determine whether the efficiency of the search can be improved.

With an improved aggregation program, it will be possible to begin to categorize models in terms of whether or not they can be simplified using this approach. By characterizing the properties of models that cannot be aggregated using structural sensitivity measures, insights will be obtained that should allow us to either develop methods to extend the applicability of the approach or to better define the limits of the approach.

Fundamental questions have been raised which should be the subject of future research efforts:

1. To test the feasibility and effectiveness of the proposed approach to aggregation based on structural sensitivities by applying it to model well-known test systems from the literature (e.g., the examples used in references 7 and 8). This should provide a comparison of the proposed approach to aggregation to some of the existing aggregation methods in terms of the effort involved, the usefulness of the resulting model, and the generality of the method.
2. To identify a suitable subsystem of an actual Air Force tatical C^3 system to be the subject of a modeling and simulation effort based on the proposed approach to aggregation.
3. In designing aggregated models of large-scale systems, a minimization must be carried out with respect to a large parameter space. Such minimizations can be difficult,

especially when there are many local minima to contend with. Attention should be given to trying to reduce the dimension of the parameter space by using the least number of parameters possible in defining the transformation functions g_v and g_Δ in this application.

4. By actually cutting the connecting links, the variables x_Δ being fed to the proposed aggregated model are actually set to zero. This is of no concern when dynamic exactness can be achieved. However, when the aggregated model can only generate an approximation to the variables of interest, one should consider the possibility of introducing bias inputs to the aggregated model at the points where the links have been cut.

5. Frequently, one may wish to constrain the form of the aggregated model, even at the cost of having a deteriorated aggregated model or one of higher dimension. For example, one may require that the aggregated model be linear and time-invariant so as to permit analytic studies of the model instead of, or in addition to, simulation studies. Methods for introducing this model constraint into the setting of structural sensitivities should be studied. This possibility was briefly considered and the simple ploy of replacing the first of equations (5) by the following equation seems promising:

$$\frac{dq_c}{dt} = f_q(q_c,q_v) + [f_{qc}(q_c,q_v,x_\Delta) - f_q(q_c,q_v)]$$

where function f_q characterizes the constrained aggregated model. However, further study is necessary to determine how variations in the constrained model parameters affect the dynamics of the sensitivity function.

6. Considerable effort should be directed toward developing rationales for various forms of figures of merits derived from structural sensitivities. Many possibilities come to mind, including integral forms (with and without weighting functions) and those based on final values, and the implications of each ought to be examined, particularly with respect to the relationship of the effectiveness of the aggregated model to the magnitude of the figure of merit.
7. Since the computations of sensitivities are so much simpler for static systems than for dynamic systems, the possibility of designing aggregated models by examining only the right-hand side of the canonical state equations should be carefully investigated.
8. Linear time-invariant systems should be studied as an important special case. Certainly, many important real systems are modeled as linear time-invariant systems. However, the fact that linear time-invariant systems yield to analysis can be quite helpful in developing valuable insights into the implications of structural sensitivities.
9. A system can be defined such that the sensitivities $\Delta q_c/\Delta\alpha$ appear as the system outputs. Using this system, the problem of aggregation can be cast as a control problem in which the sensitivities can be considered to be error sig-

nals to be driven to zero. Such an approach to the aggregation problem should be studied. It seems likely that certain aspects of control theory will prove useful here.

10. The proposed approach to aggregation should be studied with respect to the zero-state response of systems to classical test inputs (e.g., unit steps, sinusoids, etc.). It is clear that the aggregated models depend on the system's initial state. Yet, in many systems it is unlikely that certain system state variables will ever assume significant values because of the large attenuations between the input and the storage devices associated with those state variables. In such cases, determining acceptable aggregated models might be simplest by dealing only with zero-state input-output responses. For the linear case, this is equivalent to looking for aggregated transfer functions.

11. A study should be made on the controllability of systems using controls derived from aggregated models. It seems that such a study makes sense only if a weaker definition of controllability is used so as to take into account the extraordinary controls generally necessary before the neglected dynamics modes can significantly affect the outputs. Particular attention should be given to the role of the extra state variables q_v which are included in the aggregated model only for accuracy. It may be desirable to include some additional state variables for purposes of controllability. For competitive systems, it is especially

important to determine the effect that an opponent can have on controllability and to try to design an aggregated model so as to minimize this effect.

12. A study should be made to extend the results to discrete-event systems.

13. A study should be made to examine the possibility of using structural sensitivity measures for decoupling subsystems. Such an application seems straightforward in that the coupling of subsystems can be represented graphically as links in the system graph. Then, all that is required is to find the transformations to minimize the sensitivity of the subsystem variables to the cutting of these links subject to the constraints that particular system variables be identified with particular subsystems.

REFERENCES

1. E. D. Howes, B. L. Renninger, "Preliminary Notes on a C^3 I Evaluation Model", A MITRE Corp. Memorandum - Memo No. D53-M-142, April 1978.

2. H. D'Angelo, T. G. Windeknecht, "A System Graph and Canonical State Equations", Proceedings Sixth Annual Southeastern Symposium on System Theory, Baton Rouge, 1974.

3. S. I. Mason, "Feedback Theory - Some Properties of Signal Flow Graphs", Proceedings Institute of Radio Engineers, Vol. 41, September 1953, pp 1144-1156.

4. A. Y. Lee, "Signal Flow Graphs - Computer-Aided System Analysis and Sensitivity Calculations", IEEE Transactions on Circuits and Systems, Vol. CAS-21, No. 2, March 1974, pp 209-216.

5. T. G. Windeknecht, A. Y. Lee, H. D'Angelo, "An approach to the Computer-Aided Analysis of Signal-Flow Graphs without the Use of Iterative Methods or Matrix Inversions", Proceedings International Contenance on Cybernetics and Society, 1976.

6. M. J. D. Powell, "A Method for Finding the Minimum of a Function of Several Variables Without Calculating Derivatives", The Computer Journal, Vol. 7, 1964, pp 155-162.

7. E. J. Davison, "AMethod for Simplifying Linear Dynamic Systems", IEEE Transactions on Automatic Control, Vol. AC-11, No. 1, January 1966, pp 93-101.

