NASA TECHNICAL NOTE

NASA TN D-6467

NASA TN D-6467

NONLINEAR SYSTEMS IDENTIFICATION IN PRESENCE OF NONUNIQUENESS

by Jean-Noël Aubrun

Ames Research Center
Moffett Field, Calif. 94035

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • AUGUST 1971

1. Report No.
NASA TN D-6467

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle

NONLINEAR SYSTEMS IDENTIFICATION IN PRESENCE OF NONUNIQUENESS

5. Report Date
August 1971

6. Performing Organization Code

7. Author(s)

Jean-Noël Aubrun

8. Performing Organization Report No.
A-3807

9. Performing Organization Name and Address

NASA Ames Research Center
Moffett Field, Calif., 94035

10. Work Unit No.
125-19-03-11-00-21

11. Contract or Grant No.

12. Sponsoring Agency Name and Address

National Aeronautics and Space Administration
Washington, D. C., 20546

13. Type of Report and Period Covered
Technical Note

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

This report considers the problem of identifying a set of parameters that will match the input-output of a mathematical model to that of a physical system. Except for particular cases, there has been no practical method to determine if such parameter values are unique. A general process of parameter identification is described that addresses this problem and uses geometrical concepts in terms of which the nonuniqueness problem can easily be defined.

A digital computer algorithm is developed that analyzes the structure of a space defined by the parameter sensitivity functions and the output data set. The algorithm deduces an optimal set of parameters to be uniquely identified, determines the relationships between dependent parameters, and specifies which parameters can be obtained with *a priori* knowledge of others. It does not require canonical or linear equations for the model but maintains the physical identity of the parameters. A corresponding FORTRAN IV program has been written.

This technique is illustrated by the identification of the AMES three-degrees-of-freedom satellite simulator. Examples of nonuniqueness were found and analyzed successfully by the algorithm, demonstrating its ability to cope with strongly nonlinear cases.

17. Key Words (Suggested by Author(s))

nonuniqueness

18. Distribution Statement

Unclassified – Unlimited

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of Pages
82

22. Price*
$3.00

*For sale by the National Technical Information Service, Springfield, Virginia 22151

TABLE OF CONTENTS

SYMBOLS

A	parameter vector (N components $a_1, a_2, \ldots, a_N$)
a_i	*ith* parameter
B_K	set of K vectors
D	matrix of column vectors D_i ($M \times N$)
DER	matrix of the column vectors DER_i ($M \times N$)
DER_i	partial derivative of the vector $\hat{Y}$ with respect to the parameter a_i ($\partial\hat{Y}/\partial a_i$)
DET_K	value of the Gram determinant of the first K vectors D_i
D_i	unit vector colinear to $\partial\hat{Y}/\partial a_i$
E_M	output space
E_N	linear manifold defined by the N vectors DER_i (locally tangent to Σ_N)
ER	error vector (M components); $\hat{Y} - Y$
ER_o	projection of ER upon Σ_N
$F(X,U)$	vector function of X and U
G	Gram matrix of the vectors D_i ($N \times N$)
g_{ij}	element of the matrix G
iff	if and only if
i,j	running indices
K	number of vectors in a subset (or subbasis) B_K
M	total number of measurements
MN(i)	dependency index associated with the parameter a_i (FORTRAN variable)
N	number of parameters
n_i	index of the *ith* vector of the optimal basis and of the corresponding parameter

P	number of outputs
s_{K+1}	separation between the vector D_{K+1} and the set B_K
s_O	separation threshold
$t_1, t_2, \ldots$	time instants
U	input vector (forcing function)
V_i	volume of the hyperparallelepiped spanned by i vectors
X	state vector
Y	measured output vector (M components $y(t_1)$, $y(t_2)$, . . ., $y(t_M)$)
Y_O	projection of Y upon Σ_N
y(or y(t))	measured time history of an output
ε	angular threshold
Σ_N	model parameter subspace (hypersurface in output space E_M)
$(\dot{\ })$	time derivative of ( )
$(\hat{\ })$	estimated value of ( )
$(\ )^T$	transposed of the matrix ( )
$(\ )^{-1}$	inverse of the matrix ( )

Symbols Used in the Illustrative Example

b	bearing offset × mass M of the platform (three components b_x, b_y, b_z)
C	control matrix (diagonal 3 × 3 of elements C_1, C_2, C_3)
E	voltage applied to the DC motor
g	gravity vector (3 components)
h	body angular momentum vector (3 components)
h_W	total angular momentum of the reaction wheels (3 components h_{W1}, h_{W2}, h_{W3})

I	current in the DC motor
J	inertia tensor (inertias, Jx, Jy, Jz; products, Jxy, Jxz, Jyz)
J_W	inertia tensor of the wheel-rotor combination (diagonal)
K_B	back electromotive force constant
K_D	damping constant
K_T	torque constant of the DC motor
k_1,k_2,k_3	DC gains of the filters
R	resistance of the wiring
S( )	skew matrix defining a cross product
T	total torque applied to the body
T_G	gravity torque
T_W	total torque due to the reaction wheels
α	feedback matrix (diagonal 3 × 3 of elements α_1, α_2, α_3)
τ	time constant of the first order filters
ω	body angular velocity vector (3 components p, q, and r)
Ω	angular velocity of the wheel

Symbols Used in the Appendixes

c_j	j*th* component of the vector D^T ER
DET_i	product of the i first pivots
D_i	unit vector parallel to $\partial\hat{Y}/\partial a_i$
da_K	differential of a_K or error bound on the parameter a_K
E( )	expectation value of ( )
e_i	unit vector orthogonal to some space
LC/s_o	linearly close with respect to the threshold s_o
p_i	value of the i*th* pivotal element
S(D/B)	separation of some vector D from some set B

s_i	separation of some vector from some set
V	some vector of E_{K-1}
$V(B)$	volume of the hyperparallelepiped spanned by the vectors of a set B
Z	function of z_i
$z_1, z_2, \ldots$	coordinates in the space $\sum_{K-1}$
δY	error in Y
δY_o	error in Y_o
ε_K	part of $\partial\hat{Y}/\partial a_K$ orthogonal to E_{K-1}
ε_o^2	square of the error bound in Y_o
λ_i	component of the vector $\partial\hat{Y}/\partial a_K$ on the vector $\partial\hat{Y}/\partial a_i$
μ_i	component of the vector D_K on the vector D_i in a particular set
μ_i'	component of the vector D_K on the vector D_i in another particular set
$\cup$	set union
$\subset$	set inclusion
$\in$	set membership
$\forall$	universal quantifier (for any)
$\exists$	existential quantifier (there exists)

NONLINEAR SYSTEMS IDENTIFICATION IN PRESENCE OF NONUNIQUENESS

Jean-Noël Aubrun*

Ames Research Center

SUMMARY

This report considers the problem of identifying a set of parameters that will match the input-output of a mathematical model to that of a physical system. Except for particular cases, there has been no practical method to determine if such parameter values are unique. A general process of parameter identification is described that addresses this problem and uses geometrical concepts in terms of which the nonuniqueness problem can easily be defined.

A digital computer algorithm is developed that analyzes the structure of a space defined by the parameter sensitivity functions and the output data set. The algorithm deduces an optimal set of parameters to be uniquely identified, determines the relationships between dependent parameters, and specifies which parameters can be obtained with *a priori* knowledge of others. It does not require canonical or linear equations for the model but maintains the physical identity of the parameters. A corresponding FORTRAN IV program has been written.

This technique is illustrated by the identification of the Ames three-degrees-of-freedom satellite simulator. Examples of nonuniqueness were found and analyzed successfully by the algorithm, demonstrating its ability to cope with strongly nonlinear cases.

INTRODUCTION

The relationship between the input and the output of a physical system or plant may be described in most situations by a mathematical relationship or model. This paper is concerned with models that depend on a finite set of parameters and with the identification of these parameters from measurements of the plant input and output.

General theories of parameter identification have been established (e.g., refs. 1 and 2), and specific applications have been made (e.g., refs. 3-7) primarily to linear systems for which a thorough mathematical analysis can be performed. For this discussion, a distinction should be made between two types of parameter identification. In the first type, the goal is to obtain a mathematical representation of the plant from which it is possible to duplicate input-output sets of measurements regardless of the actual physics or dynamics involved in the plant. This is the concern of realization theory. The algorithm of Ho (ref. 7) and later developments introduced by

*National Research Council Postdoctoral Associate in residence at Ames Research Center.

Kalman (ref. 4) are typical of this approach. In the second type the form of the model is obtained first from an analysis of the physical processes occurring in the plant. The parameters in this case have a direct physical meaning and the goal is to obtain their actual values from input-output measurements. If these values are known exactly, plant and model will exhibit the same input-output properties. However, the converse is not true although often implicitly assumed (i.e., the parameter values that match model and plant input-output may not be unique).

This report is mainly concerned with the second type of parameter identification where it is essential to recognize the occurrence of nonunique solutions for the parameter values. This nonuniqueness may come from parameter redundancy or insufficient information in the measurements or both. While it is possible in linear systems to transform the original parameters into a uniquely identifiable reduced set (canonical parameters) (ref. 8), there is no general canonical solution for nonlinear systems. Concerning the second cause of nonuniqueness, theoretical criteria are available from estimation theory for linear systems, but because they involve idealized arithmetic operations (such as identically vanishing determinants), they may fail completely when numerical computation or noisy measurements are used.

To solve the nonuniqueness problem, this report first presents a general procedure of parameter identification in a more direct and practical way than has been done previously. This procedure is used as a framework for analyzing the mechanics of parameter identification on a practical level and allows a formulation of the nonuniqueness problem that takes into account the uncertainties in real measurements and computations. These uncertainties are expressed in the relaxation of strict mathematical concepts to the benefit of approximate ones. It is realized, for instance, that there is no such thing as a singular matrix with a digital computer and the classical "linear dependence" is replaced by a concept of "linear closeness."

A digital computer algorithm is developed that:

1. Detects the existence of nonuniqueness.

2. Determines which sets of parameters are not uniquely defined and which are independent.

3. Determines which parameters should have values specified to obtain a correct answer for the others.

4. Optionally, resets the main identification program with a reduced number of parameters that can be uniquely identified, the others being left at a constant value.

This computation technique is applied to data recorded from the Ames three-degrees-of-freedom satellite simulator. Identification of the inertia tensor and some linear and nonlinear control elements is performed to provide a typical illustration of the nonuniqueness problem.

BASIC PARAMETER IDENTIFICATION PROCEDURE

The Plant

The direct measurements of a physical system (plant) may be considered as a set of quantities that are recorded at different instants of time. It is convenient here to partition the set into input and output sets. For dynamic systems, the input is usually defined as some forcing functions that modify the state of the system. This restriction is unnecessary for the present approach to the identification problem. Here it will be required only that the plant be some kind of operator connecting input and output. In the case of an airplane, for instance, longitudinal stick position could be the input and pitch rate the output. However, depending upon the type of identification to be performed, one could also use longitudinal stick position and pitch rate as input, and pitch acceleration as output.

The Model

To define the plant completely one must know all the output sets that correspond to all the possible input sets. However, this is not very practical. Instead one tries to determine what is called a "model," that is some process by which, for any given set of input values, one is able to obtain the corresponding set of output values. It will never be possible to do so exactly, first because of the actual complexity of the plant and because a finite number of digits have to be used in computation. Hence the model is only an approximation that describes the plant to the extent that the sets of input and output values of the model are close in some sense (such as least squares) to those of the plant. The approximation, however, will focus on the important properties of the plant, so that its behavior can be predicted or modified in some desired way.

The model in itself depends upon quantities called the parameters. In this discussion we will consider only models whose parameters are invariant with respect to time and of finite number. For instance the mathematical model of an aircraft may be a system of differential equations describing its dynamics. The coefficients in these equations, such as stability derivatives, inertias, and control gains along with the initial conditions, are normally considered the parameters.

The Identification

The objective of the identification process is to determine, for a given model form, the values of the parameters that will make the model behave like the plant. A general process of identification is sketched on figure 1. If the same input is applied to both model and plant, the best values for the parameters are those that minimize (with respect to some criterion) the difference between model and plant outputs. Therefore the identification algorithm has to compare these outputs and adjust the parameter values until this minimum is reached.

It is usual, in parameter identification terminology, to distinguish between "equation error" and "output error" methods (ref. 6). It should be pointed out that both of them can be cast into the scheme of figure 1, provided a judicious choice of input and output is made. Let us consider for instance a plant defined by the system of differential equations:

$$\dot{X} = F(X, U) \tag{1}$$

where X is the state vector and U the control. An output error method will consider U as the input to a model defined by

$$\dot{\hat{X}} = \hat{F}(\hat{X}, U)$$

and compare the output $\hat{X}$ of the model to the measured output X of the plant to obtain the parameters in $\hat{F}$. In an equation error method, U and X can be considered as inputs to a model defined by

$$\dot{\hat{X}} = \hat{F}(X, U)$$

while $\dot{\hat{X}}$ is now the model output to be compared to the plant output $\dot{X}$.

In the first case the model is differential, in the second it is algebraic, but in both the only information concerning the parameters is to be found in the comparison between measured and computed outputs.

Once the form of the mathematical model is specified, two important questions must be asked:

1. If it is possible to match some particular set of input-output values of the system to that of the model, using certain values for the parameters, and is it possible to match every other set with the same parameter values?

2. Is it possible to match the same input-output set with more than one set of values for the parameters?

The first question relates to the problem of modeling; the second expresses the problem of uniqueness. A general way to perform the identification must be studied to understand where the uniqueness problem comes from.

Consider first the case of a single output, say y. Measurements of the time history y(t) have been made, corresponding to a known input. In this analysis, digital computation will be considered; hence the output set consists of M discrete values of y corresponding to the instants t_1, t_2, . . ., t_M, as represented on the top curve of figure 2(a), dotted at the measurement points. The model depends upon the N parameters of unknown values, a_1, a_2, . . ., a_N. To begin with, these parameters are given some assumed values, purely arbitrary or based upon a best guess. Since the input is known, a computed time history $\hat{y}(t)$ may be obtained from the model, particularly the M discrete values corresponding to the same instants t_1, t_2, . . ., t_M. In general, the two time histories of y and $\hat{y}$ are different as shown also on figure 2(a), and the problem is to find how the parameters must be adjusted to make these time histories as similar as possible.

An intuitive idea is to change the value of one parameter, say a_1, by a small amount da_1, and observe what happens to the time history of $\hat{y}$. A slightly different time history $\hat{y} + d\hat{y}_1$ is obtained (fig. 2(a)), and eventually some idea of the effect of the parameter a_1 and how much it should be changed could be gained. The same can be done with a_2, obtaining another time history with different characteristics, and so on with the N parameters of the model. By this trial and error procedure, one might be successful in matching the two time histories. This technique, called analog matching, has been fairly successful in the past. It is obvious, however, that it becomes impractical if the parameters are numerous, although the human mind takes definite advantage of its pattern recognition ability.

This process may be rationalized by considering the M values of y as the components of a vector Y, and the M corresponding values of $\hat{y}$, as the components of another vector, $\hat{Y}$ (fig. 2(b)). Both Y and $\hat{Y}$ belong to an M-dimensional space and there is a one-to-one correspondence between a time history and its representative vector. To say that the time histories are different is to say that the vector Y and $\hat{Y}$ are different. Their difference may be expressed directly by the geometric difference $\hat{Y} - Y$, and we define therefore an M component error vector as

$$ER = \hat{Y} - Y$$

It follows that the discrete time histories will match *iff* ER is zero. Consider now what happens when the value of a_1 is changed. The change in the time history noted previously has now a very precise meaning, which is the change $d\hat{Y}_1$ in the vector $\hat{Y}$ that becomes equal to

$$\hat{Y}_1 = \hat{Y} + d\hat{Y}_1$$

This M component vector $d\hat{Y}_1$ describes unambiguously how the time history has been modified by the increment da_1 of the parameter a_1. If da_1 is small enough, one may write

$$d\hat{Y}_1 = \left(\frac{\partial\hat{Y}}{\partial a_1}\right) da_1$$

and all the local information concerning the effect of a_1 upon the time history is contained in the vector $\partial\hat{Y}/\partial a_1$. Therefore, the first step in the identification will be to perturb each parameter a_i individually and compute the corresponding derivatives (M components vectors) $\partial\hat{Y}/\partial a_i$ by

$$\frac{\partial\hat{Y}}{\partial a_i} = \frac{(\hat{Y}_i - \hat{Y})}{da_i}$$

It has just been demonstrated that if the values of the parameters are changed, it is possible to change the vector $\hat{Y}$ in this M-dimensional space. It would be desirable to change it in such a way that it will coincide with Y (fig. 3). If *simultaneous* increments $da_1, da_2, \ldots, da_N$ are given to the corresponding parameters, $\hat{Y}$ will change by an amount

$$d\hat{Y} = \sum_i \left(\frac{\partial \hat{Y}}{\partial a_i}\right) da_i \qquad i = 1, N \tag{2}$$

This relation may be expressed conveniently in matrix form by considering each parameter as one component of an N-dimensional vector A, and each vector $\partial\hat{Y}/\partial a_i$ as a column of an M × N matrix DER. Then equation (2) is equivalent to

$$d\hat{Y} = \text{DER}\ dA \tag{3}$$

with

$$\text{DER} = \begin{pmatrix} \partial\hat{y}(t_1)/\partial a_1 & \partial\hat{y}(t_1)/\partial a_2 & \cdots & \partial\hat{y}(t_1)/\partial a_N \\ \partial\hat{y}(t_2)/\partial a_1 & \partial\hat{y}(t_2)/\partial a_2 & \cdots & \partial\hat{y}(t_2)/\partial a_N \\ \vdots & & & \\ \partial\hat{y}(t_M)/\partial a_1 & \partial\hat{y}(t_M)/\partial a_2 & \cdots & \partial\hat{y}(t_M)/\partial a_N \end{pmatrix}$$

Finally a dA is needed such that the corresponding change in $\hat{Y}$ *just* cancels the error ER, that is, such that

$$ER + DER\ dA = 0 \tag{4}$$

The left side represents the error after the parameter vector has been changed by dA, resulting in the change $d\hat{Y}$ in Y as shown on figure 3. Usually there are more measurements than parameters (M >> N) so that equation (4) cannot be solved directly. However, a least squares solution may be obtained by minimizing the quantity $(\text{ER} + \text{DER}\ dA)^2$ with respect to dA. A well-known expression for the solution is given by

$$dA = -(\text{DER}^T\ \text{DER})^{-1}\ \text{DER}^T\ \text{ER} \tag{5}$$

This solution is valid, of course, provided the inverse of the square N × N matrix (DER^T DER) exists.