8. D. A. Wilson, "Optimum Solution of Model-Reduction Problem", Proc. IEE, Vol. 117, No. 6, June 1970, pp 1161-1165

FIGURES

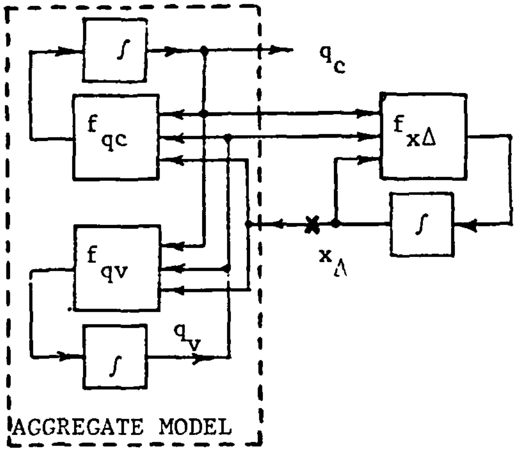

FIGURE 1. SYSTEM DIAGRAM FOR EQUATIONS 5

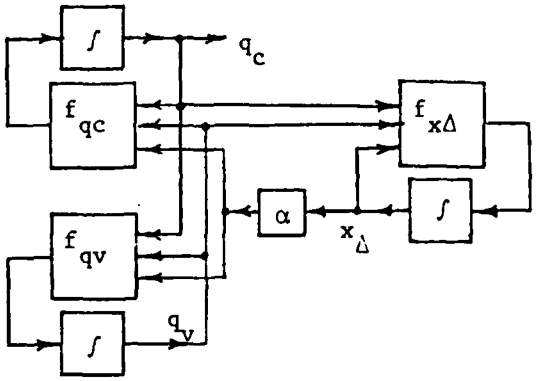

FIGURE 2. INTRODUCTION OF CUTTING PARAMETER

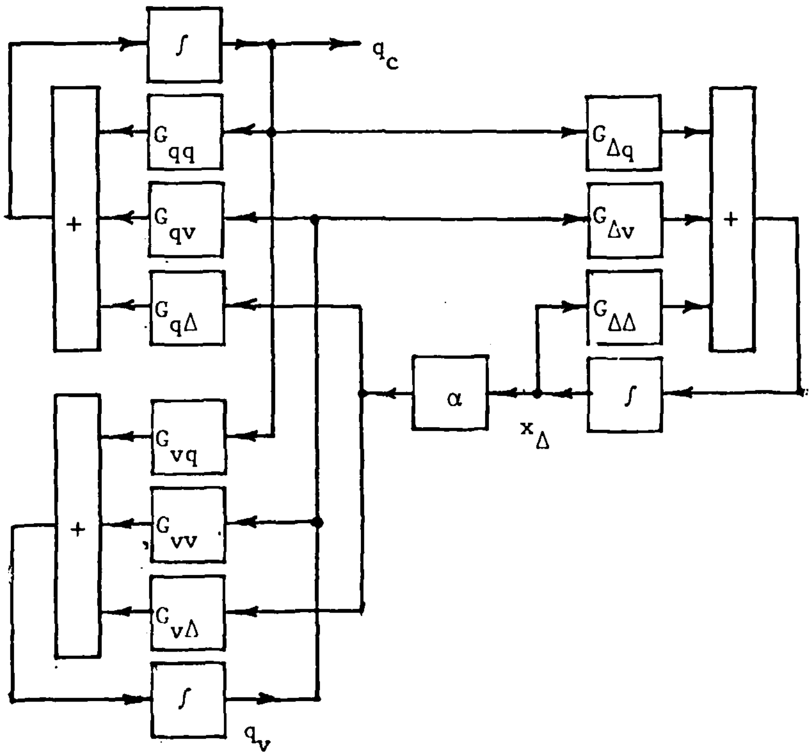

FIGURE 3. LINEAR SYSTEM: SYSTEM DIAGRAM FOR EQUATIONS 6

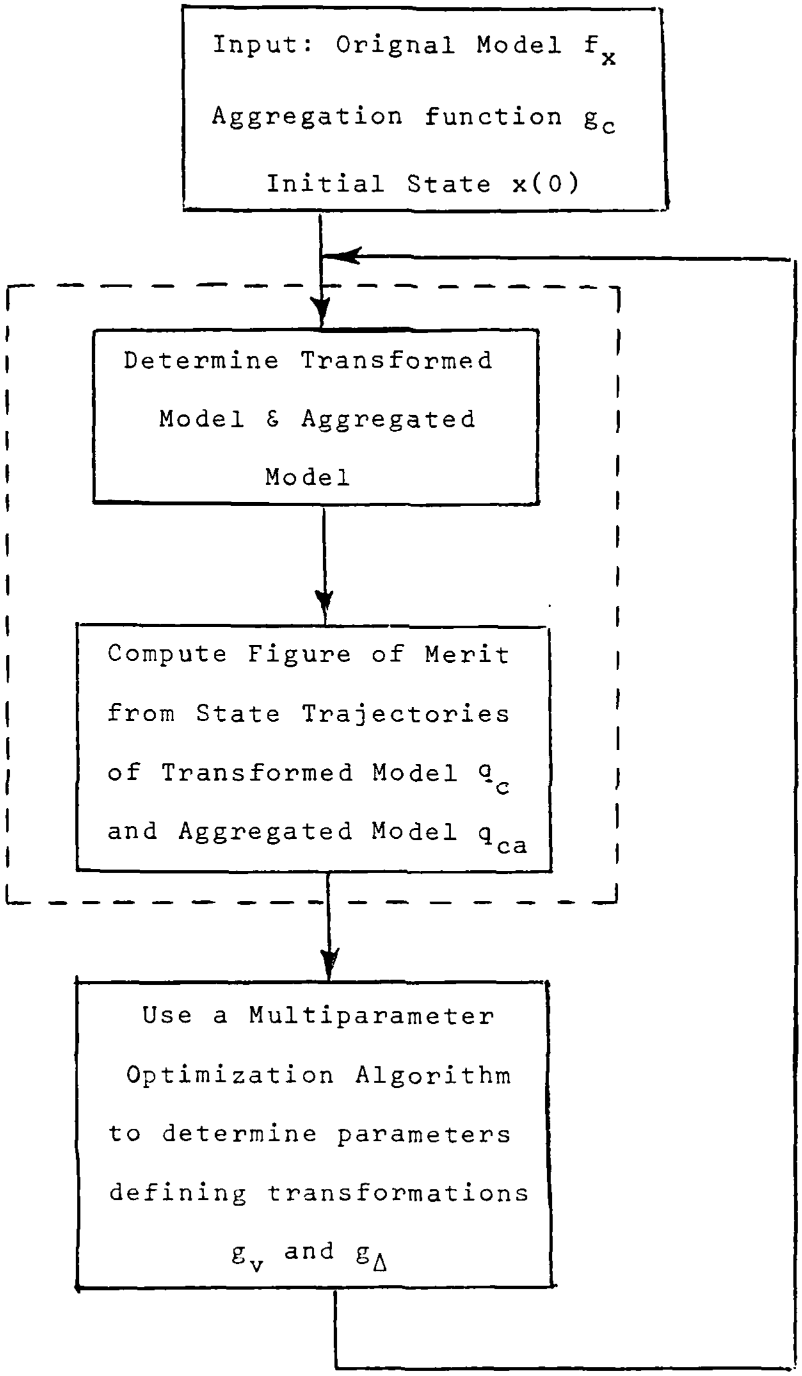

Figure 4 Aggregation Algorithm

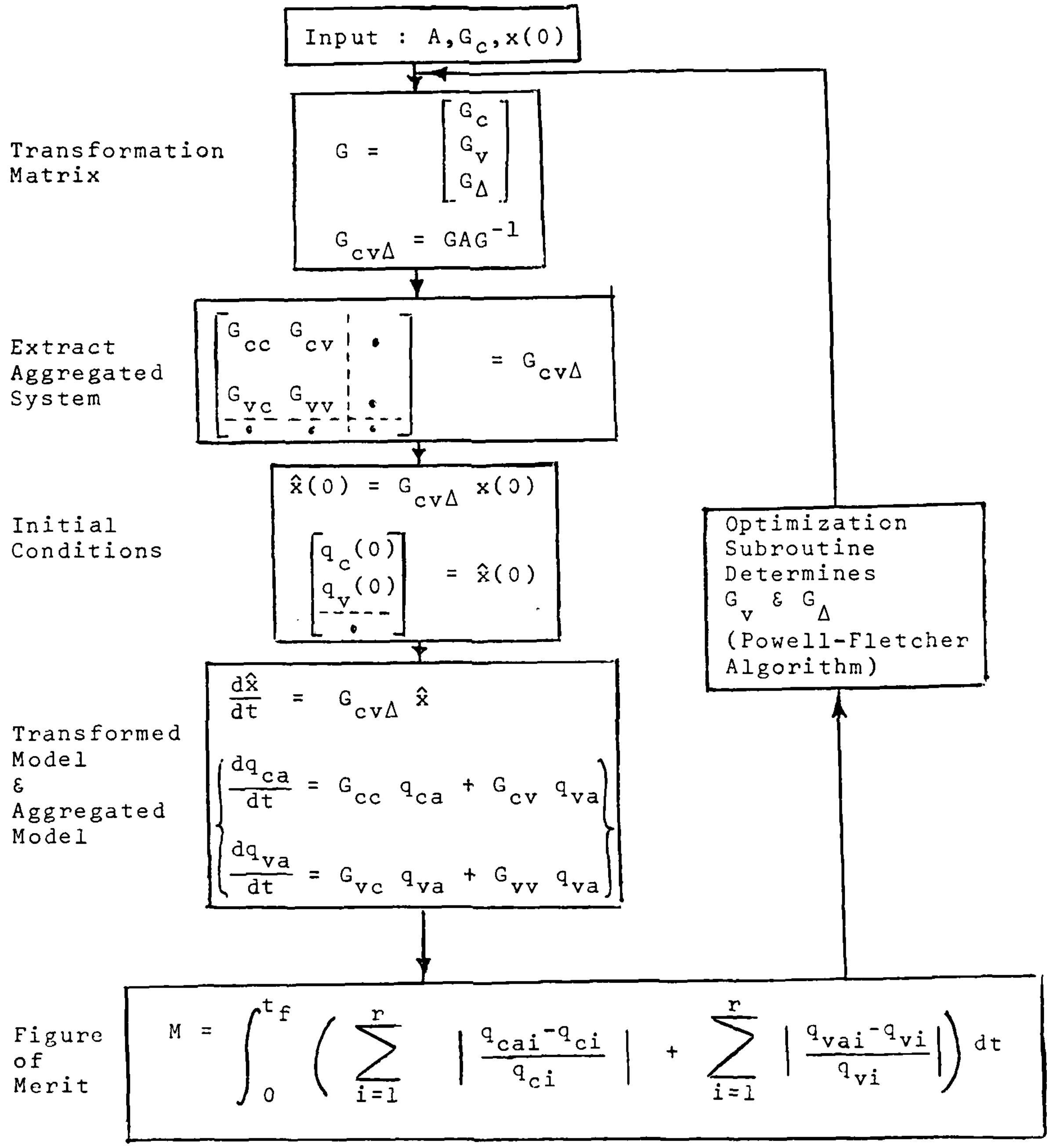

Figure 5 Aggregation Algorithm for Linear Models

APPENDIX 1

Powell's Multiparameter Optimization Method

Powell's method, which is perhaps the most powerful of the direct search methods, has the property that it converges to the exact optimum in a finite number of steps when the function being optimized is a quadratic function. The algorithm finds the maximum value of the function f(x) and the m-vector x giving that maximum. It is based on a series of one-dimensional searches, each defined by an m-dimensional direction vector p. The method of selecting directions of the one dimensional searches lies at the essence of the algorithm.

The algorithm proceeds as follows:

1) For each of the first m iterations, the direction vectors $p^{(1)}$, $p^{(2)}$, ..., $p^{(m)}$ are defined so as to step in each of the n coordinate directions.
2) α is the size of the step taken in direction p. The size of the steps in the first n iterations ($\alpha^{(1)}$,$\alpha^{(2)}$,... , $\alpha^{(m)}$), however, is not predetermined. One moves in each direction until the maximum in that direction is determined. Thus for each step, a single variable search is conducted in one of the m directions, using any efficient single-variable search algorithm. In this program, a Fibonaci search was used.

3) Generate a new set of search direction vectors as follows

$$p^{(1)} = p^{(2)}$$

$$p^{(2)} = p^{(3)}$$

$$\vdots$$

$$p^{(m-1)} = p^{(m)}$$

$$p^{(m)} = x^{(m)} - x^{(0)}$$

4) Define a new starting point $x^{(0)}$ as the maximum in the new $p^{(m)}$ direction