This type of equation is found in many identification techniques (refs. 1, 2, 3, 6). It represents a pseudoinversion of DER and has an interesting geometrical meaning. As has been said, Y and $\hat{Y}$ belong to an M-dimensional space (fig. 4) and it is possible to "move" $\hat{Y}$ in this space by changing the value of the parameters. But, since there are only N parameters (N < M), they define *at most* N independent directions; that is, $\hat{Y}$ is, in fact, constrained to stay in an N-dimensional subspace Σ_N of the M-dimensional space. If there were no noise in the measurements, if the model were perfect, and if it were possible to compute everything exactly, the vector Y would also belong to this subspace Σ_N and $\hat{Y}$ and Y could be made to coincide exactly, cancelling completely the error ER and therefore obtaining a perfect match of the time histories. This situation corresponds algebraically to compatible equations in the system expressed by equation (4). Unfortunately the conditions expressed above are never met in practice, so Y is always outside this subspace Σ_N and cannot be reached by $\hat{Y}$. Therefore the best tack is to reach the orthogonal projection Y_o of Y upon Σ_N. The distance $[(\hat{Y} - Y)^2]^{1/2}$ will then be minimal so that this can be interpreted as a least squares solution. Equation (5) expresses nothing but that the real "target" of $\hat{Y}$ is not Y but Y_o. Indeed from equations (3) and (5)

$$d\hat{Y} = \mathrm{DER}\ dA = -\mathrm{DER}(\mathrm{DER}^T\ \mathrm{DER})^{-1}\ \mathrm{DER}^T\ \mathrm{ER} = -\mathrm{ER}_o$$

It can be shown that ER_o is just the orthogonal projection of ER on Σ_N by verifying that $(\mathrm{ER} - \mathrm{ER}_o)$ is orthogonal to all the vectors $\partial\hat{Y}/\partial A_i$ which constitute a local basis in Σ_N; that is, since these vectors are contained in the matrix DER,

$$\mathrm{DER}^T(\mathrm{ER} - \mathrm{ER}_o) = \mathrm{DER}^T\ \mathrm{ER} - \mathrm{DER}^T\ \mathrm{DER}(\mathrm{DER}^T\ \mathrm{DER})^{-1}\ \mathrm{DER}^T\ \mathrm{ER} \equiv 0$$

It must be emphasized, however, that equation (5) is not the only way to operate on $\hat{Y}$. There are many different techniques for obtaining a least squares solution of equation (4) and they may not all require an actual matrix inversion as in equation (5). But it is very important to note that the projection property is independent of any technique;[1] whatever the method, $\hat{Y}$ will always be constrained in some N-dimensional subspace Σ_N and the parameters adjusted in such a way that $\hat{Y}$ can reach some target Y_o in Σ_N. Also independent of the identification technique itself is the fact that any change $d\hat{Y}$ in the output of the model has to be related to the change dA in the parameter vector by

$$d\hat{Y} = \mathrm{DER}\ dA$$

This expression defines the local properties of the space Σ_N. The extremity of $\hat{Y}$ is therefore constrained on an N-dimensional hypersurface Σ_N. The local hyperplane E_N tangent to Σ_N is spanned by the column vectors of DER (i.e., the N vectors $\partial\hat{Y}/\partial A_i$). When DER is constant with respect to the parameters, Σ_N is also a hyperplane and coincides with E_N everywhere. When it is not, Σ_N is a curved hypersurface, and equation (5) will not move $\hat{Y}$ toward Y_o but toward the projection $\hat{Y}_o$ of Y on E_N. Because $\hat{Y}_o$ is only an approximation for Y_o, equation (5) must be iterated. The evolution of $\hat{Y}$, $\hat{Y}_o$, and E_N during this iterative process is represented schematically in figure 5; $\hat{Y}$ has to move along the curve Σ_N and the tangent to Σ_N at $\hat{Y}$ is E_N. At the beginning $\hat{Y}$ is $\hat{Y}(0)$. Equation (5) approximates Y_o by the projection $\hat{Y}_o(1)$ of Y upon $E_N(0)$. A new value $\hat{Y}(1)$ is obtained for $\hat{Y}$ at the first iteration and the arc $(\hat{Y}(0)\hat{Y}(1))$ on Σ_N is approximately equal to the distance $\hat{Y}(0)\hat{Y}_o(1)$ on $E_N(0)$ if $\hat{Y}(0)$ were close to Y_o. Starting now with $\hat{Y}(1)$, a better approximation of Y_o is obtained, $\hat{Y}_o(2)$. Finally, when $\hat{Y}(i)$ is close enough to Y_o, then $E_N(i)$ and Σ_N are equivalent to

[1] When weights are used in the minimization of the error (as in maximum likelihood estimation or in general weighted least squares methods), these results remain valid provided the output vector is redefined as, say Y', such that $Y' = WY$ where W is a weighting matrix.

compute the projection of Y, and, in general, any linear property of Σ_N will also be found in E_N.

Once Y_o is reached, that is, when the time histories are matched as well as possible (i.e., when ER^2 is minimum, ER^2 being the cost of the identification since it is a measure of how good the match is), one must ask if it is possible to change the parameters in such a way that the output $\hat{Y}$ does not change. If the answer is yes, then it is possible to find another set of values for the parameters that will match the time histories as well; hence the parameters cannot be uniquely identified. Since Σ_N contains all the properties of the model, it is clear that this nonuniqueness has to be related to the properties and the structure of Σ_N, or to those of E_N if only linear properties are involved.

Before this point is developed, a last word should be said for the multioutput case. If the system has P outputs, $y_1, y_2, \ldots, y_P$, each composed of M_o measurements, then the vector Y is constructed with these $M_o \times P$ quantities. The order in which these measurements are taken to form the components of Y is of no importance provided the same order is used for the model output $\hat{Y}$. An example of the vectors and matrices involved is given in expression (6) where Y and $\hat{Y}$ are M-dimensional with $M = M_o \times P$ and $(\hat{Y} - Y)$ will be zero *iff* the P time histories of the model are equal to the corresponding time histories of the system. With this definition of Y and $\hat{Y}$, all the previous discussions and results are valid.

$$
\begin{array}{ccc}
\begin{array}{c}\text{Plant}\\ \text{output}\\ Y(M)\end{array} & \begin{array}{c}\text{Model}\\ \text{output}\\ \hat{Y}(M)\end{array} & \begin{array}{c}\text{Matrix of the derivatives of the model output}\\ \\ \text{DER } (M \times N)\end{array} \\
\left|\begin{array}{c}
y_1(t_1)\\ y_1(t_2)\\ \vdots\\ y_1(t_{M_o})\\ y_2(t_1)\\ y_2(t_2)\\ \vdots\\ y_2(t_{M_o})\\ \vdots\\ y_P(t_1)\\ y_P(t_2)\\ \vdots\\ y_P(t_{M_o})
\end{array}\right|
&
\left|\begin{array}{c}
\hat{y}_1(t_1)\\ \hat{y}_1(t_2)\\ \vdots\\ \hat{y}_1(t_{M_o})\\ \hat{y}_2(t_1)\\ \hat{y}_2(t_2)\\ \vdots\\ \hat{y}_2(t_{M_o})\\ \vdots\\ \hat{y}_P(t_1)\\ \hat{y}_P(t_2)\\ \vdots\\ \hat{y}_P(t_{M_o})
\end{array}\right|
&
\left|\begin{array}{cccc}
\partial\hat{y}_1(t_1)/\partial a_1 & \partial\hat{y}_1(t_1)/\partial a_2 & \cdots & \partial\hat{y}_1(t_1)/\partial a_N\\
\partial\hat{y}_1(t_2)/\partial a_1 & \partial\hat{y}_1(t_2)/\partial a_2 & \cdots & \partial\hat{y}_1(t_2)/\partial a_N\\
\vdots & \vdots & & \vdots\\
\partial\hat{y}_1(t_{M_o})/\partial a_1 & \partial\hat{y}_1(t_{M_o})/\partial a_2 & \cdots & \partial\hat{y}_1(t_{M_o})/\partial a_N\\
\partial\hat{y}_2(t_1)/\partial a_1 & \partial\hat{y}_2(t_1)/\partial a_2 & \cdots & \partial\hat{y}_2(t_1)/\partial a_N\\
\partial\hat{y}_2(t_2)/\partial a_1 & \partial\hat{y}_2(t_2)/\partial a_2 & \cdots & \partial\hat{y}_2(t_2)/\partial a_N\\
\vdots & \vdots & & \vdots\\
\partial\hat{y}_2(t_{M_o})/\partial a_1 & \partial\hat{y}_2(t_{M_o})/\partial a_2 & \cdots & \partial\hat{y}_2(t_{M_o})/\partial a_N\\
\vdots & \vdots & & \vdots\\
\partial\hat{y}_P(t_1)/\partial a_1 & \partial\hat{y}_P(t_1)/\partial a_2 & \cdots & \partial\hat{y}_P(t_1)/\partial a_N\\
\partial\hat{y}_P(t_2)/\partial a_1 & \partial\hat{y}_P(t_2)/\partial a_2 & \cdots & \partial\hat{y}_P(t_2)/\partial a_N\\
\vdots & \vdots & & \vdots\\
\partial\hat{y}_P(t_{M_o})/\partial a_1 & \partial\hat{y}_P(t_{M_o})/\partial a_2 & \cdots & \partial\hat{y}_P(t_{M_o})/\partial a_N
\end{array}\right|
\end{array}
\qquad (6)
$$

THE NONUNIQUENESS PROBLEM

To avoid any confusion with other possible theoretical definitions, we have to make clear that the nonuniqueness problem considered here is a practical one. The only available information is a set of input-output values of the plant from which the parameters of the model have to be identified. This could be obtained, for instance, during a flight test and the corresponding piece of data is of *finite* length. It might happen that the identification using this particular data has nonuniqueness problems, while another piece of data taken with the same airplane leads to unique values. Therefore the nonuniqueness is relative to a given set of data, although it might also be an intrinsic property of the model, in which case it will always be found.

When discussing the identification process it was said that the N parameters of the model defined locally an N-dimensional subspace E_N, the basis vectors of this subspace being the M-component vectors $\partial\hat{Y}/\partial a_i$. In fact this is true only if the basis vectors are linearly independent. Otherwise E_N is not truly N-dimensional and the fact that one vector of the basis can be expressed as a linear combination of some others means that the effect of one parameter change could be obtained also by a simultaneous change in some other parameters. Obviously there will not be a unique solution in this case. If equation (5) is used, the matrix (DER^T DER) is singular and the computation might stop at this time. Unfortunately, the computer will very likely invert it anyway, because roundoff errors make it generally impossible to obtain a hard zero for the determinant. The solution obtained in this case might very well match the time histories because of the adaptative nature of the algorithm, as we have actually observed on simulated data. If other techniques are used that do not involve an actual inversion (as in the steepest descent method), then a solution is reached anyway. Here again the nonuniqueness will be unnoticed and wrong conclusions will be drawn if one observes only the fit in the outputs. One should be careful not to judge things only for their outer properties, but to understand also the inner aspect (i.e., in this case, the real structure of the space E_N).

Consider a two-dimensional example to begin with, corresponding to a model with two parameters a_1 and a_2. Here, E_N is E_2 (i.e., a plane) and is shown on figure 6(a) with the basis vectors $\partial\hat{Y}/\partial a_1$ and $\partial\hat{Y}/\partial a_2$. The "target" is Y_o, the projection on E_2 of some M-dimensional vector Y. The amounts da_1 and da_2 by which the parameters have to be changed are equal to the components of $Y_o - \hat{Y}$ on the basis, because by definition of the components

$$Y_o - \hat{Y} = \left(\frac{\partial\hat{Y}}{\partial a_1}\right) da_1 + \left(\frac{\partial\hat{Y}}{\partial a_2}\right) da_2 \qquad (7)$$

and this is also equal to the change in $\hat{Y}$ according to equation (2). Consider now the case of figure 6(b) where the basis vectors are parallel. Here Y_o may be reached by changing either a_1 or a_2, or both. The vectorial equation (7) collapses in one algebraic equation that of course has an infinity of solutions in da_1 and da_2. The model does not really depend separately on a_1 and a_2 but on some function of them (see appendix A). However, if the true value of, say, a_1 is known (from other sets of measurements for instance), it is possible to continue the identification process and obtain the true value of a_2, a_1 being removed from the perturbation process and left at its true value. These basic ideas will be extended to the N-dimensional case, but we have been dealing so far with a strict mathematical definition of linear dependence and independence. Here there is a definite boundary between the cases of figures 6(a) and 6(b).

In the real world this yes-or-no situation does not exist and intermediate states must be considered. Because of noise in measurements, the actual output of the plant is not Y_o but may be somewhere in a domain surrounding Y_o. Assuming some kind of boundaries for this uncertainty domain, as represented on figure 7 by the shadowed area, there are corresponding uncertainties in the vectors $d\hat{Y}_1$ and $d\hat{Y}_2$ (shadowed parts of the axis in figure 7) that in turn correspond to uncertainties in da_1 and da_2. If the angle between the two vectors $\partial\hat{Y}/\partial a_1$ and $\partial\hat{Y}/\partial a_2$ is decreased, these uncertainties increase. If they become, say, one order of magnitude larger than the parameters themselves, for all practical purpose these parameters are not uniquely defined. However, the values obtained for these parameters are not independent of each other, since they must satisfy the constraint that $d\hat{Y}_1 + d\hat{Y}_2$ falls inside the uncertainty domain of Y_o. Therefore, if the angle between the two basis vectors is less than some threshold ε, nonuniqueness will be experienced almost as if this angle were zero, and these vectors are called linearly close by analogy with the situation encountered with "linearly dependent" vectors.

These ideas can be extended in a K-dimensional case and are developed in appendix B. Consider a space E_{K-1} (fig. 8) spanned by the independent vectors $\partial\hat{Y}/\partial a_1$, $\partial\hat{Y}/\partial a_2$, . . ., $\partial\hat{Y}/\partial a_{K-1}$, and a *Kth* vector, $\partial\hat{Y}/\partial a_K$. The K vectors $\partial\hat{Y}/\partial a_1$, $\partial\hat{Y}/\partial a_2$,. . ., $\partial\hat{Y}/\partial a_{K-1}$, $\partial\hat{Y}/\partial a_K$ are linearly close if the angle between $\partial\hat{Y}/\partial a_K$ and its projection on the space E_{K-1} is smaller than some threshold ε. It is also convenient to consider the distance between the extremity of $\partial\hat{Y}/\partial a_K$ and the space E_{K-1}. If $\partial\hat{Y}/\partial a_K$ is normalized to unity, this distance, s, is a measure of the "separation" between $\partial\hat{Y}/\partial a_K$ and E_{K-1}. The linear closeness condition is then expressed as

$$s < s_o = \sin(\varepsilon) \tag{8}$$

If the threshold s_o is chosen equal to zero, equation (8) defines the classical linear dependence. The choice of s_o greater than zero expresses the uncertainties that are due to real measurements and real computation, the latter introducing in addition some errors in the direction of the vectors $\partial\hat{Y}/\partial a_i$, which justifies even more strongly the necessity of the linear closeness concept.

In the set of the N vectors $\partial\hat{Y}/\partial a_i$ there may be different linearly close subsets that will be responsible for a nonunique solution in the identification. However, the nonuniqueness will affect only the parameters involved in these subsets. Therefore, if it is possible to find these dependent (or almost dependent) sets, not only the cause of nonuniqueness is found but also some cure. Indeed, by removing from the identification process one parameter in each set (i.e., leaving these parameters with a constant value) the degeneracy of the space E_N is removed and the identification may continue with a smaller number of parameters for which there will be a unique solution. Of course, this solution will depend upon the values given to the discarded parameters, but, and this is a key point, if the true value of the discarded parameter of a given dependent set is known from any other source of information, and if this value is given to this parameter, then the values obtained in the identification for the other parameters of this set are correct. Moreover, even if nothing is known, some parameters might not be involved at all in any dependent set, and for these independent parameters a correct answer will be obtained, regardless of the situation for the others. That is, if the dependent parameter is removed and the identification continued, the final value of all parameters that were not involved in the dependent sets will be correct.

An obvious point, which has also to be considered, is that, above all, a vector $\partial\hat{Y}/\partial a_i$ has to be nonzero. If such a vector is zero, this means the parameter a_i has no effect on the model, at least for the particular input-output set considered. Again one should realize that if $\partial\hat{Y}/\partial a_i$ is small enough,[2] a_i will be considered an irrelevant parameter. Therefore the irrelevance may be checked first, while computing the vectors $\partial\hat{Y}/\partial a_i$.

The problem of irrelevance, which is the simplest case of nonuniqueness, can be related to a linear closeness situation. For example, consider a problem with two parameters a_1 and a_2. If $\partial\hat{Y}/\partial a_2$ is found small enough, a_2 is declared irrelevant. Let us define then a new set of parameters a_1' and a_2' such that

$$a_1 = a_2' + a_1'$$

$$a_2 = a_2' - a_1'$$

it follows that

$$\frac{\partial\hat{Y}}{\partial a_1'} = \frac{\partial\hat{Y}}{\partial a_1} - \frac{\partial\hat{Y}}{\partial a_2}$$

$$\frac{\partial\hat{Y}}{\partial a_2'} = \frac{\partial\hat{Y}}{\partial a_1} + \frac{\partial\hat{Y}}{\partial a_2}$$

and the angular separation between these two vectors is

[2] The problem of defining a good criterion is delicate and it is difficult to give definite rules. One possible approach is to compare the relative change in the output to the relative change in the parameter. Also the precision with which the computations are performed has to be taken into account, since the change in the last digit is not very significant.

$$s < 2|\partial\hat{Y}/\partial a_2|/|\partial\hat{Y}/\partial a_1| \qquad (9)$$

which is small when $\partial\hat{Y}/\partial a_2$ is small. Therefore the irrelevance of a_2 is equivalent to a closeness between a_1' and a_2'.

In conclusion, the analysis of the structure of the space E_N provides an accurate way of detecting the occurrence of nonuniqueness and prevents this occurrence from causing a complete loss since it indicates not only the correctly identified parameters, but also what *a priori* knowledge is required to complete the identification. How such an analysis can be performed on a digital computer will be discussed next.

DEPENDENCE ANALYSIS

Two important steps that are required before considering the computer program are discussed in this section. The first step is concerned with the definition of the operations that must be performed and the mathematical principles on which they are based. The second step establishes the actual method of computing the numerical values corresponding to these operations. The mathematical background used in the section "Principle" below can be found in appendixes B and C, while the detailed analysis of the computation technique is given in appendix D. The program itself, which has been written in FORTRAN IV, is explained and listed in appendix E with an example of output.

Since the dependence analysis involves the N vectors $\partial\hat{Y}/\partial a_i$, it is of course supposed that the components of these vectors have been computed and stored, either as a part of the identification process, or only for the purpose of this analysis. The actual computation of these components will not be discussed since the technique may differ considerably depending upon the model, the type of problem and other practical considerations. However, it might be convenient to look for the irrelevant parameters at the time of this computation (i.e., check the magnitude of $\partial\hat{Y}/\partial a_i$ with respect to some criterion). In any case, a check for zero valued vectors is always made at the beginning of the dependence analysis.

Principle

The set of the M-dimensional vectors $\partial\hat{Y}/\partial a_i$ must be analyzed. Since the linear closeness involves only angular properties, a set of normalized vectors

$$D_i = (\partial\hat{Y}/\partial a_i)/|\partial\hat{Y}/\partial a_i| \qquad i = 1,N \qquad (10)$$

may be used instead. At this stage the irrelevant parameters have already been found and the corresponding vectors $\partial\hat{Y}/\partial a_i$ removed from the set, so that N is the number of relevant parameters and equation (10) is meaningful. Call B_N the set of the N vectors $\partial\hat{Y}/\partial a_i$. The problem is to find what are, if any, the almost dependent subsets B_K in the set B_N $(B_K \subset B_N,\ K = 1,N)$.

One possible method is the following: since any subset that contains a linearly close subset is itself linearly close (appendix B), one could first look for all the possible two-dimensional sets, then for the three-dimensional, and so on up to N-dimensional. Each time a subset is found linearly close, one vector could be removed and the process could be continued with a smaller number of vectors. This way would ensure that when a K-dimensional subset is found dependent (or almost dependent), by removing one vector of the subset, a K - 1 dimensional subset would be obtained that would contain only independent vectors. Unfortunately, the time involved makes the straightforward application of this method to a digital computer unfeasible since the computation of 2^N determinants is required (which already takes one minute on the IBM 360/60 for 12 parameters and this figure is doubled for each parameter added).

Another important consideration is that when a vector has to be removed, many choices are possible and some are better than others. We have seen indeed that the nonuniqueness comes from the fact that the basis vectors $\partial\hat{Y}/\partial a_i$ constitute a skew basis. Conversely, the more nearly orthogonal the basis, the less tendency to nonuniqueness. For example, if three vectors are found in a plane (fig. 9), the best choice is to discard D_2 because it leaves D_1 and D_3, which are almost orthogonal, to construct a two-dimensional basis. The worst choice is to discard D_1 which leaves a skew basis. Therefore it is advantageous if the technique provides also a way to optimize the basis. This can be done by constructing systematically an optimal basis with the vectors D_i. To start the process a first vector is chosen. Call n_1 the index of this vector (e.g., if D_3 is chosen, $n_1 = 3$); the parameter a_{n_1} corresponds to the first vector D_{n_1}. Many ways, optimal or not, may be devised to choose this vector, but generally it is a matter of common sense to know which parameter at least should be kept in the model. The next step is to find which of the N - 1 remaining vectors is the farthest from D_{n_1}. To do that, the distances s_i from the D_i to D_{n_1} are computed (fig. 10). If s_{n_2} is the largest, then the corresponding vector D_{n_2} is chosen to construct a two-dimensional subbasis with D_{n_1}. Also, the smallest value of the s_i is searched and compared to the threshold s_0. If it is larger than s_0 the name of the corresponding parameter is kept in memory as well as the value of s_i. This defines a "critical" parameter, in the sense that, if the threshold was increased, it would have been considered dependent. Therefore the knowledge of this critical value s_i might give an indication of potential difficulties. If s_i is less than s_0, the corresponding parameter is declared dependent and D_i removed from the set of vectors. The same process is now applied to find the third basis vector D_{n_3}. That is, all the distances from the N - 2 remaining vectors to the plane (D_{n_1}, D_{n_2}) are computed, and D_{n_3} corresponds to the largest distance. On the example of figure 10, when the four-dimensional subbasis is being formed, the distance from D_j to the three-dimensional subbasis $(D_{n_1}, D_{n_2}, D_{n_3})$ is found to be less than s_0. Therefore D_j is discarded as linearly close to this subbasis. To memorize this event involving the parameter a_j, a corresponding integer MN(j) is set to 0 in the program. This integer, or "dependency index," will be used internally by the computer to keep track of the dependent, independent or discarded parameters. Then, by computing the projections of D_j on the subbasis, D_j is found to be in the plane (D_{n_2}, D_{n_3}). Therefore the actual almost dependent set is (D_j, D_{n_2}, D_{n_3}). This is memorized by setting $MN(n_2)$ and $MN(n_3)$

equal to 1 (meaning that a_{n_2} and a_{n_3} are dependent parameters) and by storing j, n_2 and n_3 in a logical array (equivalent to the information, *the first dependent set contains the parameters a_j, a_{n_2} and a_{n_3}*). More details on the program are given in appendix E.

The above process will continue until all the N vectors have been either used in the basis or discarded. Since MN was set to 2 for all the parameters to begin the analysis, if a vector has never been involved in any dependent set, its dependency index will keep this value, so that a 2 will indicate an independent parameter. Finally a union of the dependent sets may be performed if they have a nonempty intersection. For instance if the dependent sets (a_5, a_7) and (a_2, a_5, a_8) are found, they may be combined in the set (a_2, a_5, a_7, a_8), and in this case if both a_7 and a_8 are discarded, the final values found in the identification for a_5 and a_2 will depend upon those of a_7 and a_8. It is essential to note at this point that if some linearly close subsets have been found, the final basis will be composed of only N' vectors (N' < N). This means that only N' parameters can be uniquely identified. Hence N' - N parameters have to be removed from the identification process (i.e., their value will not be changed). An optimal choice is to remove those for which MN = 0, and this can be done automatically in the program. Otherwise the user has to decide which parameter he wants to remove in each dependent subset and reset his identification program correspondingly. Therefore, although the removal of some vectors D_i is required in the dependence analysis in order to find the dependent subsets, it does not imply the removal of the corresponding parameters. That operation belongs only to the application of the analysis.

Computation Technique

Once the irrelevant parameters have been eliminated, it is possible to compute the normalized vectors D_i and start the construction of the optimal basis. At each step of this construction, the projections of a vector on the last subbasis must be computed. This still seems very strenuous, but we have found that all these quantities can be obtained during the recursive computation of a single N × N determinant (which is quite an improvement compared to 2^N determinants needed in a straightforward method).

Consider indeed the matrix D whose N columns are the M-component vectors D_i and form the product

$$G = D^T D$$

where G is an N × N matrix, positive semidefinite, and the elements g_{ij} are equal to the inner products $D_i^T D_j$.[3] Because of the normalization, the diagonal elements are 1's. The determinant of G is called the *Gram determinant* of the vectors $D_1, D_2, \ldots, D_N$, and is equal to the square of the

[3] It is also interesting to have a quick look at this matrix because if an off-diagonal element g_{ij} is equal or very close to 1, it indicates that D_i and D_j are very close (i.e., the parameters a_i and a_j are dependent). In this case s is simply equal to $(1 - g_{ij}^2)^{1/2}$.

volume of the N-dimensional hyperparallelepiped determined by these vectors (ref. 9).

$$\mathrm{DET}(G) = V_N^2$$

Computing this determinant is equivalent to computing a volume. Since there is a recursive method of computing an N-dimensional volume which involves the distances s_i, a recursive method of computing the determinant should exist that involves these quantities. If V_K is the volume corresponding to a K-dimensional subbasis, the volume obtained by adding a unit vector D_{K+1}, at the distance s_{K+1} from the subbasis, is indeed

$$V_{K+1} = V_K \times s_{K+1}$$

Squaring the two sides of this relation leads to the recursive relation for the determinant. It is shown in appendix D that this can be obtained by pivoting about the diagonal elements. At the *Kth* step, the situation is as represented in figure 11. By rows and columns exchange, and renormalization, the K first rows (corresponding to the K basis vectors $D_{n_1}, D_{n_2}, \ldots, D_{n_K}$) form an upper triangle with the diagonal elements equal to 1. The other diagonal elements such as $g_{ii}, g_{jj}, \ldots$ are, respectively, equal to the square of the distance from $D_i, D_j, \ldots$ to the subbasis $(D_{n_1}, D_{n_2}, \ldots, D_{n_K})$. Indeed, it can be seen directly that the determinant corresponding to this subbasis *plus*, say D_i, is equal to the determinant corresponding to the subbasis multiplied by g_{ii}. Symbolically

$$\mathrm{DET}_{K+1} = \mathrm{DET}_K \times g_{ii}$$

which is precisely the recursive relation discussed above. Therefore searching the elements $g_{ii}, g_{jj}, \ldots$ for the largest and the smallest will end in the determination of the next basis vector and of the critical one. If for instance g_{jj} is found to be less than s_o^2, then it is a simple matter to obtain the components of D_j on the subbasis to determine the members of the dependent set (appendix D), because the system of equations to be solved is already triangular. In the same way a few extra computations at the end of the analysis gives the solution dA' of the equation

$$(D^T\ D)dA' = D^T\ ER$$

from which the solution for equation (5) is simply obtained by

$$da_i = -da_i'/|\partial\hat{Y}/\partial a_i| \qquad (i = 1,N \text{ and } MN(i) \neq 0)$$

Computer Output

During the computation of the Gram determinant the following information was obtained:

1. Name and separation of the new basis parameter

2. Name and separation of the critical parameter

3. Names of the parameters found in a dependent set

4. Name and separation of the parameter optimally discarded from this set

5. Also correlation coefficients between this parameter and the other parameters of the dependent set (see appendix A)

6. Values of the dependency index (0, 1, or 2) for each parameter

7. United sets and name of the optimally discarded parameters

These are the type of items an engineer may want and this is what we have displayed in our program (appendix E).

Certainly, there are as many ways to use this subroutine as there are users, but some hints may still be indicated in this discussion. Two kinds of corrective action can be undertaken following this analysis. One is entirely automatic and controlled by the zeroes of the dependency index that are used as a signal for the computer to bypass any computation related to the corresponding parameter. This is compulsory if a matrix inversion is used in the identification algorithm. It will speed up the computation in other cases. The second kind of action is a human decision based on the information obtained and a general insight of the particular problem. Consider for instance the list of the basic parameters. It will in fact contain all the parameters if the subroutine is called with a threshold equal to 0.0. Because of the optimization procedure, the parameters are sorted by increasing degree of dependence. Since the separation is also a measure of the confidence in the parameter estimate (appendix B), this list expresses the intrinsic properties of the identification independently of any statistical property of the noise. Consider now the choice of the discarded parameter when a threshold is specified. What is indicated by the computer is only an optimal choice, which will give the best results in the identification of the remaining parameters. In the example in appendix E, a_9 and a_{14} were discarded in the first dependent subset. If the value of these parameters could be obtained from other measurements, the identification could be run again with these values. However, the engineer may find that known values are available only for a_8 and a_{13}, in which case he will decide to identify a_9, a_{11}, and a_{14}. This kind of tradeoff is typical of real problems and cannot be ignored. It requires more skill than brute logic; therefore it cannot always be mechanized automatically on the computer.

APPLICATION TO THE IDENTIFICATION OF A SATELLITE SIMULATOR

Data Acquisition System

The Ames Research Center three-degrees-of-freedom satellite simulator is an air-bearing supported platform provided with a system of servo-gimbals and mercury connectors allowing accurate measurements of the attitude angles and a torque-free link with the outside (ref. 10). This particular air-bearing configuration allows about ±20° amplitude in roll and pitch and there is no limit in yaw.

On figure 12 is sketched the general configuration of the whole system for data acquisition and processing used for the identification of the platform. The attitude rates (p, q, and r) are sensed by three rate gyros whose analog outputs are digitized, after filtering, and recorded on magnetic tape by an E.A.I. 8400 computer. This tape has to be converted to an IBM compatible tape from which data cards can be punched. These cards contain therefore the dynamic data (i.e., the discrete time histories of the three rates). The initial attitude angles of the platform (Euler angles) are read on the gimbal readout display and the corresponding values punched manually on cards (static data). The identification is run on an IBM 360 digital computer which receives the identification program. A special set of control cards is used to specify the different operations performed by the identification program.

Equations of Motion

The nonlinear equations used to describe the system in the case of large angular motion may be written:

$$\dot{g} = S(\omega)g \tag{11}$$

$$\dot{h} = S(\omega)h + T \tag{12}$$

$$\omega = J^{-1}h \qquad J = \begin{pmatrix} Jx & Jxy & Jxz \\ Jxy & Jy & Jyz \\ Jxz & Jyz & Jz \end{pmatrix} \tag{13}$$

where ω is the angular velocity of the platform (of components p, q, and r), $S(\omega)$ is a matrix expression for the crossproduct:

$$S(\omega) = \begin{pmatrix} 0 & r & -q \\ -r & 0 & p \\ q & -p & 0 \end{pmatrix} \qquad (S(\omega)g = g \times \omega)$$

g is the gravity acceleration vector, h the angular momentum, J the inertia tensor of the platform and T represents the applied torque.

These equations have been developed in reference 11 for attitude control analysis. They are written here in body coordinates with the origin at the center of the bearing. The first equation is purely kinematic, as it expresses the change in the coordinate systems (in its general form it involves the whole attitude matrix, but in this particular problem, where no control in yaw angle was considered, the direction of the vertical only was required), the second equation is a torque equation, and the last one is just the classical definition of the angular momentum.

In normal operation (i.e., for satellite simulation purpose) the platform is completely balanced, that is, the center of gravity coincides with the bearing center. In our experiments the center of gravity was offset deliberately to make the system pendulus. If M is the total mass and b/M the bearing offset (three components vector), the torque exerted by the gravity is:

$$T_G = -S(b)g \tag{14}$$

The natural damping of the system, mainly due to aerodynamic effects, was quite small and in some tests it was convenient to use the reaction wheels as dampers. The three axes were damped separately; the corresponding control system is shown on figure 13 for one axis. Considering one axis, we use the classical linear model for the DC motor,

$$E = RI + K_B\Omega \tag{15}$$

where E is the applied voltage, I the current in the motor, R the resistance of the wiring K_B a back electromotive force constant, and Ω the angular velocity of the shaft (which is also that of the wheel). The torque available is proportional to the current and satisfies

$$T_M = K_T I = J_W\dot{\Omega} + K_D\Omega \tag{16}$$

where J_W is the inertia of the rotor-wheel system and K_D a damping constant. When the wheel is used as damper, the applied voltage E is controlled by the platform rate, say the roll rate p:

$$E = C_1 p \tag{17}$$

where C_1 is a control gain. Eliminating E and I from equations (15), (16), and (17) and calling h_{W1} the angular momentum of the roll wheel ($h_{W1} = J_W\Omega$), we obtain finally an equation of the type

$$\dot{h}_{W1} = C_1 p - \alpha_1 h_{W1} \tag{18}$$

where α_1 is a constant.

Considering now the total angular momentum h_W of the wheels (of components h_{W1}, h_{W2}, h_{W3}), the control equation may be written as

$$\dot{h}_W = C\omega - \alpha h_W \tag{19}$$

where C and α are diagonal matrices corresponding, respectively, to control gains and back electromotive force plus damping factors. The main nonlinearities were introduced by the torque and speed limiters. They enter the preceding equation as limits in the values of the components of $C\omega$ and $\dot{h}_W$, respectively.

Finally the torque exerted by the wheels on the platform can be expressed as

$$T_W = S(\omega)h_W - \dot{h}_W \tag{20}$$

and the complete system is modeled by letting the torque T in equation (12) be

$$T = T_W + T_G \tag{21}$$

When the wheels are not used, T_W is replaced by a linear term in ω to model the natural damping, that is $DMP\omega$, where DMP is a diagonal matrix.

In all cases, the "output" of the system was obtained by the three rates p, q, and r seen through the first-order filters and the digitizer (fig. 12). Thus three equations have to be added to model the system completely:

$$\left.\begin{aligned} \tau\dot{y}_1 &= k_1p - y_1 \\ \tau\dot{y}_2 &= k_2q - y_2 \\ \tau\dot{y}_3 &= k_3r - y_3 \end{aligned}\right\} \tag{22}$$

The DC gains of the filters (k) were obtained by independent calibration of the gyros and computer links, and the time constant τ was known from the setting of the analog filters.