5) Repeat these steps with the new $x^{(0)}$ and the new directions $p^{(1)}, p^{(2)}, \ldots, p^{(m)}$

Powell has modified his algorithm to give better convergence when the initial guess of $x^{(0)}$ is a bad one.

1) Define $p^{(1)}, p^{(2)}, \ldots, p^{(m)}$ in the m coordinate directions. Start at $x^{(0)}$

2) For k=1,2,...m

$$x^{(k)} = x^{(k-1)} + \alpha^{(k)} p^{(k)}$$

where $\alpha^{(k)}$ is selected so as to maximize

$$f(x^{(k-1)} + \alpha^{(k)} p^{(1)}), \max_{\alpha(k)} \left[f(x^{(k-1)} + \alpha^{(k)} p^{(k)}) \right] \equiv f(x^{(k)})$$

3) Find integer j, $1 \leq j \leq m$, so that $f(x^{(j)}) - f(x^{(j-1)})$ is a maximum and define $\Delta \equiv f(x^{(j)}) - f(x^{(j-1)})$

4) Compute

$$f_3 = f(x^{(0)} + 2(x^{(m)} - x^{(0)})) = f(2x^{(m)} - x^{(0)})$$

Define

$$f_1 = f(x^{(0)})$$

$$f_2 = f(x^{(m)})$$

5) If <u>either</u>

$$f_3 \geq f_1$$

<u>or</u>

$$(f_1-2f_2+f_3)(f_1-f_2-\Delta)^2 \geq \tfrac{1}{2}\Delta\cdot(f_1-f_3)^2$$

use old directions $p^{(1)}, p^{(2)}, \ldots, p^{(m)}$ and new $x^{(0)} = x$ for the next iteration.

6) If neither inequalities in (5) are true, define

$$p \equiv (x^{(m)}-x^{(0)}),$$

look for new $x^{(0)}=x^{(m)}+\alpha p$ where α minimizes $f(x^{(m)}+\alpha p)$ and define new direction vectors $p^{(1)}, p^{(2)}, \ldots, p^{(j-1)}, \ldots, p^{(m)}, p$

APPENDIX 2

Important FORTRAN Variables

A = an n^2 vector derived from the A matrix

$= a_{11}\ a_{21} \cdots a_{n1}\ a_{12}\ a_{22} \cdots a_{n2} \cdots a_{1n}\ a_{2n} \cdots a_{nn}$

GC = an r•n vector derived from the G_c matrix

$= g_{c11}\ g_{c21} \cdots g_{cr1} \cdots g_{c1n}\ g_{c2n} \cdots g_{crn}$

GV = a p•n vector derived from the the G_v matrix

$= g_{v11}\ g_{v21} \cdots g_{vp1} \cdots g_{v1n}\ g_{c2n} \cdots g_{cpn}$

GD = a (n-r-p)•n vector derived from the G_Δ matrix

$= g_{\Delta 11}\ g_{\Delta 21} \cdots g_{\Delta(n-r-p)1} \cdots g_{\Delta 1n}\ g_{\Delta 2n} \cdots g_{\Delta(n-r-p)n}$

Y = a p•n + (n-r-p)•n vector derived from the G_v and the G_Δ matrices (Y is the vector to be determined in the minimization process)

$= [GV \vdots GD]$

G = an n^2 vector derived from the G_c, G_v, and G_Δ matrices

$= [GC \vdots GV \vdots GD]$

GAG = an n^2 vector derived from the $G_{cv\Delta}$ matrix

($G_{cv\Delta}$ = GAG^{-1})

X = an n vector representing the state of the original model

(X = x)

XHAT = an vector representing the state of the transformed model

(XHAT = $\hat{x}$)

APPENDIX 3

FORTRAN Program

```
      COMMON S(10),Y(10),P(10,10),V(10),R(10),X(10,10),FUNCT(10)
      COMMON E0,XX,H,JE,F0,F,F9,C9,F3,D,C8,N,J5,FGRAD
      COMMON/XXX/A(100),XS(10),GC(10),NN,RR,PP,DELT,FMERIT,TFINAL
      COMMON/GET1/GCC
      COMMON/TRAJEC/JOUT
      DIMENSION GCC(10)
      INTEGER RR,PP
      JE=0
      JOUT=0
C(150)  INPUT CONVERGENCE DATA(730)
      CALL CONVDATA
C(350)  INPUT INITIAL PARAMETERS(620)
      CALL INPPARAM
C  INPUT SYSTEM PARAMETERS
      CALL SYSPAR(A,XS,GC,RR,PP,DELT,TFINAL,NN)
C(230)  OUTPUT INITIAL PARAMETERS AND CONVERGENCE DATA(840)
      CALL OUTPARAM
C  ESTIMATION OF PARAMETERS
C  ************************
C  ************************
C(370)  COMPUTE INITIAL F(1820)
      CALL CMPINITF
C(390)  INITIALIZE DIRECTION VECTOR P(1920)
      CALL INDIRVEC
C(410)  STEP(2020) IN EACH DIRECTION P
410   CALL STEPP
C(430)  CHECK FOR CONVERGENCE(2220)
      CALL CHEKCONV
C   C9=1 IMPLIES CONVERGENCE
      IF(C9.EQ.1) GO TO 590
C(450) FIND DIRECTION GIVING GREATEST INCREASE IN F(2420)
      CALL RAPIDF
C(470)  COMPUTE TEST POINT F(2X(N)-X(0))(2530)
      CALL TESTPNT
C(490)  CHECK WHETHER DIRECTION VECTORS ARE TO BE CHANGED(2600)
      CALL CHKDIRVK
C   C8=1 IMPLIES DIRECTION VECTORS ARE TO BE CHANGED
      IF(C8.EQ.1) GO TO 509
C(504)  DON'T CHANGE DIRECTION VECTORS(2710)
      CALL DIRVECOK
C(507)  REPEAT PROCESS WITH NEW STARTING VALUES
      GO TO 410
C(509)  CHANGE DIRECTION VECTORS(2820)
509   CALL NEWDIREC
C(530)  REVISE ESTIMATES OF PARAMETERS(2910)
      CALL REVEST
```

```
C(540) COMPUTE NEW DIRECTION VECTORS P(2960)
       CALL COMPNEWP
C(570)  REPEAT ESTIMATION PROCESS WITH NEW STARTING ESTIMATES
       GO TO 410
590    CONTINUE
       STOP
       END
730    SUBROUTINE CONVDATA
       COMMON S(10),Y(10),P(10,10),V(10),R(10),X(10,10),FUNCT(10)
       COMMON EO,XX,H,JE,FO,F,F9,C9,F3,D,C8,N,J5,FGRAD
C  N=NUMBER OF PARAMETERS
C EO=THE CONVERGENCE FACTOR FOR THE MAIN MINIMIZATION
C  CONVERGENCE IMPLIES  ABS(SUM(XN-XO)) < EO
C  H=THE CONVERGENCE FACTOR FOR THE ONE-DIMENSIONAL SEARCH
C  JE=THE NUMBER OF TIMES THE FUNCTION IS COMPUTED
       READ,N,EO,H,FGRAD
       RETURN
       END
620    SUBROUTINE INPPARAM
       COMMON S(10),Y(10),P(10,10),V(10),R(10),X(10,10),FUNCT(10)
       COMMON EO,XX,H,JE,FO,F,F9,C9,F3,D,C8,N,J5,FGRAD
       READ,(S(I),I=1,N)
       DO 630 I=1,N
630    X(1,I)=S(I)
       RETURN
       END
840    SUBROUTINE OUTPARAM
       COMMON S(10),Y(10),P(10,10),V(10),R(10),X(10,10),FUNCT(10)
       COMMON EO,XX,H,JE,FO,F,F9,C9,F3,D,C8,N,J5,FGRAD
       WRITE(01,842)N
842    FORMAT('THE NUMBER OF PARAMETERS IS',I5)
       WRITE(01,843)EO
843    FORMAT(' CONVERGENCE FACTOR FOR  MAIN MINIMIZATION :',F10.7)
       WRITE(01,844)H
844    FORMAT(' CONVERGENCE FACTOR FOR  ONE-DIMENSIONAL SEARCH :',F10.7)
       WRITE(01,845)FGRAD
845    FORMAT(' GRADIENT CUTOFF FOR ONE-DIM. SEARCH :',F10.7)
       WRITE(01,850)N
850    FORMAT('THE ',I5,' INITIAL PARAMETER VALUES ARE:')
       WRITE(01,855)(S(I),I=1,N)
855    FORMAT(10E12.4)
       RETURN
       END
1820   SUBROUTINE CMPINITF
       COMMON S(10),Y(10),P(10,10),V(10),R(10),X(10,10),FUNCT(10)
       COMMON EO,XX,H,JE,FO,F,F9,C9,F3,D,C8,N,J5,FGRAD
       COMMON/XXX/A(100),XS(10),GC(10),NN,RR,PP,DELT,FMERIT,TFINAL
       INTEGER RR,PP
       DO 1850 I=1,N
1850   Y(I)=S(I)
       CALL FUNCTION
       FO=F
       RETURN
       END
```

```
4970  SUBROUTINE FUNCTION
      COMMON S(10),Y(10),P(10,10),V(10),R(10),X(10,10),FUNCT(10)
      COMMON EO,XX,H,JE,FO,F,F9,C9,F3,D,C8,N,J5,FGRAD
      COMMON/XXX/A(100),XS(10),GC(10),NN,RR,PP,DELT,FMERIT,TFINAL
      INTEGER RR,PP
      JE=JE+1
      CALL MERIT(A,Y,GC,XS,NN,RR,PP,DELT,FMERIT,TFINAL)
      F=-FMERIT
4972  CONTINUE
      RETURN
      END
1920  SUBROUTINE INDIRVEC
      COMMON S(10),Y(10),P(10,10),V(10),R(10),X(10,10),FUNCT(10)
      COMMON EO,XX,H,JE,FO,F,F9,C9,F3,D,C8,N,J5,FGRAD
      DO 1950 I=1,N
      DO 1950 J=1,N
1950  P(I,J)=0
      DO 2000 I=1,N
2000  P(I,I)=1
      RETURN
      END
2020  SUBROUTINE STEPP
      COMMON S(10),Y(10),P(10,10),V(10),R(10),X(10,10),FUNCT(10)
      COMMON EO,XX,H,JE,FO,F,F9,C9,F3,D,C8,N,J5,FGRAD
      DO 2190 I=1,N
      DO 2120 J=1,N
      V(J)=P(I,J)
      IF(I.GT.1)GO TO 2110