Finally the complete sets of equations corresponding to the free pendulus (natural damping) case, and the reaction-wheel-damped case may be written:

Case A: natural damping

$$\left|\begin{matrix} p \\ q \\ r \end{matrix}\right| = \begin{bmatrix} J_x & J_{xy} & J_{xz} \\ J_{xy} & J_y & J_{yz} \\ J_{xz} & J_{yz} & J_z \end{bmatrix}^{-1} \left|\begin{matrix} h_1 \\ h_2 \\ h_3 \end{matrix}\right|$$

$$\dot{\left|\begin{matrix} g_1 \\ g_2 \\ g_3 \\ h_1 \\ h_2 \\ h_3 \end{matrix}\right|} = \begin{bmatrix} 0 & r & -q & 0 & 0 & 0 \\ -r & 0 & p & 0 & 0 & 0 \\ q & -p & 0 & 0 & 0 & 0 \\ 0 & -b_z & b_y & -DMP_x & -h_3 & h_2 \\ b_z & 0 & -b_x & h_3 & -DMP_y & -h_1 \\ -b_y & b_x & 0 & -h_2 & h_1 & -DMP_z \end{bmatrix} \left|\begin{matrix} g_1 \\ g_2 \\ g_3 \\ p \\ q \\ r \end{matrix}\right| + a_{13}$$

with the equations of the filters:

$$\tau\dot{y}_1 = k_1 p - y_1$$

$$\tau\dot{y}_2 = k_2 q - y_2$$

$$\tau\dot{y}_3 = k_3 r - y_3$$

The parameter a_{13} appearing in the last torque equation was needed to describe a constant torque in yaw experienced during the tests (because of some drift of the system not corrected at this time).

Case B: reaction-wheel-damped case

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} J_x & J_{xy} & J_{xz} \\ J_{xy} & J_y & J_{yz} \\ J_{xz} & J_{yz} & J_z \end{bmatrix}^{-1} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix}$$

$$\dot{\begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ h_1 \\ h_2 \\ h_3 \end{bmatrix}} = \begin{bmatrix} 0 & r & -q & 0 & 0 & 0 \\ -r & 0 & p & 0 & 0 & 0 \\ q & -p & 0 & 0 & 0 & 0 \\ 0 & -b_z & b_y & -C_1 & -h_3 & h_2 \\ b_z & 0 & -b_x & h_3 & -C_2 & -h_1 \\ -b_y & b_x & 0 & -h_2 & h_1 & -C_3 \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ p \\ q \\ r \end{bmatrix} + \begin{bmatrix} \alpha_1 & r & -q \\ -r & \alpha_2 & p \\ q & -p & \alpha_3 \end{bmatrix} \begin{bmatrix} h_{W1} \\ h_{W2} \\ h_{W3} \end{bmatrix} + a_{13}$$

$$\dot{\begin{bmatrix} h_{W1} \\ h_{W2} \\ h_{W3} \end{bmatrix}} = \begin{bmatrix} C_1 & 0 & 0 \\ 0 & C_2 & 0 \\ 0 & 0 & C_3 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} - \begin{bmatrix} \alpha_1 & 0 & 0 \\ 0 & \alpha_2 & 0 \\ 0 & 0 & \alpha_3 \end{bmatrix} \begin{bmatrix} h_{W1} \\ h_{W2} \\ h_{W3} \end{bmatrix}$$

$$\tau\dot{y}_1 = k_1 p - y_1$$

$$\tau\dot{y}_2 = k_2 q - y_2$$

$$\tau\dot{y}_3 = k_3 r - y_3$$

with the limiting conditions:

(motor torque) $|C_1 p| < a_{16}$, $|C_2 q| < a_{17}$, $|C_3 r| < a_{18}$

(wheels spin) $|h_{W1}| < a_{22}$, $|h_{W2}| < a_{23}$, $|h_{W3}| < a_{24}$

Parameter Identification Problems

An analysis of the equations can show that, in case A or B, there is not a unique solution for all unknown parameters. One parameter has to be known; hence the lateral offset b_x was measured independently because it was easy to obtain by adding a small weight on the side of the platform, at a known distance from the bearing. For case A, 12 unknown parameters remain to be identified: the 6 inertias and products (Jx, Jy, Jz, Jxy, Jxz, Jyz), the 2 center-of-gravity offsets (b_y, b_z), the 3 damping terms (DMP_x, DMP_y, DMP_z) and the turbine torque a_{13}. In case B, 21 parameters are to be identified: 9 of the parameters as in case A (all except DMP_x, DMP_y, DMP_z) plus the 3 control gains C_1, C_2, and C_3, the 3 back electromotive force and motor damping terms α_1, α_2, α_3, the 3 limits in motor torque a_{16}, a_{17}, a_{18}, and the 3 limits in wheel momentum a_{22}, a_{23}, a_{24}.

Experimental Results

In the two cases, the platform was first held in some position by an electromechanical device, and the attitude angles (pitch θ_o and roll ϕ_o) measured. The initial conditions therefore were

$$g_1 = -\sin(\theta_o)$$

$$g_2 = \cos(\theta_o)\sin(\phi_o) \qquad p = q = r = 0$$

$$g_3 = \cos(\theta_o)\cos(\phi_o)$$

Because of the gravity restoring torque, once the platform was released an oscillatory motion was observed about the three axes, and the corresponding discrete time histories of p, q, and r were recorded. About 200 points were used in each, covering 5 or 6 periods, thus the vector Y had about 600 components.

When the motion of a body is excited about one axis only, it only depends upon the inertia about this axis and no information is available concerning the other inertias. This has actually been the current method to determine the inertias of aircraft where great care is taken to obtain single-axis excitations. In the case of the platform, it is intuitive that motions about the three axes should be excited to obtain a good identification of the inertias and products. Therefore some difficulties may be expected because the yaw motion is poorly excited by the gravity restoring torque. This motion depends of course on the lateral center-of-gravity offset and the initial conditions, but both are limited by the angular limitations of the platform, and in our tests the yaw rate was at most 0.1 or 0.2 of the roll or pitch rate.

Case A- These data were taken in the free-oscillation case (no reaction wheels damping), with the initial conditions:

$$p, q, \text{ and } r = 0$$

pitch angle = -8.44°

roll angle = 10.44°

Lateral bearing offset on the x axis: $b_x = 0.211$ ft-lb. The first identification was run with a threshold set at $s_o = 0$. All 12 parameters were identified and a very good fit in the time histories was obtained after a few iterations. This fit was measured by a quantity denoted as "COST" which was simply the sum of the square of the errors, that·is $|ER|^2$. The decimal logarithm of this quantity is plotted on figure 14 as is the evolution of the values for the six elements of the inertia tensor. Although the behavior of the parameters is somewhat erratic at the beginning, they finally reach an asymptotic value for which the cost remains essentially constant. Therefore one might have been satisfied with these results and considered that these final values of the parameters were correct. The dependence analysis, however, indicated a low value for the separation of the parameter (J_{xz}). Consequently, the identification was run again, with the same initial values for the parameters, but with a threshold set at $s_o = 0.02$. As it appears on figure 14, the evolution of the parameter values is quite different in this run. Indeed, J_{xz} was discarded at the second and following iterations because of its dependence on J_x, J_z, and a_{13}. The diagnostic issued by the computer when the final values were obtained is given in appendix E. The important fact to note here is that these final values are different from those obtained before, yet the match in the time histories is as good. To illustrate this similar match, a plot of computed and measured time histories of the roll rate is shown on figure 15 for these two identification runs.

The final numerical results obtained in these two runs are given in table 1. These results deserve some comments and explanations. The "error bounds" were computed according to equation (B8):

$$(da_K)^2 = \frac{\varepsilon_o^2}{s_K^2|\partial\hat{Y}/\partial a_K|^2}$$

where the measurement error ε_o was estimated from the residual error between model and plant output when the best match was obtained. Note that the values of the independent parameters (dependency index = 2), although different in each run, are within the predicted error. The dependent parameters, however, (dependency index = 1) exhibit large variations as a result of the nonuniqueness. Another quantity given in these tables is the *sensitivity*. It is a measure of the model response to parameter change and it is defined by

$$\text{Sensitivity to the parameter } a_i = \frac{a_i}{|\hat{Y}|} \times \left|\frac{\partial\hat{Y}}{\partial a_i}\right|$$

A sensitivity of 1 corresponds to a direct proportionality between the corresponding parameter and the output. If the sensitivity is very small with respect to 1 it indicates that the output is almost independent of the parameter, and large errors may be expected in the estimation of this parameter.

Therefore this quantity can be used as a criterion for deciding whether a parameter is irrelevant. Sensitivity and error bounds are not a part of the dependence analysis but are computed independently in the identification program.

Finally, a partial check of these identifications was made. The value of b_z, vertical offset of the center of gravity, was determined independently by a static test of the platform, and found equal to 0.78 ft-lb, which agrees well with the results of the two identification runs.

Case B- To test the algorithm in a strongly nonlinear case, another identification was performed with reaction wheel damping. It provided an interesting illustration of the problems encountered and how they were detected by the algorithm.

The initial conditions for this case were:

$$p,\ q,\ \text{and}\ r = 0$$

$$\text{pitch angle} = -0.65°$$

$$\text{roll angle} = 12.98°$$

Lateral bearing offset on the x axis:

$$b_x = 0.134\ \text{ft-lb}$$

$$\text{threshold} = 0.01$$

On figure 16 are shown the computed and measured time histories of the roll rate. (The wheel spin is also plotted exhibiting sharp saturation effects.) A residual systematic (not random) error is observed here that must be attributed to a modeling error. Indeed, many assumptions were made in modeling the control system, mainly concerning the linearity of the DC motors and the absence of friction in the wheel bearings. However, considering the approximations made, the agreement is quite satisfactory. The final results of the identification are given in table 2. The parameters a_{18} and a_{24} were found to be irrelevant. The latter represents the limit of the yaw wheel momentum, corresponding to the spin limitation of the yaw wheel. Indeed, the yaw wheel momentum time history showed that it never exceeded its limit; therefore it was not possible for the computer to determine this limit. For a similar reason, the limit in torque a_{18} could not be determined. (Note that the sensitivity of these two parameters was found equal to zero.)

CONCLUDING REMARKS

A method has been described that determines which parameters of a system model can be uniquely identified from a given set of measurements. It applies to nonlinear systems and does not require any special form for the model, providing that the model is completely defined and depends upon a finite number of parameters. The basis of this method is that all the necessary information is contained in the derivatives of the error vector with respect to the parameters. These derivatives are vectors in an M-dimensional space defined by the measurements (their components are often called "sensitivity functions" or "influence functions"). The nonuniqueness has been shown to be the result of linear dependence between these vectors in the ideal case of perfect measurements and exact computation. The concept of "linear closeness" has been introduced to take into account the uncertainties introduced by the errors occurring in the measurements and in the computation, thus allowing a realistic analysis of practical problems.

To perform this analysis, an algorithm has been developed for digital computation that can easily be integrated in a complete identification procedure. This is done practically at no extra cost in computation time if the procedure already requires the computation of the sensitivity functions (as in multiple linear regression or in quasilinearization for instance).

The detailed comments put out by the computer not only point out where the identification has, or is going to fail, but also indicate what could be done to remove the nonuniqueness. A very useful feature of this algorithm is its ability to obtain the correct values of some parameters (independent parameters) despite the fact that the complete solution might not have been unique.

Successful results obtained in the identification of a three-degrees-of-freedom satellite simulator indicate that the air-bearing technique might be applied to the determination of the inertia tensor of an actual aircraft. The identification of a highly nonlinear control system implemented in the simulator has also proven the effectiveness of the algorithm in such cases.

It must be noted finally that beyond the strict application of this technique to identification, there are interesting potentialities in other fields, in modeling for instance. Indeed, the algorithm indicates how the model could be simplified since the most dependent parameters can be eliminated by raising the value of the threshold. As the threshold is raised more and more, the model becomes more and more simple, although less and less accurate. Another possible use is the practical tailoring of the input to obtain optimal results in the identification. In this case, one could maximize the separation of a given parameter, or some average upon a group of parameters, by adjusting the parameters defining the input sequence.

Ames Research Center
National Aeronautics and Space Administration
Moffett Field, Calif., 94035, May 14, 1971

APPENDIX A

LINEAR DEPENDENCE AND FUNCTIONAL RELATIONSHIPS

The linear dependence between the K (M-component) vectors $\partial\hat{Y}/\partial a_1$, $\partial\hat{Y}/\partial a_2$, . . ., $\partial\hat{Y}/\partial a_K$ may be expressed by

$$\frac{\partial\hat{Y}}{\partial a_K} = \sum_{i=1}^{K-1} \lambda_i \frac{\partial\hat{Y}}{\partial a_i} \tag{A1}$$

where the λ_i are some nonzero functions of the parameters a_i, and the K - 1 vectors $\partial\hat{Y}/\partial a_i$ are supposed independent. When a_i are varying, $\hat{Y}$ is constrained to stay on a K - 1 dimensional hypersurface. It is generally possible to find a system of coordinates $(z_1, z_2, \ldots, z_{K-1})$ on this surface so that $\hat{Y}$ may be expressed as a function Z of the coordinates z:

$$\hat{Y}(a_1, a_2, \ldots, a_K) = Z(z_1, z_2, \ldots, z_{K-1}) \tag{A2}$$

Therefore $\hat{Y}$ *is not really a function of the K parameters but depends upon K - 1 functions of them.* If the partials of both sides of equation (A2) are taken with respect to a and compared to equation (A1), it can be shown that the functions z satisfy the system of equations:

$$\frac{\partial z_j}{\partial a_K} = \sum_{i=1}^{K-1} \lambda_i \frac{\partial z_j}{\partial a_i} \qquad j = 1, K - 1 \tag{A3}$$

Given the values of a, the values of z are well defined. The converse is not true, but if a_K is known, then there must be a one-to-one correspondence between z and the first K - 1 parameters, since $\hat{Y}$ is now a function of these K - 1 parameters only. The necessary and sufficient condition on the functions z is that their Jacobian with respect to the K - 1 a is not zero; that is,

$$\frac{\partial(z_1, z_2, \ldots, z_{K-1})}{\partial(a_1, a_2, \ldots, a_{K-1})} \neq 0 \tag{A4}$$

We now examine the case where the vectors $\partial\hat{Y}/\partial a_i$ are only linearly close (see appendix B for definitions). Equation (A1) becomes

$$\frac{\partial\hat{Y}}{\partial a_K} = \sum_{i=1}^{K-1} \lambda_i \frac{\partial\hat{Y}}{\partial a_i} + \varepsilon_K \tag{A5}$$

where ε_K is a (M-component) vector orthogonal to the space E_{K-1} spanned by the $K - 1$ vectors $\partial\hat{Y}/\partial a_i$ and of small magnitude. In this case $\hat{Y}$ depends also upon a_K and may be written as

$$\hat{Y}(a_1, a_2, . . ., a_K) = Z(z_1, z_2, . . ., z_{K-1}, a_K)$$

if z_i are chosen in such a way that they verify equation (A3), then using equations (A3) and (A5) one obtains

$$\frac{\partial Z}{\partial a_K} = \varepsilon_K \qquad (A6)$$

The change in $\hat{Y}$ when a_K is changed while z_i are kept constant is thus

$$d\hat{Y} = dZ = \varepsilon_K da_K$$

For a small enough ε_K, $d\hat{Y}$ will be negligible and, because of equation (A6), the property equation (A2) will be verified.

Let us now evaluate the total change in $\hat{Y}$ for arbitrary da,

$$d\hat{Y} = \sum_{i=1}^{K-1} \left(\frac{\partial\hat{Y}}{\partial a_i}\right) da_i + \left(\frac{\partial\hat{Y}}{\partial a_K}\right) da_K$$

Eliminating $\partial\hat{Y}/\partial a_K$ with equation (A5) it follows that

$$d\hat{Y} = \sum_{i=1}^{K-1} (da_i + \lambda_i da_K)\left(\frac{\partial\hat{Y}}{\partial a_i}\right) + \varepsilon_K da_K \qquad (A7)$$

If ε_K is small enough, then for a given change da_K there exist da_i such that $d\hat{Y}$ stays close to zero. From equation (A7) we see that the values of such da_i are given by

$$da_i = -\lambda_i da_K \qquad (A8)$$

Therefore if a value was given to a_K with an error da_K, the corresponding errors in the values of the a_i obtained in the identification are given by equation (A8). Consequently, it is very useful to compute the quantities λ_i. In the algorithm (see appendix D) a normalized set of vectors is used:

$$D_j = \frac{(\partial\hat{Y}/\partial a_j)}{|\partial\hat{Y}/\partial a_j|} \qquad j = 1, K$$

and the quantities μ_i are computed. The μ_i are defined by transforming equation (A1) into

$$D_K = \sum_{i=1}^{K-1} \mu_i D_i$$

They are the components of D_K upon the vectors D_i and are related to λ_i by

$$\lambda_i = \frac{\mu_i |\partial\hat{Y}/\partial a_K|}{|\partial\hat{Y}/\partial a_i|}$$

APPENDIX B

DEFINITION AND PROPERTIES OF LINEAR CLOSENESS

Consider a set B_K of K unit vectors $(D_1, D_2, \ldots, D_K)$ spanning a space E_K. The first $K - 1$ vectors span a space E_{K-1}. Let e_K $(e_K \in E_K)$ be a unit vector orthogonal to E_{K-1}. Then D_K may be decomposed as

$$D_K = s_K e_K + \sum_{i=1}^{K-1} \mu_i D_i \tag{B1}$$

The μ_i are the components of D_K on the D_i in the basis $(e_K, D_1, D_2, \ldots, D_{K-1})$ and s_K is a positive scalar (this can always be done by a proper choice of e_K) equal to the sine of the angle between D_K and its projection on E_{K-1}. When s_K is zero, equation (B1) defines a strict linear dependence between the vectors D. If s_K is arbitrarily small, these vectors become almost dependent and s_K is a measure of their degree of dependence or of the separation between D_K and the space E_{K-1}. We may therefore define the linear closeness as:

> *The vectors of a given set B_K are linearly close with respect to the threshold s_o iff there exists a vector D_K of the set such that its separation s_K from the others is less than or equal to s_o.*

We may write this symbolically: B_K is LC/s_o *iff* $\exists\, D_K \in B_K$ such that $s_K \leqslant s_o$.

What has been defined is obviously an angular property, therefore it will not change if each vector D is multiplied by a different scalar. For any set of vectors, it is possible to define a corresponding set of normalized vectors, B_K, and *iff* B_K is LC the original set is also LC. For convenience and ease in the proofs, we always work with the normalized set.