      R(J)=S(J)
      GO TO 2120
2110  R(J)=X(I-1,J)
2120  CONTINUE
      CALL MAX
      FUNCT(I)=F
      DO 2180 J=1,N
2180  X(I,J)=R(J)+XX*V(J)
2190  CONTINUE
      F9=FUNCT(N)
      RETURN
      END
4180  SUBROUTINE MAX
      COMMON S(10),Y(10),P(10,10),V(10),R(10),X(10,10),FUNCT(10)
      COMMON EO,XX,H,JE,FO,F,F9,C9,F3,D,C8,N,J5,FGRAD
      REAL M1,M2
      X4=0.1
      XO=0
4200  M1=XO-X4
      M2=XO+X4
      XX=XO
      CALL STEPFUNC
      QO=F
      XX=M1
      CALL STEPFUNC
      Q1=F
```

```
      XX=M2
      CALL STEPFUNC
      Q2=F
4330  IF(Q0.GT.Q1)GO TO 4400
      IF(Q0.GT.Q2)GO TO 4480
      WRITE(01,4300)
4300  FORMAT('MINIMUM DISCOVERED IN SUBROUTINE MAX')
      WRITE(6,4302)Q1,Q0,Q2
4302  FORMAT('Q1=',E12.4,'  Q0=',E12.4,'  Q2=',E12.4)
      IF(Q2.LT.Q1)GO TO 4350
      X0=M2
      GO TO 4200
4350  X0=M1
      GO TO 4200
      STOP
4400  IF(Q0.GT.Q2)GO TO 4550
4410  X4=2*X4
      M1=X0
      Q1=Q0
      X0=M2
      Q0=Q2
      M2=X0+X4
      FOLD=Q0
      XX=M2
      CALL STEPFUNC
      Q2=F
      IF(ABS(F-FOLD).GT.FGRAD)GO TO 4430
      GO TO 4570
4430  GO TO 4400
4480  X4=2*X4
      M2=X0
      Q2=Q0
      X0=M1
      Q0=Q1
      M1=X0-X4
      FOLD=Q0
      XX=M1
      CALL STEPFUNC
      Q1=F
      IF(ABS(F-FOLD).GT.FGRAD)GO TO 4485
      GO TO 4570
4485  GO TO 4330
4550  A=M1
      B=M2
      GO TO 4580
C  ************************
C  ******* **************
C  ONE-DIMENSIONAL SEARCH AN INTERVAL (A,B)
4580  ANEW=(A+B)/2-0.1*(B-A)
      BNEW=(A+B)/2+0.1*(B-A)
4582  FORMAT(2E15.5)
      XX=ANEW
      CALL STEPFUNC
      Q1NEW=F
      XX=BNEW
```

```
      CALL STEPFUNC
      Q2NEW=F
      IF(Q1NEW.GT.Q2NEW)GO TO 4585
      A=ANEW
      GO TO 4590
4585  B=BNEW
4590  IF((B-A).GT.H)GO TO 4580
      XX=(B+A)/2
      CALL STEPFUNC
4570  CONTINUE
      RETURN
      END
4760  SUBROUTINE STEPFUNC
      COMMON S(10),Y(10),P(10,10),V(10),R(10),X(10,10),FUNCT(10)
      COMMON E0,XX,H,JE,F0,F,F9,C9,F3,D,C8,N,J5,FGRAD
      DO 4790 I=1,N
4790  Y(I)=R(I)+XX*V(I)
      CALL FUNCTION
4795  FORMAT(4E15.5)
      RETURN
      END
2220  SUBROUTINE CHEKCONV
      COMMON S(10),Y(10),P(10,10),V(10),R(10),X(10,10),FUNCT(10)
      COMMON E0,XX,H,JE,F0,F,F9,C9,F3,D,C8,N,J5,FGRAD
      COMMON/GET1/GCC
      COMMON/TRAJEC/JOUT
      DIMENSION GCC(10)
      IF(ABS(F0-F9).GT.FGRAD)GO TO 2250
      WRITE(01,2240)
2240  FORMAT('FGRAD > ABS(F0-F9)')
      C9=1
      GO TO 2298
2250  SUM=0
      DO 2260 J=1,N
2260  SUM=SUM+ABS(X(N,J)-S(J))
      WRITE(01,2290)SUM,F
2290  FORMAT('SUM=',E12.4,'  F=',E12.4)
      WRITE(01,2294)N
2294  FORMAT('THE',I5,'  PARAMETER VALUES ARE:')
      WRITE(01,2295)(X(N,J),J=1,N)
2295  FORMAT(10E12.4)
      IF(SUM.GT.E0) GO TO 2325
      C9=1
2298  CALL SUMMARY
      JOUT=1
      CALL FUNCTION
      GO TO 2330
2325  C9=0
2330  RETURN
      END
2340  SUBROUTINE SUMMARY
      COMMON S(10),Y(10),P(10,10),V(10),R(10),X(10,10),FUNCT(10)
      COMMON E0,XX,H,JE,F0,F,F9,C9,F3,D,C8,N,J5,FGRAD
      COMMON/GET1/GCC
```

```
      COMMON/XXX/A(100),XS(10),GC(10),NN,RR,PP,DELT,FMERIT,TFINAL
      DIMENSION GCC(10)
      INTEGER RR,PP
      WRITE(01,2350)F
2350  FORMAT('THE FUNCTION VALUE IS',E15.5)
      CALL GETOUT(GCC,Y,N,NN,RR,PP)
      WRITE(01,2390)JE
2390  FORMAT('THE FUNTION WAS COMPUTED   ',I10,'  TIMES.')
      RETURN
      END
2420  SUBROUTINE RAPIDF
      COMMON S(10),Y(10),P(10,10),V(10),R(10),X(10,10),FUNCT(10)
      COMMON E0,XX,H,JE,F0,F,F9,C9,F3,D,C8,N,J5,FGRAD
      DI=FUNCT(1)-F0
      D=DI
      J5=1
      IF(N.EQ.1)GO TO 2515
      DO 2510 K=2,N
      DI=FUNCT(K)-FUNCT(K-1)
      IF(DI.LT.D)GO TO 2510
      D=DI
      J5=K
2510  CONTINUE
2515  CONTINUE
      RETURN
      END
2530  SUBROUTINE TESTPNT
      COMMON S(10),Y(10),P(10,10),V(10),R(10),X(10,10).FUNCT(10)
      COMMON E0,XX,H,JE,F0,F,F9,C9,F3,D,C8,N,J5,FGRAD
      DO 2560 J=1,N
2560  Y(J)=2*X(N,J)-S(J)
      CALL FUNCTION
      F3=F
      RETURN
      END
2600  SUBROUTINE CHKDIRVK
      COMMON S(10).Y(10),P(10,10),V(10),R(10),X(10,10).FUNCT(10)
      COMMON E0,XX,H,JE.F0,F,F9,C9.F3,D,C8,N,J5,FGRAD
      F5=(F0-2*F9+F3)*(F0-F9-D)**2
      F6=(1/2)*D*(F0-F3)**2
      IF(F3.LT.F0)GO TO 2670
      IF(F5.LT.F6)GO TO 2690
2670  C8=0
      GO TO 2700
2690  C8=1
2700  RETURN
      END
2710  SUBROUTINE DIRVECOK
      COMMON S(10),Y(10),P(10,10),V(10),R(10),X(10,10),FUNCT(10)
      COMMON E0,XX,H,JE,F0,F,F9,C9,F3,D,C8,N,J5,FGRAD
      DO 2740 J=1,N
2740  S(J)=X(N,J)
      F0=F9
      RETURN
      END
```

```
2820  SUBROUTINE NEWDIREC
      COMMON S(10),Y(10),P(10,10),V(10),R(10),X(10,10),FUNCT(10)
      COMMON E0,XX,H,JE,F0,F,F9,C9,F3,D,C8,N,J5,FGRAD
      DO 2860 J=1,N
      V(J)=X(N,J)-S(J)
2860  R(J)=X(N,J)
      CALL MAX
      F0=F
      RETURN
      END
2910  SUBROUTINE REVEST
      COMMON S(10),Y(10),P(10,10),V(10),R(10),X(10,10),FUNCT(10)
      COMMON E0,XX,H,JE,F0,F,F9,C9,F3,D,C8,N,J5,FGRAD
      DO 2940 J=1,N
2940  S(J)=R(J)+XX*V(J)
      RETURN
      END
2960  SUBROUTINE COMPNEW
      COMMON S(10),Y(10),P(10,10),V(10),R(10),X(10,10),FUNCT(10)
      COMMON E0,XX,H,JE,F0,F,F9,C9,F3,D,C8,N,J5,FGRAD
      IF(J5.EQ.N)GO TO 3050
      J6=N-1
      DO 3010 I=J5,J6
      DO 3010 J=1,N
3010  P(I,J)=P(I+1,J)
      DO 3040 J=1,N
3040  P(N,J)=V(J)
3050  RETURN
      END
      SUBROUTINE GETOUT(GCC,Y,N,NN,RR,PP)
      INTEGER RR,PP,RP
      DIMENSION GCC(10),Y(10)
      WRITE(01,10)N
10    FORMAT('THE ',I5,'  FINAL PARAMETER VALUES ARE:')
      WRITE(01,12)(Y(I),I=1,N)
12    FORMAT(10E12.4)
      RP=RR+PP
      WRITE(01,15)RR,RR
15    FORMAT('THE  ',I5,' X ',I5,'  AGGREGATED SYST. COEF. MATRIX:')
      DO 100 I=1,RP
      J1=I
      J2=(RP-1)*RP+I
      WRITE(01,200)(GCC(J),J=J1,J2,RP)
100   CONTINUE
200   FORMAT(10E12.4)
      RETURN
      END
      SUBROUTINE MERIT(A,Y,GC,XINIT,N,R,P,DELT,FMERIT,TFINAL)
      DIMENSION A(100),X(10),Y(10),GC(10),GD(10),G(100),GA(100)
      DIMENSION XINIT(10),XHAT(10),QCB(10),QCA(10)
      DIMENSION QC(10),LXX(10),MXX(10),GV(10),GAG(100),GCC(10)
      COMMON/GET1/GCC
      COMMON/TRAJEC/JOUT
      INTEGER R,P
```

```
C  EXTRACT GV FROM Y
      K=0
      IF(P.EQ.0)GO TO 1500
      DO 1001 I=1,P
      DO 1001 J=1,N
      K=K+1
1001  GV(K)=Y((I-1)*P+J)
1500  CONTINUE
C EXTRACT GD FROM Y
      NRP=N-R-P
      IF(NRP.EQ.0)GO TO 2010
      K=0
      DO 2000 I=1,NRP
      K=K+1
      GD((I-1)*N+1)=1
      NM1=N-1
      DO 2000 J=1,NM1
      K=K+1
2000  GD(K)=Y(P*N+(I-1)*NRP+J)
2010  CONTINUE
C  FORM G FROM GC, GV, AND GD
      K=0
      DO 6000 I=1,N
      DO 3000 J=1,R
      K=K+1
3000  G(K)=GC((I-1)*R+J)
      IF(P.EQ.0)GO TO 4500
      DO 4000 J=1,P
      K=K+1
4000  G(K)=GV((I-1)*P+J)
4500  CONTINUE
      IF(NRP.EQ.0)GO TO 6000
      DO 5000 J=1,NRP
      K=K+1
5000  G(K)=GD((I-1)*NRP+J)
6000  CONTINUE
C  COMPUTE XHAT(0)
      CALL GMPRD(G,XINIT,XHAT,N,N,1)
C  EXTRACT QCA(0) FROM XHAT(0)
      K=0
      DO 6555 I=1,R
      K=K+1
6555  QCA(K)=XHAT(K)
      T=0
      IF(JOUT.EQ.0)GO TO 6666
      WRITE(01,9550)T,(QCA(I).QCA(I),I=1,R)
6666  CONTINUE
C OBTAIN GAG
      CALL GMPRD(G.A,GA,N,N,N)
      CALL MINV(G,N.DXX,LXX.MXX)
      CALL GMPRD(GA.G,GAG,N,N,N)
C  EXTRACT GCC FROM GAG
      K=0
      KRP=R+P
```

```
      DO 7000 I=1,KRP
      DO 7000 J=1,KRP
      K=K+1
7000  GCC(K)=GAG((I-1)*N+J)
      FMERIT=0
      T=0
8000  IF(T.GT.TFINAL)GO TO 9999
C  XHAT(T+DELT)=XHAT(T)+DELT*(GAG*XHAT(T))
      CALL GMPRD(GAG,XHAT,X,N,N,1)
      DO 9000 I=1,N
9000  X(I)=DELT*X(I)
      CALL GMADD(XHAT,X,XHAT,N,1)
C  EXTRACT QC FROM XHAT
      K=0
      DO 9300 I=1,R
      K=K+1
9300  QC(K)=XHAT(K)
C  QCA(T+DELT)=QCA(T)+DELT*(GCC*QCA(T))
      CALL GMPRD(GCC,QCA,QCB,R,R,1)
      DO 9500  I=1,R
9500  QCB(I)=DELT*QCB(I)
      CALL GMADD(QCA,QCB,QCA,R,1)
C  COMPUTE FMERIT
      DO 9600 I=1,R
      FMERIT=FMERIT+DELT*ABS((QCA(I)-QC(I))/QC(I))
9600  CONTINUE
      IF(JOUT.EQ.0)GO TO 9555
      TD=T+DELT
      WRITE(01,9550)TD,(QC(I),QCA(I),I=1,R)
9550  FORMAT(F10.6,10E12.4)
9555  CONTINUE
      T=T+DELT
      GO TO 8000
9999  CONTINUE
      RETURN
      END
      SUBROUTINE SYSPAR(A,X,GC,R,P,DELT,TFINAL,N)
      DIMENSION A(100),X(10),GC(10)
      INTEGER R,P
      READ, N,R,P,DELT,TFINAL
      WRITE(01,39)N
39    FORMAT('DIMENSION OF ORIGINAL LINEAR SYSTEM:',I5)
      NN=N*N
      WRITE(01,79)R
79    FORMAT('NUMBER OF STATE VARIABLES OF INTEREST:  ',I5)
      WRITE(01,78)P
78    FORMAT('NUMBER OF EXTRA STATE VARIABLES:  ',I5)
      WRITE(01,77)DELT,TFINAL
77    FORMAT('INTEGRATION INTERVAL=',F10.5,'  TIME INTERVAL=',F10.5)
      WRITE(01,75)
75    FORMAT('ORIGINAL SYSTEM A-MATRIX:')
      DO 70 I=1,N
      J1=I
      J2=(N-1)*N+I
```

```
      READ,(A(J),J=J1,J2,N)
70    WRITE(01,76)(A(J),J=J1,J2,N)
76    FORMAT(10F12.4)
      READ,(X(I),I=1,N)
      WRITE(01,55)
55    FORMAT('THE INITIAL STATE IS:')
      WRITE(01,59)(X(I),I=1,N)
59    FORMAT(10E12.4)
      WRITE(01,65)R,N
65    FORMAT('THE ',I5,' X ',I5,' AGGREGATION MATRIX:')
      DO 60 I=1,R
      J1=I
      J2=(N-1)*R+I
      READ,(GC(J),J=J1,J2,R)
60    WRITE(01,76)(GC(J),J=J1,J2,R)
      RETURN
      END
/INC GMPRD,MINV,GMADD
/DATA
2,.01,.01,.000001
0,0,0
3,2,0,.05,1
0,1,0
-257.9047,-10,0
1,1,-5
1,1,1
1,.00001,.00001
.00001,1,.00001
```