Consider a set B_K where D_K is given by equation (B1). Introduce another vector, say D_{K+1}, to construct a new set B_{K+1} $(B_{K+1} = B_K \cup D_{K+1})$. Then D_K may be written

$$D_K = s_{K+1} e_{K+1} + \sum_{i=1}^{K-1} \mu_i' D_i + \mu_{K+1}' D_{K+1} \tag{B2}$$

where e_{K+1} is a unit vector orthogonal to D_i and D_{K+1}, and μ_i' are some scalars. Recalling that

$$e_K^T e_K = 1 \qquad e_K^T D_i = 0$$

$$e_{K+1}^T e_{K+1} = 1 \qquad e_{K+1}^T D_i = 0 \qquad i = 1, K - 1$$

$$e_{K+1}^T D_{K+1} = 0$$

one obtains from equation (B2)

$$e_{K+1}^T D_K = s_{K+1}$$

and from equation (B1)

$$e_{K+1}^T D_K = s_K(e_{K+1}^T e_K)$$

therefore

$$s_{K+1} = (e_K^T e_{K+1}) s_K$$

$(e_K^T e_{K+1})$ is positive (because s_K and s_{K+1} are positive) and less or equal to 1 (because equal to $\cos(\theta)$, where θ is the angle between the two vectors e), thus we obtain the important inequality

$$s_{K+1} \leqslant s_K \tag{B3}$$

Property 1- This result shows that *the increase in dimensionality* always *increases the dependence* (unless the new vector added is orthogonal to the others in which case nothing is changed), which means that *when the number of parameters is increased there is more chance for a nonunique solution.*

Now suppose that a set B_N is given in which the subset B_K ($B_K \subset B_N$) is LC. From the definition

$$s_K \leqslant s_o$$

Starting from B_K, add the vector D_{K+1} to construct the set $B_{K+1} \subset B_N$. Because of equation (B3)

$$s_{K+1} \leqslant s_o$$

therefore B_{K+1} is also LC. Continuing to add vectors this way, we see that ultimately $s_N \leqslant s_o$ and we have

Property 2- If a set includes a LC subset, it is itself LC, and its separation is less than or equal to that of the subset.

To summarize the preceding results, let us introduce this notation: if B is a set of vectors spanning some space E, and D a vector not belonging to the set B (but belonging eventually to the space E), the separation s between D and B is written as

$$s = S(D/B)$$

(As has been seen, $0 \leqslant s \leqslant 1$.)

Definition of the Linear Closeness

B is LC/s_o *iff* $\exists\ D \in B$ such that $S(D/B') \leqslant s_o$, where $B = B' \cup D$.

Properties-

LC 1:

$$S(D/B \cup D') \leqslant S(D/B) \qquad \forall B, D \text{ and } D'$$

LC 2: If $B' \subset B$ and B' is LC/s_o, then B is LC/s_o' with $s_o' \leqslant s_o$.

LC 3:

$$\frac{S(D_1/B \cup D_2)}{S(D_2/B \cup D_1)} = \frac{S(D_1/B)}{S(D_2/B)}$$

The proof of this last property is very simple; consider the volume of the hyperparallelepiped constructed with the set $B \cup D_1 \cup D_2$:

$$V(B \cup D_1 \cup D_2) = V(B \cup D_1) \times S(D_2/B \cup D_1) = V(B \cup D_2) \times S(D_1/B \cup D_2)$$

In the same way

$$V(B \cup D_1) = V(B) \times S(D_1/B) \quad \text{and} \quad V(B \cup D_2) = V(B) \times S(D_2/B)$$

which leads to LC 3 by substitution.

Relationship Between Separation and Error in Parameter Estimates

Let s_K be the separation between D_K and the set B_{K-1}. We have as in equation (B1)

$$D_K = s_K e_K + \sum_{i=1}^{K-1} \mu_i D_i \tag{B4}$$

When there is some "noise" in the measurement vector Y, not only Y lies outside of the subspace E_K, but also its projection Y_o on this subspace is shifted by some amount δY_o from its value in the absence of noise. As a matter of fact, because the identification procedure tends to make the computed vector $\hat{Y}$ reach the target Y_o, the component of the noise orthogonal to E_K does not affect the result, whereas the component in E_k, δY_o, is directly responsible for the error in the parameter estimates. This uncertainty δY_o may result from random perturbations of the system, or instrument errors, quantization errors, nonlinearities or other unknowns, but we may assume that some bound is known for δY_o, for instance,

$$\delta Y_o^T \cdot \delta Y_o \leqslant \varepsilon_o^2 \tag{B5}$$

Therefore a set of parameter values that causes $\hat{Y}$ to satisfy

$$d\hat{Y}^T d\hat{Y} = (\hat{Y} - Y_o)^T \cdot (\hat{Y} - Y_o) \leqslant \varepsilon_o^2 \tag{B6}$$

is as good (or as bad) as any since Y_o is not known exactly. If the parameters are changed by the amounts $da_1, da_2, \ldots, da_K$, the corresponding change in $\hat{Y}$ may be written as

$$d\hat{Y} = \sum_{i=1}^{K-1} \left|\frac{\partial \hat{Y}}{\partial a_i}\right| D_i \, da_i + \left|\frac{\partial \hat{Y}}{\partial a_K}\right| D_K \, da_K$$

using equation (B4) leads to

$$d\hat{Y} = \sum_{i=1}^{K-1} \left(\left|\frac{\partial \hat{Y}}{\partial a_i}\right| da_i + \mu_i \left|\frac{\partial \hat{Y}}{\partial a_K}\right| da_K \right) D_i + s_K \left|\frac{\partial \hat{Y}}{\partial a_K}\right| da_K \; e_K \tag{B7}$$

or

$$d\hat{Y} = V + \left(s_K \left|\frac{\partial \hat{Y}}{\partial a_K}\right| da_K \right) e_K \qquad \text{(where } V \in E_{K-1} \text{ and } e_K \perp V\text{)}$$

finally

$$d\hat{Y}^T \cdot d\hat{Y} = |V|^2 + s_K^2 \left|\frac{\partial \hat{Y}}{\partial a_K}\right|^2 (da_K)^2$$

Comparing with equation (B6) gives us an upper bound for da_k, the error in a_K

$$(da_K)^2 \leqslant \frac{\varepsilon_o^2}{s_K^2 |\partial \hat{Y}/\partial a_K|^2} \tag{B8}$$

When a_K is not coupled with the other parameters (i.e., when D_K is orthogonal to E_{K-1}), s_K is equal to 1. Thus the quantity $\varepsilon_o^2/|\partial\hat{Y}/\partial a_K|^2$ may be interpreted as an "uncoupled error." The effect of the closeness is to increase the error in the parameter estimate as appears clearly in relation (B8).

It is interesting to compare expression (B8) to the classical least squares result for the variance of the parameters (ref. 2), obtained also in quasi-linearization methods (ref. 12):

$$E(\delta a_K^2) = (DER^T DER)_{kk}^{-1} E(\delta Y^2) \tag{B9}$$

$(DER^T DER)_{kk}^{-1}$ is the *kth* diagonal element of the matrix $(DER^T DER)^{-1}$ and it can be shown easily that it is just equal to $1/(s_K^2|\partial\hat{Y}/\partial a_K|^2)$. But, whereas (B7) supposes a white ergodic process for δY, our result assumes only the existence of some kind of bound on the measurement error.

APPENDIX C

DEFINITION AND PROPERTIES OF THE OPTIMAL BASIS

Given a set of vectors belonging to a K-dimensional space E_K, K vectors of this set $D_1, D_2, \ldots, D_K$ are said to form an optimal basis in E_K with respect to the threshold s_o if they have the following properties:

For any i (i from 1 to K - 1) and any j $(i + 1 \leqslant j \leqslant K)$:

OB1 $$S(D_j/B_i) > s_o$$

OB2 $$S(D_{i+1}/B_i) \geqslant S(D_j/B_i)$$

where B_i is defined by

$$B_i = D_1 \cup D_2 \cup D_3 \ldots \cup D_i$$

Given a set of N vectors $(D_1, D_2, \ldots, D_N)$ and a threshold s_o it is possible to construct an optimal basis using the recursive process:

(a) To start the process, an arbitrary vector, D_1, is chosen from the set.

(b) The following steps are then taken:

Step 1 $$B_1 = D_1$$

D_2 is the vector of the set such that $S(D_2/B_1) \geqslant S(D_i/B_1)$

$\forall i \neq 1,2$

$$\vdots \qquad \vdots$$

Step j $$B_j = B_{j-1} \cup D_j$$

D_{j+1} is the vector of the set such that $S(D_{j+1}/B_j) \geqslant S(D_i/B_j)$

$\forall i \neq 1, 2, \ldots, j + 1$

$$\vdots \qquad \vdots$$

(c) At each step j, D_i is discarded from the set if $S(D_i/B_j) \leqslant s_o$.

It is clear that (c) leads to the property OB1 and (b) to OB2. Since some vectors might have been discarded, K is less than or equal to N.

We shall now derive some properties of the optimal basis from its definition and the properties of the linear closeness shown in appendix B.

OB3- In an optimal basis, the vectors are sorted by decreasing values of their separation from the preceding subbasis:

$$S(D_{i+1}/B_i) \leqslant S(D_i/B_{i-1})$$

Proof:

$$S(D_{i+1}/B_i) = S(D_{i+1}/B_{i-1} \cup D_i) \leqslant S(D_{i+1}/B_{i-1}) \qquad \text{(from LC1)}$$

and

$$S(D_{i+1}/B_{i-1}) \leqslant S(D_i/B_{i-1}) \qquad \text{(from OB2)}$$

OB4- In any subbasis B_j

$$S(D_{j-1}/B_{j-2} \cup D_j) \geqslant S(D_j/B_{j-1})$$

Proof: From LC3 and $S(D_i/B_{j-2} \cup D_{j-1}) = S(D_i/B_{i-1})$

$$S(D_{j-1}/B_{j-2} \cup D_j) = S(D_j/B_{j-1}) \times \frac{S(D_{j-1}/B_{j-2})}{S(D_j/B_{j-2})}$$

The result follows because the fraction is larger than 1 because of OB2.

APPENDIX D

COMPUTATION TECHNIQUE USED IN THE DEPENDENCE ANALYSIS

This appendix describes the elementary operations performed by the computer when executing the dependence analysis program whose FORTRAN IV version is given in appendix E.

Given the N (M-components) vectors $\partial\hat{Y}/\partial a_i$, first the normalized vectors are computed

$$D_i = (\partial\hat{Y}/\partial a_i)/|\partial\hat{Y}/\partial a_i|$$

Then the elements g_{ij} of the Gram matrix G corresponding to the N vectors D_i are computed

$$g_{ij} = D_i^T D_j$$

(All the diagonal elements are unity and the determinant of this matrix is DET(G).) Consider now the operations:

1. Define the quantity

$$DET_1 = g_{11} = 1$$

Multiply row 1 of the determinant successively by the elements of column 1 and subtract from the corresponding rows to obtain zeroes in column 1. The Gram determinant becomes:

$$DET(G) = \begin{vmatrix} 1 & g_{12} & g_{13} & \cdots & g_{1N} \\ 0 & g'_{22} & g'_{23} & & g'_{2N} \\ 0 & g'_{32} & g'_{33} & & g'_{3N} \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ 0 & g'_{N2} & g'_{N3} & & g'_{NN} \end{vmatrix}$$

with

$$g'_{ij} = g_{ij} - g_{i1}g_{1j} \qquad i,j = 2,N$$

Now divide row 2 by g'_{22} which is classically called the "pivot." Call it p_2.

2. Define

$$DET_2 = p_2 \times DET_1 = g'_{22}$$

The determinant is now

$$DET(G) = p_2 \times \begin{vmatrix} 1 & g_{12} & g_{13} & \cdots & g_{1N} \\ 0 & 1 & g''_{23} & \cdots & g''_{2N} \\ 0 & g'_{32} & g'_{33} & \cdots & g'_{3N} \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ 0 & g'_{N2} & g'_{N3} & \cdots & g'_{NN} \end{vmatrix}$$

$$g''_{2j} = g'_{2j}/p_2 \qquad (j = 3,N)$$

Now obtain zeroes below the diagonal element in column 2. The new g_i are

$$g'''_{ij} = g'_{ij} - g'_{i2}g''_{2j} \qquad i,j = 3,N$$

The new pivot is now $p_3 = g'''_{33}$. Row 3 is then divided by p_3.

3. Define

$$DET_3 = p_3 \times DET_2$$

After K steps like those just described, one has:

$$DET_K = p_K \times DET_{K-1} = p_K \times p_{K-1} \cdots p_3 \times p_2 \times 1 \tag{D1}$$

and the determinant is

$$\mathrm{DET}(G) = 1 \times p_2 \times p_3 \; . \; . \; . \times p_{K-1} \times p_K \times \left| \begin{array}{ccccccc} 1 & x & \cdots & x & x & \cdots & x \\ 0 & 1 & \cdots & x & x & \cdots & x \\ \vdots & & & & & & \\ 0 & 0 & & 1 & x & \cdots & x \\ 0 & 0 & & x & x & \cdots & x \\ \vdots & & & & & & \\ 0 & 0 & \cdots & x & x & \cdots & x \end{array} \right| \begin{array}{l} \left.\begin{array}{c} \\ \\ \\ \\ \end{array}\right\} K \text{ rows} \\ \\ \\ \\ \end{array}$$

where x is used just to indicate the location of nonzero elements.

Obviously, if there were only K rows and K columns (corresponding to K vectors D_i), DET(G) would just equal DET_K. Therefore DET_i in general is the value of the Gram determinant of the first i vectors. Since the Gram determinant is equal to the square of the volume constructed with the vectors, we may write

$$\mathrm{DET}_K = V(B_K)^2 = V(B_{K-1})^2 \times S(D_K/B_{K-1})^2$$

$$\mathrm{DET}_{K-1} = V(B_{K-1})^2$$

Compared with equation (D1)

$$p_K = S(D_K/B_{K-1})^2$$

This shows that the pivotal element is precisely the square of the separation of D_K from the preceding subbasis B_{K-1}.

The construction of an optimal basis requires that, at each step, the largest diagonal element be chosen as pivot. In general the sequence of vectors will not be 1, 2, 3, . . ., N as described before, but it is always possible at each step, to exchange rows and columns to build up the triangular form. In practice, however, such an exchange is time consuming and will not really be performed by the computer. It is replaced with advantage by a bypassing process. Indeed, once a diagonal element has been chosen as pivot, say p_i, and the corresponding row normalized, row and column i will not be changed any more. To memorize this fact that the elements of row and column i must not be modified, a logical variable associated to the index i is set. The general formula, by which the new value of the elements g_{ij} is obtained, may be expressed in the general form

$$(g_{ij})_{\text{new}} = (g_{ij})_{\text{old}} - g_{i \text{ next}} \times g_{\text{next } j}$$

where "next" is the index of the pivotal element. Each time this formula is to be applied, the indices i and j are checked and this computation is bypassed if row i or j has been already normalized. Also, when the next pivotal element is searched, all the pivot values are examined from 1 to N except those already used since the same pivotal row cannot be used twice.

In the same way, whenever a row, of index i say, is discarded (corresponding to the parameter a_i), it is memorized and all computations involving this index are bypassed. This is also done, right from the beginning, for the irrelevant or unused parameters.

As we have seen previously, if a diagonal element, say g_{KK}, is found smaller than the square of the threshold s_o, the corresponding vector D_K has to be discarded; that is, row and column K are not considered any more in the computation process. In this case a special procedure is started to determine the subset of B_{K-1} to which D_K is linearly close. This is done by computing the components μ_i of D_K upon the K - 1 vectors belonging to B_{K-1}. The following equation has to be solved for the μ_i:

$$\sum_i \mu_i D_i = D_K \qquad i = 1, K - 1$$

Multiplying both sides by D_j^T gives the equivalent system:

$$\sum_i \mu_i g_{ji} = g_{jK} \qquad i \text{ and } j = 1, K - 1$$

It can be shown that an equivalent system is obtained by transforming g_{ij} with the same rules used to reduce the determinant. Since the g_{ij}, at the step K, form an upper triangle, it is then a simple matter of backward elimination to obtain the unknown μ_i.

In the same way, the solution of any system of the form

$$\sum_i g_{ij}\ da_i' = c_j \qquad (i \text{ and } j = 1, N \textit{ and } MN(i), MN(j) \neq 0)$$

where the da_i' are the unknowns and c_j are given quantities, can be solved be incorporating the c_j as an extra column to the Gram matrix and performing the corresponding transformations. In this case, however, the pivot will never be chosen in this extra column, and the backward elimination is performed once the Gram determinant is completely transformed in its triangular form. If c_j are precisely chosen as the components of the vector D^T ER, the da_j' are the components of the vector dA' solution of the matrix equation

$$(D^T D)dA' = D^T ER$$

It is then easy to obtain the solution dA of equation (5),

$$(\mathrm{DER}^T\ \mathrm{DER})\mathrm{dA} = -\mathrm{DER}^T\ \mathrm{ER}$$

which is simply

$$da_i = -da_i'/|\partial\hat{Y}/\partial a_i| \qquad (i = 1,N \text{ and } MN(i) \neq 0)$$

$$da_i = 0 \qquad \text{if } MN(i) = 0$$

APPENDIX E

A FORTRAN IV PROGRAM FOR THE DEPENDENCE ANALYSIS

This program has been written as a subroutine activated by the statement

```
CALL COR(N1, N3, THR, MODE)
```

N1	number of parameters of the model	$1 \leqslant \mathrm{N1} \leqslant 36$
N3	total number of output measurements	$1 \leqslant \mathrm{N3} \leqslant 2400$
THR	value of the threshold	$0 \leqslant \mathrm{THR} \leqslant 1$

MODE integer controlling the output; may take values from 0 to 6 (0 prints nothing, 6 prints out the full information, between 0 and 6 partial results are given). If 10 is added (i.e., from 10 to 16) the inversion of the Gram matrix is also performed.

The other variables needed as input or output for this subroutine are passed via labeled COMMON's which are used throughout the whole identification program.