APPENDIX 4

A Sample Run

```
DIMENSION OF ORIGINAL LINEAR SYSTEM:    3
NUMBER OF STATE VARIABLES OF INTEREST:      2
NUMBER OF EXTRA STATE VARIABLES:      0
INTEGRATION INTERVAL=   0.01000  TIME INTERVAL=   1.00000
ORIGINAL SYSTEM A-MATRIX:
      0.0000       1.0000       0.0000
   -257.9048     -10.0000       0.0000
      1.0000       1.0000      -5.0000
THE INITIAL STATE IS:
 -0.1000E 01 -0.1000E 01 -0.1000E 01
THE      2 X     3 AGGREGATION MATRIX:
      1.0000       1.0000       1.0000
      1.0000       0.0000       0.0000
THE NUMBER OF PARAMETERS IS    2
 CONVERGENCE FACTOR FOR  MAIN MINIMIZATION : 0.0100000
 CONVERGENCE FACTOR FOR  ONE-DIMENSIONAL SEARCH : 0.0100000
 GRADIENT CUTOFF FOR ONE-DIM. SEARCH : 0.0000010
THE      2 INITIAL PARAMETER VALUES ARE:
  0.0000E 00  0.0000E 00
SUM=  0.1071E 01  F= -0.2782E 00
THE     2  PARAMETER VALUES ARE:
  0.2383E-01 -0.1047E 01
SUM=  0.1332E 00  F= -0.5726E-01
THE     2  PARAMETER VALUES ARE:
  0.1567E-01 -0.9223E 00
SUM=  0.9421E-02  F= -0.3173E-01
THE     2  PARAMETER VALUES ARE:
  0.1692E-01 -0.9304E 00
THE FUNCTION VALUE IS   -0.31729E-01
THE      2  FINAL PARAMETER VALUES ARE:
  0.1692E-01 -0.9304E 00
THE       2 X     2  AGGREGATED SYST. COEF. MATRIX:
 -0.7946E 01 -0.2458E 03
  0.9821E 00 -0.2038E 01
THE FUNTION WAS COMPUTED          116  TIMES.
```

T (TIME UNITS)	Q1(T) (ORIGINAL SYSTEM VARIABLE)	QA1(T) (AGGRE-GATED SYSTEM VARIABLE)	Q2(T) (ORIGINAL SYSTEM VARIABLE)	QA2(T) (AGGREGATED SYSTEM VARIABLE)
0.000000	-0.3000E 01	-0.3000E 01	-0.1000E 01	-0.1000E 01
0.010000	-0.3010E 00	-0.3037E 00	-0.1010E 01	-0.1009E 01
0.020000	0.2208E 01	0.2201E 01	-0.9932E 00	-0.9915E 00
0.030000	0.4476E 01	0.4463E 01	-0.9520E 00	-0.9497E 00
0.040000	0.6463E 01	0.6442E 01	-0.8894E 00	-0.8865E 00
0.050000	0.8137E 01	0.8109E 01	-0.8084E 00	-0.8052E 00
0.060000	0.9479E 01	0.9444E 01	-0.7127E 00	-0.7091E 00
0.070000	0.1048E 02	0.1044E 02	-0.6056E 00	-0.6019E 00
0.080000	0.1114E 02	0.1109E 02	-0.4909E 00	-0.4871E 00
0.090000	0.1146E 02	0.1140E 02	-0.3720E 00	-0.3683E 00
0.100000	0.1146E 02	0.1140E 02	-0.2523E 00	-0.2488E 00
0.110000	0.1118E 02	0.1111E 02	-0.1350E 00	-0.1318E 00
0.120000	0.1062E 02	0.1055E 02	-0.2299E-01	-0.2000E-01

0.130000	0.9834E 01	0.9760E 01	0.8135E-01	0.8402E-01
0.140000	0.8854E 01	0.8778E 01	0.1758E 00	0.1782E 00
0.150000	0.7720E 01	0.7642E 01	0.2588E 00	0.2607E 00
0.160000	0.6472E 01	0.6394E 01	0.3289E 00	0.3305E 00
0.170000	0.5150E 01	0.5074E 01	0.3853E 00	0.3866E 00
0.180000	0.3795E 01	0.3721E 01	0.4276E 00	0.4285E 00
0.190000	0.2444E 01	0.2372E 01	0.4558E 00	0.4563E 00
0.200000	0.1130E 01	0.1062E 01	0.4701E 00	0.4703E 00
0.210000	-0.1136E 00	-0.1788E 00	0.4712E 00	0.4712E 00
0.220000	-0.1262E 01	-0.1323E 01	0.4601E 00	0.4598E 00
0.230000	-0.2291E 01	-0.2348E 01	0.4379E 00	0.4374E 00
0.240000	-0.3184E 01	-0.3236E 01	0.4061E 00	0.4055E 00
0.250000	-0.3928E 01	-0.3976E 01	0.3662E 00	0.3654E 00
0.260000	-0.4515E 01	-0.4558E 01	0.3198E 00	0.3189E 00
0.270000	-0.4941E 01	-0.4980E 01	0.2685E 00	0.2677E 00
0.280000	-0.5207E 01	-0.5242E 01	0.2142E 00	0.2133E 00
0.290000	-0.5319E 01	-0.5350E 01	0.1584E 00	0.1575E 00
0.300000	-0.5285E 01	-0.5312E 01	0.1026E 00	0.1017E 00
0.310000	-0.5116E 01	-0.5139E 01	0.4834E-01	0.4748E-01
0.320000	-0.4828E 01	-0.4848E 01	-0.3158E-02	-0.3961E-02
0.330000	-0.4436E 01	-0.4453E 01	-0.5075E-01	-0.5149E-01
0.340000	-0.3958E 01	-0.3972E 01	-0.9350E-01	-0.9418E-01
0.350000	-0.3413E 01	-0.3425E 01	-0.1307E 00	-0.1313E 00
0.360000	-0.2820E 01	-0.2830E 01	-0.1617E 00	-0.1622E 00
0.370000	-0.2198E 01	-0.2207E 01	-0.1863E 00	-0.1867E 00
0.380000	-0.1565E 01	-0.1572E 01	-0.2042E 00	-0.2046E 00
0.390000	-0.9381E 00	-0.9446E 00	-0.2156E 00	-0.2159E 00
0.400000	-0.3334E 00	-0.3389E 00	-0.2205E 00	-0.2208E 00
0.410000	0.2354E 00	0.2306E 00	-0.2194E 00	-0.2196E 00
0.420000	0.7561E 00	0.7520E 00	-0.2127E 00	-0.2128E 00
0.430000	0.1219E 01	0.1215E 01	-0.2010E 00	-0.2011E 00
0.440000	0.1617E 01	0.1613E 01	-0.1850E 00	-0.1851E 00
0.450000	0.1943E 01	0.1940E 01	-0.1654E 00	-0.1655E 00
0.460000	0.2195E 01	0.2192E 01	-0.1430E 00	-0.1430E 00
0.470000	0.2373E 01	0.2370E 01	-0.1186E 00	-0.1186E 00
0.480000	0.2476E 01	0.2473E 01	-0.9295E-01	-0.9291E-01
0.490000	0.2508E 01	0.2505E 01	-0.6679E-01	-0.6673E-01
0.500000	0.2473E 01	0.2470E 01	-0.4085E-01	-0.4077E-01
0.510000	0.2377E 01	0.2374E 01	-0.1578E-01	-0.1568E-01
0.520000	0.2227E 01	0.2224E 01	0.7836E-02	0.7955E-02
0.530000	0.2031E 01	0.2027E 01	0.2950E-01	0.2963E-01
0.540000	0.1797E 01	0.1793E 01	0.4879E-01	0.4894E-01
0.550000	0.1535E 01	0.1531E 01	0.6540E-01	0.6556E-01
0.560000	0.1252E 01	0.1248E 01	0.7908E-01	0.7925E-01
0.570000	0.9583E 00	0.9539E 00	0.8971E-01	0.8990E-01
0.580000	0.6619E 00	0.6572E 00	0.9724E-01	0.9743E-01
0.589999	0.3704E 00	0.3655E 00	0.1017E 00	0.1019E 00
0.599999	0.9120E-01	0.8596E-01	0.1032E 00	0.1034E 00
0.609999	-0.1695E 00	-0.1751E 00	0.1019E 00	0.1022E 00
0.619999	-0.4065E 00	-0.4122E 00	0.9814E-01	0.9835E-01
0.629999	-0.6152E 00	-0.6212E 00	0.9209E-01	0.9230E-01
0.639999	-0.7925E 00	-0.7987E 00	0.8411E-01	0.8432E-01
0.649999	-0.9361E 00	-0.9425E 00	0.7456E-01	0.7475E-01
0.659999	-0.1045E 01	-0.1051E 01	0.6379E-01	0.6397E-01