Input Variables

MNS controls the parameters to be actually used in the identification: MNS(I) = 0 if the user does not wish to identify the parameter a_I. MNS(N1 + 1) determines the first vector to start the analysis. If set equal to 0, the subroutine will choose the first vector.

DER contains the vectors $\partial\hat{Y}/\partial a$ (DER(j,i) = $\partial\hat{Y}_j/\partial a_i$)

ER contains the error vector (ER(j) = $\hat{Y}_j - Y_j$)

MN is used as input, only to indicate the irrelevance of a parameter (MN(i) = 0) if this has been found during the computation of the DER's

Output Variables (Returned to the Calling Program)

C contains the elements of the normalized Gram matrix, optionally will contain the elements of the inverse if MODE $\geqslant$ 10

PIV contains the values of the pivots

AD contains the norms of the vectors $\partial\hat{Y}/\partial a$

DDA solution vector of the equation $(\mathrm{DER}^T\ \mathrm{DER})\mathrm{DDA} = \mathrm{DER}^T\ \mathrm{ER}$

MN magic number used to control the identification (dependency index):
MN(i) = 0 means a_i is dependent (or irrelevant) and will not be identified
MN(i) = 1 means a_i is dependent but will be identified
MN(i) = 2 means a_i is independent and will be identified

Output Variables Used in the Print-Out Only

These variables are not returned to the calling program but are printed out at different steps of the computation. Therefore some may have different meanings in different parts of the program. The meanings given below correspond to those they have in WRITE statements, in the order they appear in the program.

NEXT index of the next basic vector found to build the optimal basis

SAVE separation of this vector from the preceding subbasis

IOUT index of the critical vector (also, a fortiori, index of a discarded one)

SMIN separation of the critical vector

SEPAR separation of a discarded vector in a dependent set

G(K,K) error in the dependent parameter a_K corresponding to a unit error in the discarded parameter of the dependent set

NSETC index of a dependent set after union has been performed

DET value of the determinant of the normalized Gram matrix after removal of discarded vectors

TIME value of the computation time

WR array used to write the names of parameters involved in different comments issued by the subroutine and is obtained as output of the auxiliary subroutine TRADUC

Auxiliary Variables

G contains the elements of the normalized Gram matrix during the successive transformations into an upper triangle; G is identical to C at the beginning

CLEAR logical array: CLEAR(i) = .TRUE. if the vector D_i has already been used in the optimal basis (or discarded)

IBASIS IBASIS(i) is the index of the i*th* vector of the basis

KG dimension of a subbasis

SET logical array representing the different dependent subset: SET(i,j) = .TRUE. indicates that the parameter a_i belongs to the dependent set number j

IR logical array: IR(i) = .TRUE. indicates that a_i is irrelevant

Other variables are used as running subscripts or transitory storage and there is no need to catalog them here.

Auxiliary Subroutine TRADUC(FOUND, N1, K)

This subroutine is used to translate logical information into parameter names in order to ease the reading of the comments. It is a part of the formatting of the output. The names of the parameters are given through the COMMON/WRITE/, in the array WA. The output of the subroutine is found in the array WR of the same COMMON.

General Organization of the Subroutine COR

COR has been divided for convenience into seven sections or logical units.

Section 1- This section computes the norms AD, the normalized Gram matrix C and the vector (D^T)(ER) that is originally stored in the array DDA. It also chooses the first basis vector as being the closest to ER (the corresponding parameter would thus give the best fit if all the others had to be discarded). This choice will be ignored, however, if MNS(N1 + 1) is not zero. Control will be returned to the calling program at this point if MODE = -1.

Section 2- After some formatting the matrix C can be written optionally and initialization of the variables performed.

Section 3- The reduction of the G matrix to an upper triangle is performed along with the transformation of the vector DDA. Transformation of columns of the matrix C is optional. At each step, the diagonal elements of G are searched for the biggest and the smallest. The latter is compared to the square of the threshold (S = THR^2). In case of dependence, control is transferred to section 4. Otherwise, basic and critical parameters are printed out with their separations. The pivotal row and the corresponding component of DDA are normalized (optionally the corresponding row of C) and the next step is started. When all the vectors have been used (IREMN = 0) control is transferred to section 5.

Section 4- The dependent parameter is discarded (MN = 0) and control returns to section 3 if MODE = 0. Otherwise a backward elimination is performed to obtain the components of the discarded vector on the subbasis and the dependent subset stored in SET. Control is returned to section 3.

Section 5- The solution vector DDA is finally obtained by backward elimination (optionally the inverse of C).

Section 6- When dependent sets have been found, they are printed out with their separations and the error in dependent parameters is computed and printed. Then the dependent sets are united eventually (if MODE.GT.3).

Section 7- In this last section some logical and formatting manipulations are performed and final comments are delivered.

Remark: The analysis of the G matrix and the inversions are all performed in a double precision arithmetic.

A listing of the program is given in the following pages.

```
      SUBROUTINE COR(N1,N3,THR,MODE1)
      COMMON/BASIS/IBASIS(36),PIV(36),G(36,36)
      COMMON/NVRT/AD(36),DDA(36),C(36,36)
      COMMON/WRITE/WR(36),WA(36)
      COMMON/PRM/MN(37),MNS(37),SENSIV(36),A(36),AMIN(36),AMAX(36)
     1,DA(36),ER(2400),DER(2400,36)                                     PRM
      DIMENSION SEPAR(18),SET(18,18),CLEAR(36),CLEARB(36),IR(36)
      DOUBLE PRECISION AD,DDA,G,C,GRAD,S,SAVE,SMAX,SMIN,DET
      LOGICAL *1 CLEAR,CLEARB,SET,IR,INV
      DATA PEQ,COMMA/3H' =,1H,/
      CALL CLOCK(ISTART)
      MODE = MODE1
C *SECTION 1* ------------COMPUTATION OF THE GRAM MATRIX---------------------
      SMAX = 0.D0
      DO 500 I=1,N1
      AD(I) = 0.D0
      IF(MN(I).EQ.0)         GO TO 2500
 1503 DO 503 K=1,N3
  503 AD(I) = AD(I) + DBLE(DER(K,I))**2
      IF(AD(I).LT.1.D-70)     MN(I) = 0
      AD(I) = DSQRT(AD(I))
 2500 DO 500 J=1,N1
  500 C(I,J) = 0.D0
 1501 DO 501 I=1,N1
      IF(MN(I).EQ.0)         GO TO 501
 1502 DO 502 J=1,I
      IF(I.EQ.J)             GO TO 502
      IF(MN(J).EQ.0)         GO TO 502
 1504 DO 504 K=1,N3
  504 C(I,J) = C(I,J) + DBLE(DER(K,I))*DBLE(DER(K,J))
      C(I,J) = C(I,J)/(AD(I)*AD(J))
      C(J,I) = C(I,J)
C   DEFINE THE FIRST BASIC PARAMETER
      IF(SMAX.GT.DABS(C(I,J)) ) GO TO 502
      SMAX = DABS(C(I,J))
      NEXT = I
  502 CONTINUE
      C(I,I) = 1.D0
      GRAD = 0.D0
 1510 DO 510 K=1,N3
  510 GRAD = GRAD + DBLE(DER(K,I))*DBLE(ER(K))
      DDA(I) = GRAD/AD(I)
  501 CONTINUE
      N2 = N1 + 1
      IF(MNS(N2).NE.0)         NEXT = MNS(N2)
```

```
C *SECTION 2*----------------INITIALISATION---------------------------------
      IF(MODE.EQ.-1)           RETURN
      INV = MODE1.GE.10
      IF(INV)      MODE = MODE1-10
      MN(N2) = 1
      IF(MODE.EQ.0)            GO TO 507
      IF(N1.GT.15)             GO TO 507
  505 WRITE(6,6505)
 6505 FORMAT(  1H1,10X,'PARAMETERS CORRELATION MATRIX')
 1506 DO 506 I=1,N1
  506 WRITE(6,6506) WA(I),(C(I,J),J=1,I)
 6506 FORMAT(/1X,A4,2X,15F8.4)
  507 IF(MODE.LT.4)            GO TO 1508
      WRITE(6,601) WA(NEXT)
  601 FORMAT('1   BASIC   ',30X,' CRITICAL'/'   PARAMETER',5X,'SEPARATION'
     1,15X,'PARAMETER',5X,'SEPARATION'/4X,A4/)
C-----INITIALISE
 1508 DO 508 I=1,N1
 1509 DO 509 J=1,N1
      IF(I.LE.18.AND.J.LE.18)     SET(I,J) = .FALSE.
      G(I,J) = C(I,J)
  509 C(I,J) = 0.D0
      C(I,I) = 1.D0
      IR(I) = MN(I).EQ.0
      MN(I) = ((MN(I)+1)/2)*2
      CLEAR(I) = .FALSE.
      IF(.NOT.IR(I))           GO TO 508
      DDA(I) = 0.D0
      CLEAR(I) = .TRUE.
  508 MN(N2) = MIN0(MN(N2),MN(I))
      DET = 1.D0
      S = THR**2
      KG = 0
      NSET = 0
      PIV(NEXT) = 1.
```

```
C *SECTION 3*---------ANALYSIS OF THE VECTOR SET------------------------------
C THE NEW BASIS VECTOR IS NEXT
    1 CLEAR(NEXT) = .TRUE.
      IF(IR(NEXT))           GO TO 2004
      KG = KG + 1
      IBASIS(KG) =  NEXT
      IF(KG.GE.N1)           GO TO 20
C COMPUTE THE NEW ARRAY OF REMNANT VECTORS
 1002 DO 2 J=1,N1
      IF(CLEAR(J))           GO TO 2
C      TRANSFORM INPUT VECTOR DDA
      DDA(J) = DDA(J) - DDA(NEXT)*G(J,NEXT)
      IF(.NOT.INV)           GO TO 1003
 1203 DO 203 I=1,N1
  203 C(J,I) = C(J,I) - C(NEXT,I)*G(J,NEXT)
 1003 DO 3 I=1,J
      IF(CLEAR(I))           GO TO 3
      G(J,I) = G(J,I) - G(J,NEXT)*G(NEXT,I)
      G(I,J) = G(J,I)
    3 CONTINUE
    2 CONTINUE
C FIND THE DEPENDENT, THE OPTIMAL AND THE CRITICAL VECTORS
 2004 IREMN = 0
      SMAX = 0.D0
      SMIN = 1.D0
 1004 DO 4 I=1,N1
      IF(CLEAR(I))           GO TO 4
      IREMN = IREMN + 1
      SAVE = G(I,I)
      IF(SAVE.GT.SMIN)       GO TO 5
C STORE THE CRITICAL
      SMIN = SAVE
      IOUT = I
    5 IF(SAVE.LE.SMAX)       GO TO 4
C STORE THE OPTIMAL
      SMAX = SAVE
      NEXT = I
    4 CONTINUE
      IF(IREMN.EQ.0)         GO TO 20
C CHECK THE DEPENDENT
      IF(SMIN.LE.S)          GO TO 8
C COMPUTE THE VALUE OF THE DETERMINANT
      PIV(NEXT) = SMAX
      DET = DET*SMAX
C NORMALISE ROW AND COLUMN NEXT BEFORE THE NEW CYCLE
      SAVE = DSQRT(SMAX)
      IF(MODE.LT.4)          GO TO 1007
      SMIN = DSQRT(SMIN)
      WRITE(6,600) WA(NEXT),SAVE,WA(IOUT),SMIN
  600 FORMAT(4X,A4, 8X,1PE9.2,18X,A4, 8X,E9.2)
 1007 DO 7 I=1,N1
      IF(INV)     C(NEXT,I) = C(NEXT,I)/SMAX
      IF(CLEAR(I))           GO TO 7
      G(NEXT,I) = G(NEXT,I)/SMAX
    7 CONTINUE
      DDA(NEXT) = DDA(NEXT)/SMAX
                             GO TO 1
```

```
C *SECTION 4*----------ANALYSIS OF THE DEPENDENCE WITHIN THE SUB-BASIS-------
    8 NSET = NSET + 1
      DDA(IOUT) =0.D0
      MN(IOUT) = 0
      CLEAR(IOUT) = .TRUE.
      IF(MODE.EQ.0)           GO TO 2004
      SET(IOUT,NSET) = .TRUE.
C FIND THE COMPONENTS OF IOUT ON THE BASIS
 1009 DO 9 I=1,KG
      KBACK = KG+1-I
      K = IBASIS(KBACK)
      IF(I.EQ.1)              GO TO 9
 1010 DO 10 J=2,I
      KBACK1 = KG-I+J
      K1 = IBASIS(KBACK1)
   10 G(K,IOUT) = G(K,IOUT) - G(K,K1)*G(K1,IOUT)
    9 CONTINUE
 1011 DO 11 I=1,KG
      K = IBASIS(I)
      GRAD = SMIN + (G(K,IOUT)**2)*PIV(K)
      IF(DABS(GRAD).LE.S)     GO TO 11
C MEMORISE THE DEPENDENT VECTOR IN THE LOGICAL ARRAY ''SET''
      SET(K,NSET) = .TRUE.
C SET THE MAGIC NUMBER TO 1 FOR DEPENDENCE OF THE NEXT BASIS VECTOR WITH IOUT
      MN(K) = 1
   11 CONTINUE
C MEMORISE THE SEPARATION OF THE DEPENDENT VECTOR IN THE ARRAY ''SEPAR''
      SEPAR(NSET) = DSQRT(DMAX1(0.D0,SMIN) )
                              GO TO 2004
C *SECTION 5* -----SOLVE THE EQUATION G*DDA = D'*ER    ( INVERT G EVENTUALLY)-
   20 IF(KG.EQ.1)             GO TO 1121
 1120 DO 120 I=2,KG
      KBACK = KG+1-I
      K = IBASIS(KBACK)
 2120 DO 120 J=2,I
      KBACK1 = KG-I+J
      K1 = IBASIS(KBACK1)
      IF(.NOT.INV)            GO TO 120
 1220 DO 220 L1 = 1,KG
      L = IBASIS(L1)
  220 C(K,L) = C(K,L) - G(K,K1)*C(K1,L)
  120 DDA(K) = DDA(K) - G(K,K1)*DDA(K1)
 1121 DO 121 I=1,KG
      K = IBASIS(I)
  121 DDA(K) = DDA(K)/AD(K)
```

```
C *SECTION 6* -----------------FIND THE STRUCTURE OF THE SET-----------------
      IF(MODE.EQ.0)           GO TO 9999
      IF(NSET.EQ.0)           GO TO 24
      IF(MODE.LT.5)           GO TO 24
C WRITE THE DEPENDENT SETS
      WRITE(6,621)
  621 FORMAT(/11X,'DEPENDENT SETS OF PARAMETERS'/ 1X,'SEPARATION'/)
 1022 DO 22 J=1,NSET
 1023 DO 23 I=1,N1
      IF(MN(I).EQ.0.AND.SET(I,J) )      IOUT = I
   23 CLEAR(I) = SET(I,J)
      CALL TRADUC(CLEAR,N1,NW)
      WRITE(6,622) SEPAR(J), (WR(I),COMMA,I=1,NW)
  622 FORMAT(    1X,E9.2/(10X,24(A4,A1 )/))
      IF(MODE.LT.6)           GO TO 22
      K = 0
 1012 DO 12 I=1,N1
      IF(MN(I).EQ.0)          GO TO 12
      IF(.NOT.SET(I,J) )   GO TO 12
      K = K + 1
      G(K,K)     = -G(I,IOUT)*AD(IOUT)/AD(I)
      WR(K) = WA(I)
   12 CONTINUE
   14 WRITE(6,614) WA(IOUT),PEQ,(WR(I),PEQ,G(I,I),I=1,K)
  614 FORMAT(/' IF ',A4,A3,' 1, THEN',6(3X,A4,A3,1PE9.2)/
     1(19X,6(3X,A4,A3,E9.2)/))
   22 CONTINUE
   24 WRITE(6,624) N1,THR,(MN(K),K=1,N1)
  624 FORMAT('1 ***THERE ARE',I3,' PARAMETERS IN THIS PROBLEM .
     1THE SEPARATION THRESHOLD WAS',E9.2  /' MAGIC NUMBER ',36I2/)
      IF(NSET.EQ.0)           GO TO 32
      IF(MODE.LT.3)           GO TO 32
C*********************** UNION OF THE DEPENDENT SETS ***********************
 1125 DO 125 I=1,N1
  125 CLEAR(I) = .FALSE.
      NSETC = 0
 1025 DO 25 J=1.NSET
      IF(CLEAR(J))            GO TO 25
      NSETC = NSETC + 1
 1026 DO 26 K=J,NSET
      IF(CLEAR(K))            GO TO 26
 1027 DO 27 I=1,N1
      IF(SET(I,J).AND.SET(I,K))   GO TO 1028
   27 CONTINUE
                              GO TO 26
 1028 DO 28 I=1,N1
   28 SET(I,NSETC) = SET(I,J).OR.SET(I,K)
      SEPAR(NSETC) = AMIN1(SEPAR(J),SEPAR(K))
      CLEAR(K) = .TRUE.
   26 CONTINUE
   25 CONTINUE
```

```
C *SECTION 7* --------PREPARE AND WRITE THE FINAL DIAGNOSTIC-----------------
 1029 DO 29 J=1,NSETC
 1030 DO 30 I=1,N1
      CLEAR(I) = (MN(I).EQ.1).AND.SET(I,J)
   30 CLEARB(I) = (MN(I).EQ.0).AND.SET(I,J)
      CALL TRADUC(CLEAR,N1,NW)
      WRITE(6,631) J,SEPAR(J),(WR(I),COMMA,I=1,NW)
  631 FORMAT(//' ***DEPENDENT SET NUMBER',I3,'    ***SEPARATION =',E9.2/
     1' *'/' * A TRUE VALUE IS OBTAINED FOR  '/(' *',8X,24(A4,A1 )/) )
      CALL TRADUC(CLEARB,N1,NW)
      WRITE(6,629) (WR(I),COMMA,I=1,NW)
  629 FORMAT(' * IF IS KNOWN THE TRUE VALUE OF '/(' *',8X,24(A4,A1 )/) )
   29 CONTINUE
   32 IF(MODE.LT.2) RETURN
CWRITE THE NAME OF INDEPENDENT, IRRELEVANT, DROPPED AND UNUSED PARAMETERS
 1033 DO 33 I=1,N1
      CLEAR(I) = MN(I).EQ.2
   33 CLEARB(I) = IR(I).AND.(MNS(I).NE.0)
      CALL TRADUC(CLEAR,N1,NW)
      WRITE(6,634) (WR(I),COMMA,I=1,NW)
  634 FORMAT(//' *  INDEPENDENT PARAMETERS  '/(' *',2X,24(A4,A1 )/))
      CALL TRADUC(CLEARB,N1,NW)
      WRITE(6,635) (WR(I),COMMA,I=1,NW)
  635 FORMAT(' *'/' *  IRRELEVANT PARAMETERS  '/(' *',2X,24(A4,A1 )/))
 1036 DO 36 I=1,N1
      CLEAR(I) = MN(I).EQ.0.AND.MNS(I).NE.0
   36 CLEARB(I) = MNS(I).EQ.0
      CALL TRADUC(CLEAR,N1,NW)
      WRITE(6,637) (WR(I),COMMA,I=1,NW)
  637 FORMAT(' *'/' *  NOT ESTIMATED'/(' *',2X,24(A4,A1 )/))
      CALL TRADUC(CLEARB,N1,NW)
      WRITE(6,638) (WR(I),COMMA,I=1,NW)
  638 FORMAT(' *'/' *  NOT USED'/(' *',2X,24(A4,A1 )/))
      CALL CLOCK(IEND)
      TIME = (IEND-ISTART)/100.
      WRITE(6,606) DET,TIME
  606 FORMAT(///' DET=',D22.15,'    TIME =',F9.3,' SEC')
 9999 RETURN
      END
      SUBROUTINE TRADUC(FOUND,N1,K)
      COMMON/WRITE/WR(36),WA(36)
      DIMENSION FOUND(N1)
      LOGICAL *1 FOUND
      DATA BLANK,EMPTY/'    ','NONE'/
      K=0
 1001 DO 1 I=1,N1
      WR(I) = BLANK
      IF(.NOT.FOUND(I) )     GO TO 1
      K = K + 1
      WR(K) = WA(I)
    1 CONTINUE
      IF(K.GT.0)       RETURN
      K=1
      WR(1) = EMPTY
      RETURN
      END
```

EXAMPLE OF COMPUTER OUTPUT

To illustrate the capabilities of the program, two examples of output are given on figures 17 and 18. In the first example, a test matrix DER with known relationships between the columns was input to the subroutine COR to simulate a case of nonuniqueness where the model had 14 parameters (designated here by the symbolic names A_1, A_2, . . ., A_{14}). The threshold was set equal to 0.01 and A_{10} was discarded *a priori* (MNS(10) set to 0) and A_5 was found to be irrelevant. Figure 17 shows first the construction of the optimal basis. It indicates that the first parameter (chosen by the user) is A_2. Then A_4 is selected with a separation equal to 1 (indicating orthogonality between the vectors $\partial\hat{Y}/\partial A_2$ and $\partial\hat{Y}/\partial A_4$). At the same time A_3 is found to be critical with a separation of 0.347. The third parameter selected is A_8 with again a separation of 1, and A_3 is still critical. At this point, however, A_9 was found to be dependent and was discarded, but this is displayed later on. This process continues, the basis vectors becoming less and less orthogonal (property OB3 of the optimal basis shown in appendix C).

Some parameters are missing in the first column; indeed they have been discarded either during the analysis (as dependent parameters) or before (as irrelevant or discarded *a priori* by the user right from the beginning). As for the discarded parameters, they appear in the next display, where we are told that A_8 and A_9 are dependent with a separation of 0.0025. A_9 was discarded and its value will be unchanged. The value of A_8 that will be obtained in the identification is a function of the value of A_9, and the next line indicates that if the value of A_9 is increased by one unit (absolute variation - symbolized by a prime) then A_8 will decrease by 40 units. This information is very useful to estimate the error in A_8 because of the lack of knowledge of A_9, and also it can be used to correct *a posteriori* the value of A_8 if a better value has been obtained for A_9 from other measurements, without running the whole identification again.

Then another dependent set is found with three parameters, A_1, A_2, and A_3. The parameter A_3 was discarded and the next line indicates how a variation in A_3 will affect the values found for A_1 and A_2. The last dependent set includes four parameters and A_{14} was discarded. Note that A_8 appears again, therefore the first and the last set will be united in the final results.

The final results are given next on this output. Two dependent sets appear; A_9 and A_{14} have been discarded from the first and their values should be known in order to obtain correct values for A_8, A_{11}, and A_{13}. In the same way A_3 was discarded from the second set. Then the names of the independent and irrelevant parameters are displayed. Finally, four parameters have been discarded and will not be estimated in this identification, and also, the computer reminds us that A_{10} was not used at all because of the *a priori* decision to do so at the beginning of the identification. The last piece of information concerns the value of the determinant of the reduced Gram matrix (that is after suppression of rows and columns corresponding to the discarded parameters) and, gives the computation time in seconds. All of these results agreed exactly with the known initial data.

Another example is shown on figure 18. It corresponds to the real case (case A) of the platform identification, when the reaction wheels are not used (natural damping). The dependence between the parameters Jxz, Jx, Jz and a_{13} is found and the dependence coefficients are denoted by the "prime" following the name, so that the last line of the first set of comments should read: *if Jxz is incremented by one unit, then the value obtained for Jx will change by an amount of 2.55 10^{-1} unit, Jz by 4.0 units and a_{13} will be decreased by 4.83 10^{-6} unit.* Note that the analysis of this 13 parameter problem only required 2 sec of computation time.

REFERENCES

1. Deutsch, Ralph: Estimation Theory. Prentice-Hall, Inc., 1965.

2. Smith, Gerald L.: On the Theory and Methods of Statistical Inference. NASA TN R-251, 1967.

3. Taylor, Lawrence W., Jr.; Iliff, Kenneth W.; and Powers, Bruce G.: A Comparison of Newton-Raphson and Other Methods for Determining Stability Derivatives From Flight Data. AIAA Paper 69-315, 1969.

4. Kalman, R. E.; Falb, P. L.; and Arbib, M. A.: Topics in Mathematical System Theory. Part 4: Advanced Theory of Linear Systems. McGraw-Hill, 1969.

5. Mehra, Raman K.: Identification of Stochastic Linear Dynamic Systems. Proc. IEEE 1969 Symposium on Adaptive Processes, Univ. Park, Pa., Nov. 1969.

6. Denery, Dallas G.: An Identification Algorithm Which is Insensitive to Initial Parameter Estimates. AIAA Paper 70-34, 1970.

7. Ho, B. L.: An Effective Construction of Realizations From Input/Output Descriptions. Ph.D. Thesis, Stanford, 1966.

8. Bryson, A. E., Jr.; and Ho, Y. C.: Applied Optimal Control. Blaisdell Pub. Co., Waltham, Mass., 1969.

9. Shilov, Georgij E. (R. A. Silverman, trans.): An Introduction to the Theory of Linear Spaces. Prentice-Hall Inc., 1961, p. 169 ff.

10. Moran, Francis J.; and Dishman, Bruce H.: Analysis and Verification of a Spacecraft Wide-Angle Attitude-Control System. NASA TN D-5716, 1970.

11. Meyer, George: On the Use of Euler's Theorem on Rotations for the Synthesis of Attitude Control Systems. NASA TN D-3643, 1966.

12. Schalow, R. D.: Quasilinearization and Parameter Estimation Accuracy. Ph.D. Thesis, Syracuse Univ., 1967.

TABLE 1.- RESULTS OF THE IDENTIFICATION FOR TWO DIFFERENT IDENTIFICATION RUNS IN CASE A (FREE OSCILLATION CASE)

Name	Value	Error bounds	Units	Sensitivity	Dependency index
Run 1: Final results when no parameter is discarded (threshold = 0)					
J_x	1.23 10^2	±1.6	slug-ft^2	12.4	2
J_y	1.22 10^2	±2.7	↓	6.7	2
J_z	1.60 10^2	±19	↓	1.3	2
J_{xy}	-3.24	±1.4	↓	.32	2
J_{xz}	9.64	±5	↓	.54	2
J_{yz}	-1.31	±5.4	↓	.036	2
DMP_x	3.25 10^{-2}	±2.2 10^{-2}	ft-lb-sec	.037	2
DMP_y	3.09 10^{-2}	±2.8 10^{-2}	↓	.018	2
DMP_z	1.19 10^{-1}	±6.9 10^{-2}	↓	.016	2
b_y	9.87 10^{-3}	±1.8 10^{-3}	ft-lb	.057	2
b_z	7.77 10^{-1}	±7.6 10^{-3}	↓	12.5	2
a_{13}	-2.53 10^{-4}	±2.8 10^{-5}	↓	.13	2
Run 2: Final results when J_{xz} is discarded by the dependence analysis (threshold = 0.02)					
J_x	1.18 10^2	±1.3	slug-ft^2	12.7	1
J_y	1.20 10^2	±1.8	↓	7.7	2
J_z	1.21 10^2	±7.9	↓	1.06	1
J_{xy}	-1.70	±1.2	↓	.17	2
J_{xz}	-1.67	0	↓	.091	0
J_{yz}	2.44 10^{-2}	±2.6	↓	.00085	2
DMP_x	2.78 10^{-2}	±1.6 10^{-2}	ft-lb-sec	.032	2
DMP_y	5.14 10^{-2}	±2.5 10^{-2}	↓	.030	2
DMP_z	8.03 10^{-2}	±3.5 10^{-2}	↓	.014	2
b_y	1.00 10^{-2}	±1.5 10^{-3}	ft-lb	.055	2
b_z	7.73 10^{-1}	±6.8 10^{-3}	↓	12.5	2
a_{13}	-1.89 10^{-4}	±1.4 10^{-5}	↓	.15	1

TABLE 2.- RESULTS OF THE IDENTIFICATION IN CASE B (REACTION WHEELS DAMPING)

Name	Value	Error bounds	Units	Sensitivity	Dependency index
J_x	119	±5.3	slug-ft^2	1.32	2
J_y	125	±5		1.7	2
J_z	213	±130		.02	2
J_{xy}	4.7	±2.4		.2	2
J_{xz}	-8.0	±2.7		.03	2
J_{yz}	-6.1	±2.6		.02	2
α_1	67	±4	ft-lb-sec	1.4	2
α_2	121	±3.5		.7	2
α_3	147	±16		.13	2
b_y	$-3.4\ 10^{-3}$	$\pm 1.10^{-3}$	ft-lb	.09	2
b_z	-.76	$\pm 1.3\ 10^{-2}$		3.7	2
a_{13}	$-1.0\ 10^{-3}$	$\pm 3.10^{-4}$		.2	2
a_{16}	.137	$\pm 2.10^{-2}$		.38	2
a_{17}	.135	$\pm 7.10^{-3}$		1.2	2
a_{18}	.122	0		.0	0
c_1	$4.24\ 10^{-2}$	$\pm 1.3\ 10^{-2}$	cps	.13	2
c_2	$3.30\ 10^{-2}$	$\pm 1.3\ 10^{-2}$		.13	2
c_3	$4.40\ 10^{-2}$	$\pm 1.8\ 10^{-2}$		.14	2
a_{22}	1.36	$\pm 2.10^{-2}$	ft-lb-sec	4.5	2
a_{23}	1.35	$\pm 3.10^{-2}$		4.0	2
a_{24}	1.00	0		0	0

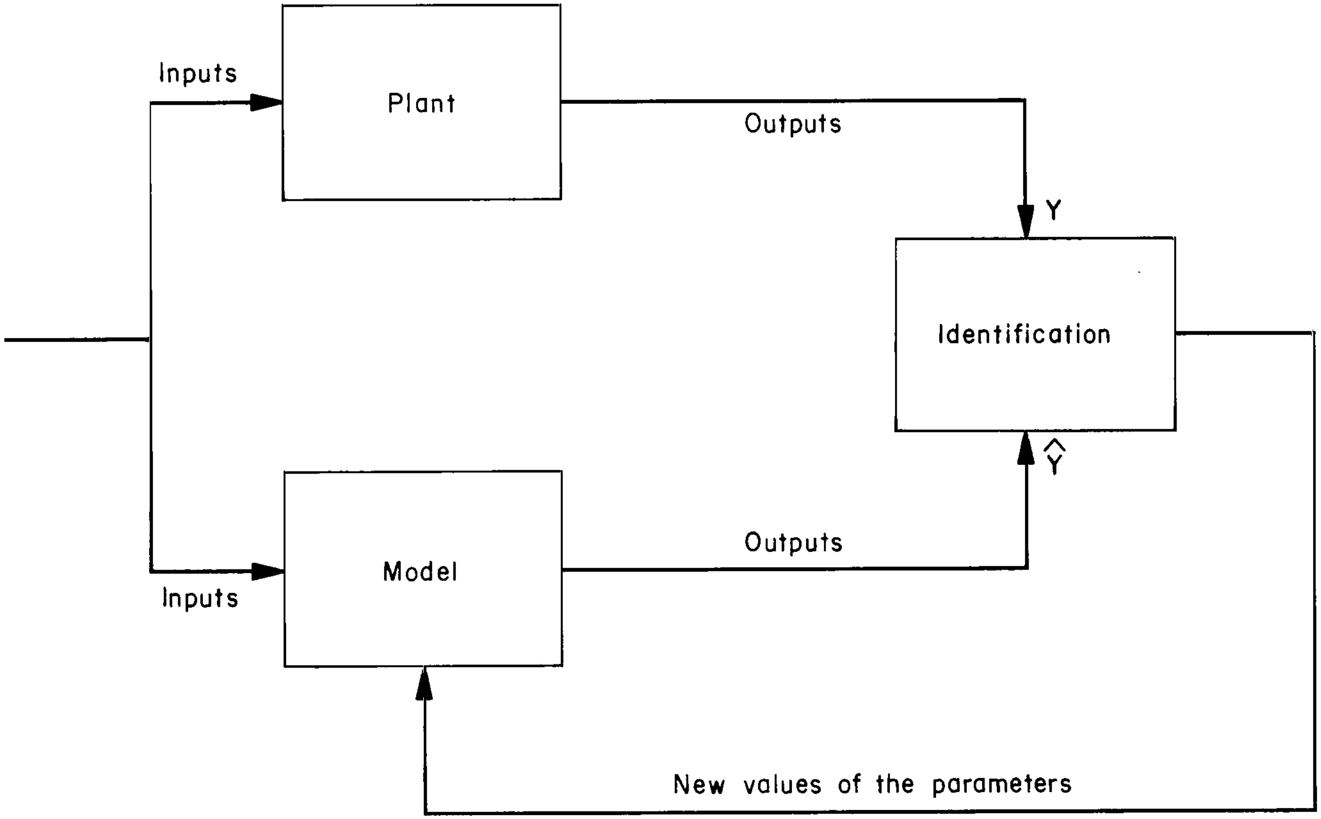
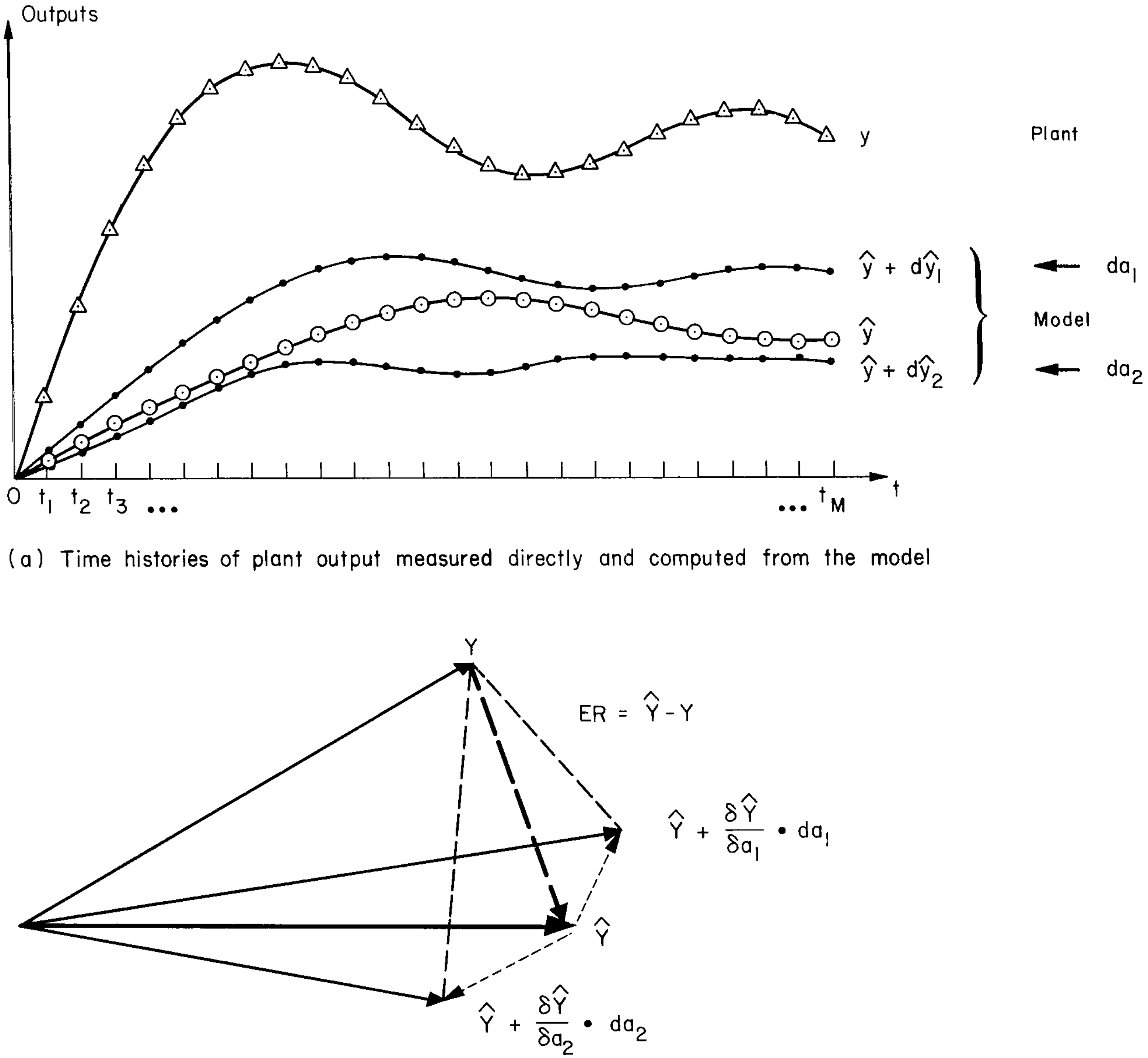


Figure 1.- General scheme of the identification process.