0.669999	-0.1118E 01	-0.1125E 01	0.5218E-01	0.5234E-01
0.679999	-0.1158E 01	-0.1164E 01	0.4008E-01	0.4023E-01
0.689999	-0.1164E 01	-0.1171E 01	0.2785E-01	0.2797E-01
0.699999	-0.1140E 01	-0.1146E 01	0.1580E-01	0.1591E-01
0.709999	-0.1088E 01	-0.1094E 01	0.4244E-02	0.4324E-02
0.719999	-0.1012E 01	-0.1018E 01	-0.6565E-02	-0.6512E-02
0.729999	-0.9152E 00	-0.9211E 00	-0.1640E-01	-0.1638E-01
0.739999	-0.8020E 00	-0.8077E 00	-0.2509E-01	-0.2509E-01
0.749999	-0.6766E 00	-0.6818E 00	-0.3248E-01	-0.3251E-01
0.759999	-0.5429E 00	-0.5477E 00	-0.3849E-01	-0.3855E-01
0.769999	-0.4051E 00	-0.4095E 00	-0.4306E-01	-0.4314E-01
0.779999	-0.2670E 00	-0.2709E 00	-0.4618E-01	-0.4628E-01
0.789999	-0.1322E 00	-0.1356E 00	-0.4787E-01	-0.4800E-01
0.799999	-0.4037E-02	-0.6846E-02	-0.4821E-01	-0.4835E-01
0.809999	0.1148E 00	0.1125E 00	-0.4728E-01	-0.4744E-01
0.819999	0.2219E 00	0.2202E 00	-0.4519E-01	-0.4536E-01
0.829999	0.3154E 00	0.3142E 00	-0.4210E-01	-0.4228E-01
0.839999	0.3938E 00	0.3931E 00	-0.3815E-01	-0.3833E-01
0.849999	0.4563E 00	0.4561E 00	-0.3351E-01	-0.3369E-01
0.859999	0.5024E 00	0.5027E 00	-0.2835E-01	-0.2852E-01
0.869999	0.5322E 00	0.5328E 00	-0.2284E-01	-0.2300E-01
0.879999	0.5461E 00	0.5470E 00	-0.1715E-01	-0.1730E-01
0.889999	0.5448E 00	0.5461E 00	-0.1144E-01	-0.1158E-01
0.899999	0.5297E 00	0.5311E 00	-0.5863E-02	-0.5977E-02
0.909999	0.5020E 00	0.5036E 00	-0.5458E-03	-0.6385E-03
0.919999	0.4635E 00	0.4652E 00	0.4391E-02	0.4321E-02
0.929999	0.4159E 00	0.4176E 00	0.8849E-02	0.8801E-02
0.939999	0.3611E 00	0.3628E 00	0.1275E-01	0.1272E-01
0.949999	0.3011E 00	0.3027E 00	0.1603E-01	0.1603E-01
0.959999	0.2378E 00	0.2392E 00	0.1865E-01	0.1867E-01
0.969999	0.1731E 00	0.1743E 00	0.2060E-01	0.2064E-01
0.979999	0.1087E 00	0.1097E 00	0.2187E-01	0.2193E-01
0.989999	0.4634E-01	0.4710E-01	0.2248E-01	0.2256E-01
0.999999	-0.1258E-01	-0.1210E-01	0.2247E-01	0.2257E-01
1.009998	-0.6679E-01	-0.6661E-01	0.2188E-01	0.2199E-01

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFOSR-TR- 80-0258	2. GOVT ACCESSION NO.: AD-A083 577
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): A QUANTITATIVE APPROACH TO AGGREGATION IN THE MODELING OF TACTICAL COMMAND, CONTROL, AND COMMUNICATION SYSTEMS	5. TYPE OF REPORT & PERIOD COVERED: Final
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Henry D'Angelo	8. CONTRACT OR GRANT NUMBER(s): AFOSR 79-0101
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Boston University, College of Engineering, Boston, MA 02215	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 61102F 2304/A1
11. CONTROLLING OFFICE NAME AND ADDRESS: Air Force Office of Scientific Research/NM, Bolling AFB, Washington, DC 20332	12. REPORT DATE: March 1980
	13. NUMBER OF PAGES: 46
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Modeling, Simulation, Aggregation, Large-scale Systems, Model in Simplification, Dynamic Systems, Structural Sensitivity, System Graphs

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This research effort is directed at determining the feasibility of using structural-sensitivity measures as the basis for simplifying large mathematical models of dynamic systems (e.g., models of tactical command, control, and communication systems). Toward this end, a FORTRAN program was written which can be used to simplify models characterized by sets of linear differential equations. Specifically, the program determines the optimal simplified model (i.e., the set of coefficients characterizing the set of linear differential equations) of specified dimension corresponding to the original linear model of higher dimension. OVER

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED (OVER)

20. Abstract cont.

The algorithm, on which the FORTRAN program is based, minimizes an objective function which is defined in terms of the structural sensitivities of the state variables to be preserved in the simplified model. The program inputs are the set of coefficients which define (1) the linear differential equations representing the original model, (2) the dimension of the simplified model to be determined, and (3) the set of parameters defining the objective function to be minimized. The program outputs are the set of coefficients which define the optimal simplified model and the value of the objective function corresponding to the optimal simplified model.