(a) Time histories of plant output measured directly and computed from the model

(b) Vectorial representation of the time histories and perturbation process

Figure 2.- Vectorial representation of the time histories and perturbation process.

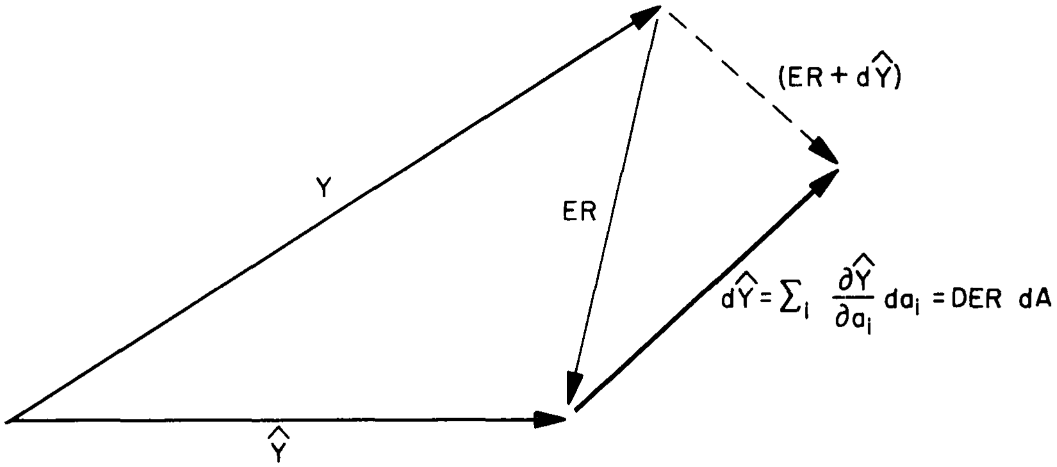

$(ER + d\hat{Y})$
Y
ER
$d\hat{Y} = \sum_i \frac{\partial \hat{Y}}{\partial a_i} da_i = DER\ dA$
$\hat{Y}$

Figure 3.- Geometric interpretation of the parameters adjustment in a least squares method.

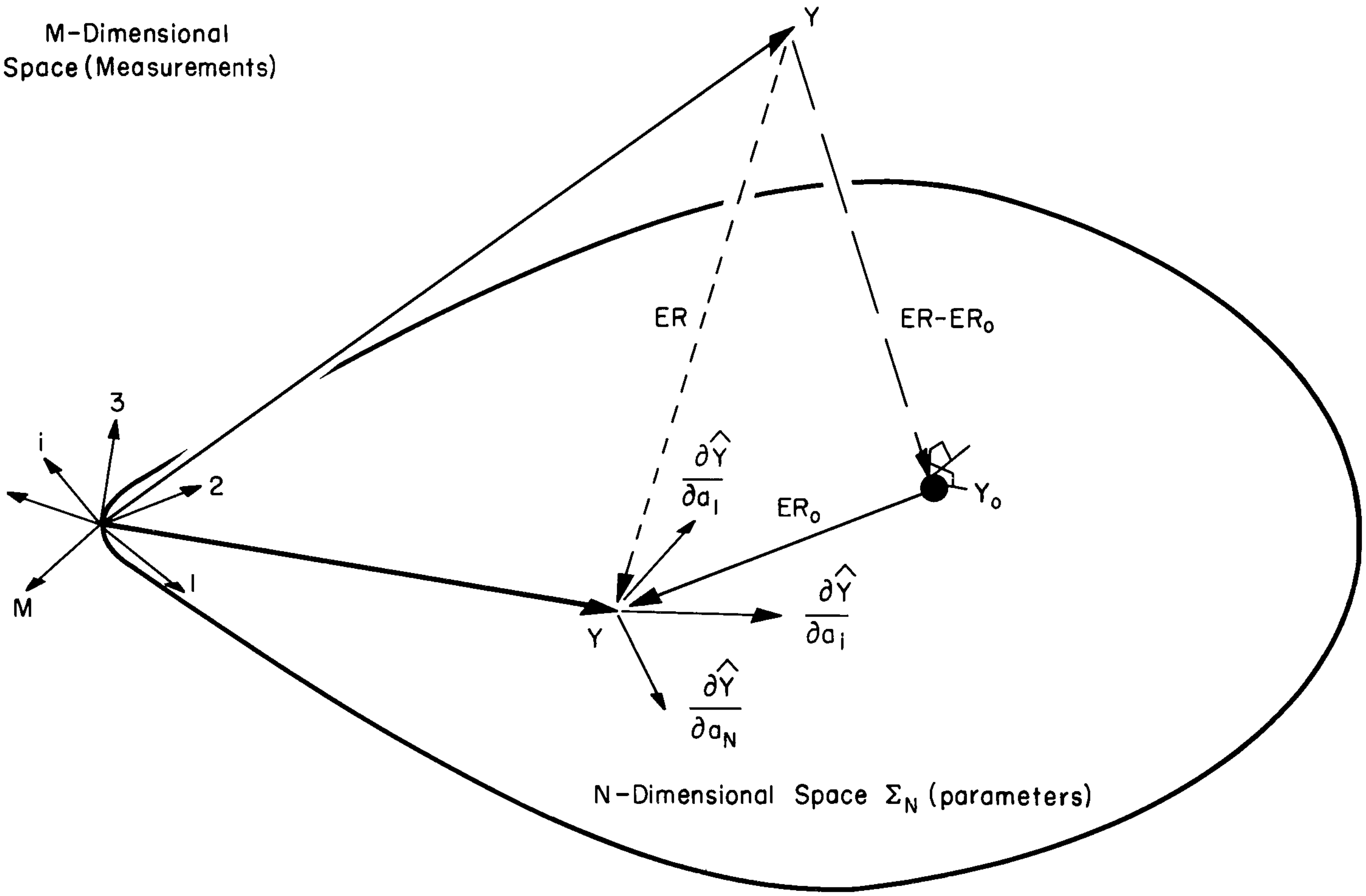

M-Dimensional Space (Measurements)
Y
ER
ER-ER$_o$
3
i
2
M
1
$\frac{\partial \hat{Y}}{\partial a_1}$
ER$_o$
Y$_o$
Y
$\frac{\partial \hat{Y}}{\partial a_i}$
$\frac{\partial \hat{Y}}{\partial a_N}$
N-Dimensional Space Σ_N (parameters)

Figure 4.- Output measurements space E_M and model space Σ_N.

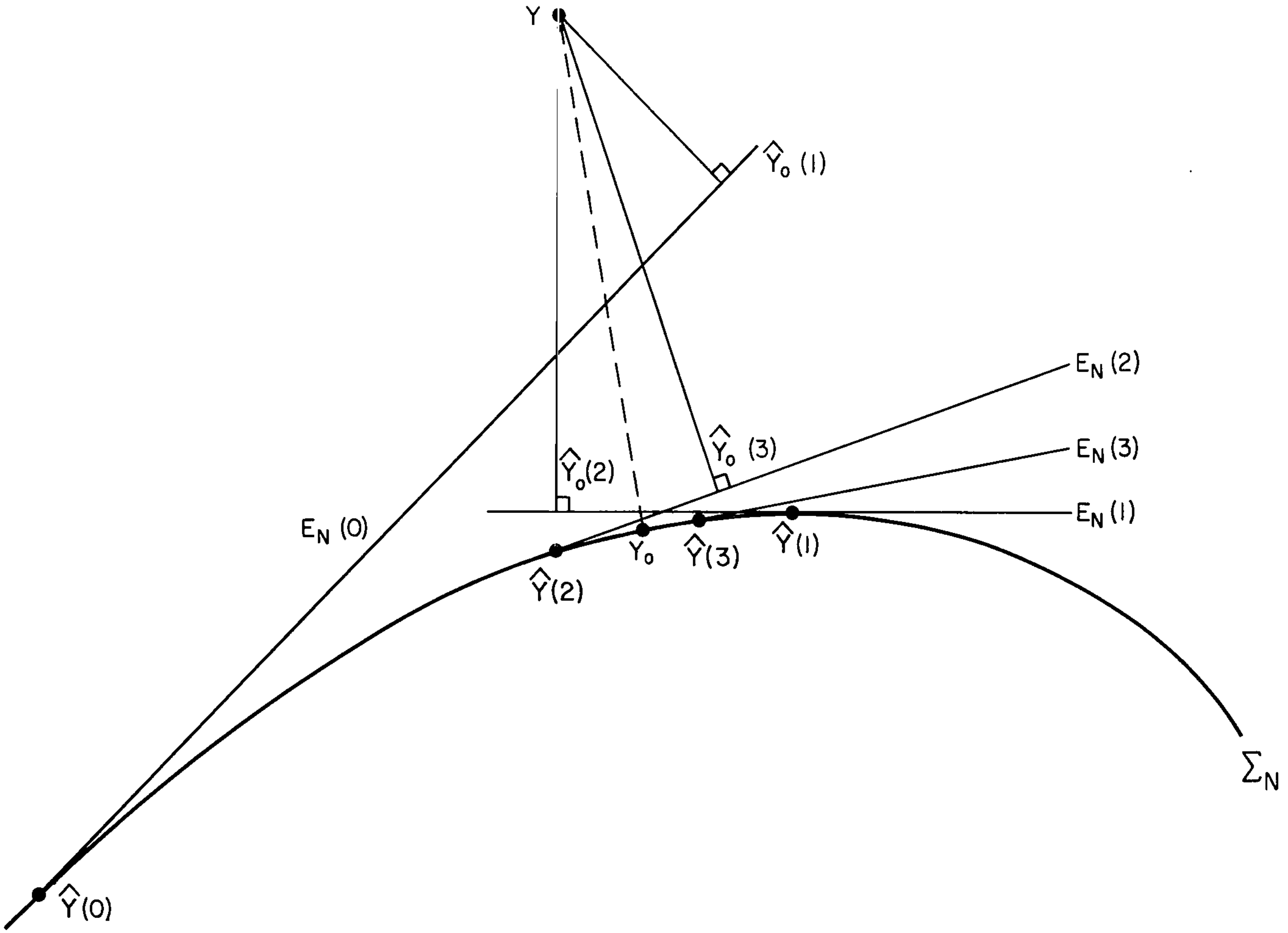


Figure 5.- The iterative process in a nonlinear case.

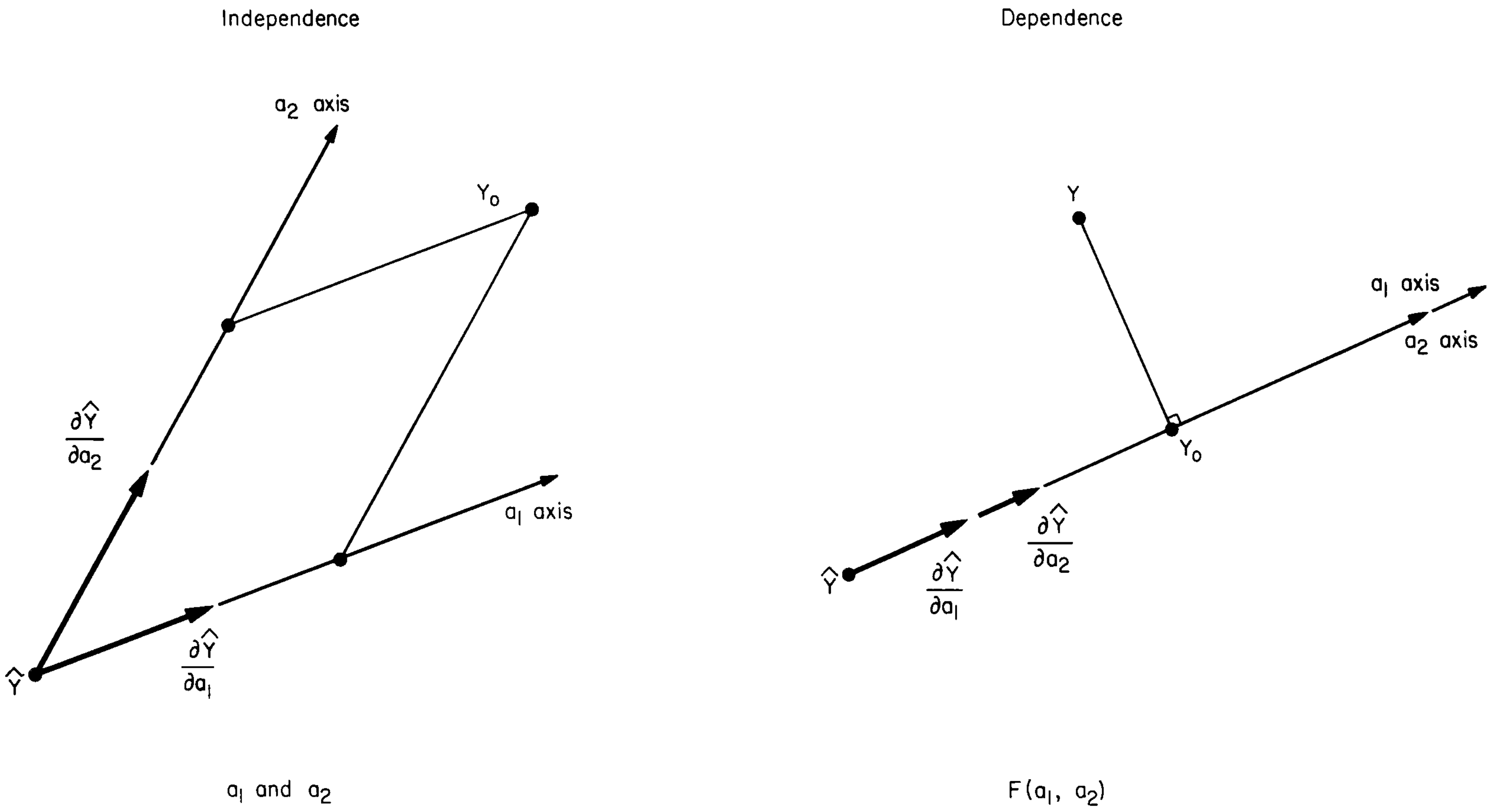

Independence
Dependence
a_2 axis
Y_o
Y
a_1 axis
a_2 axis
$\frac{\partial \hat{Y}}{\partial a_2}$
Y_o
a_1 axis
$\frac{\partial \hat{Y}}{\partial a_2}$
$\hat{Y}$
$\frac{\partial \hat{Y}}{\partial a_1}$
$\hat{Y}$
$\frac{\partial \hat{Y}}{\partial a_1}$
a_1 and a_2
$F(a_1, a_2)$

Figure 6.- Linear dependence and independence.

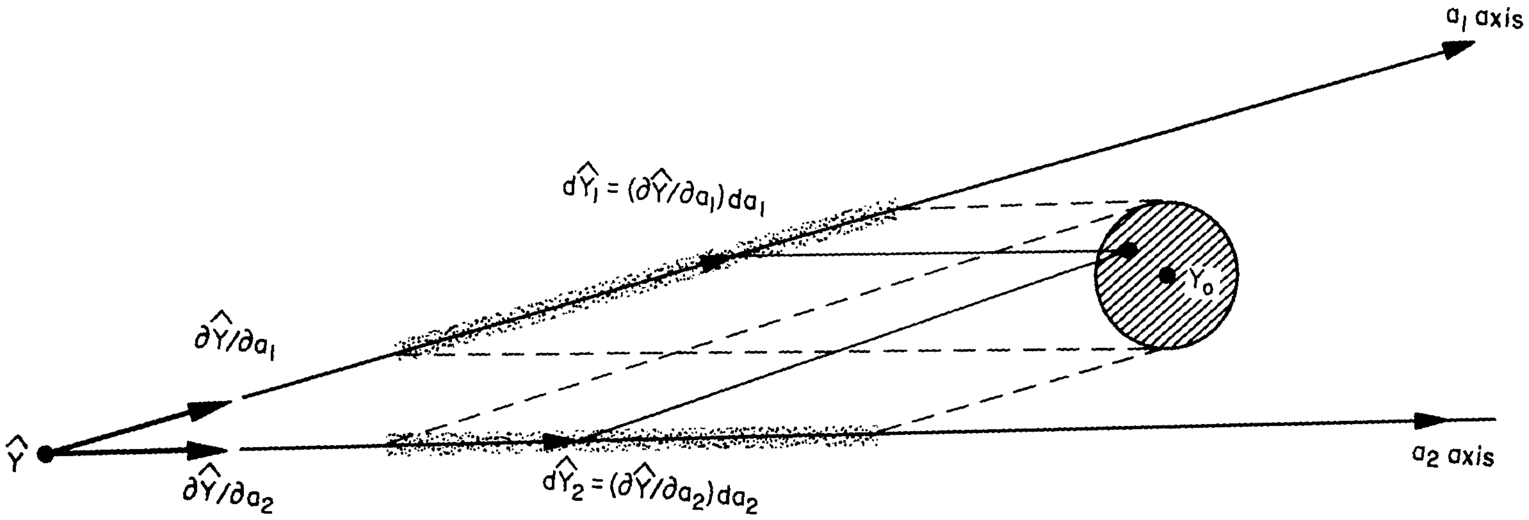

a_1 axis
$d\hat{Y}_1 = (\partial\hat{Y}/\partial a_1)da_1$
$\partial\hat{Y}/\partial a_1$
Y_o
$\hat{Y}$
$\partial\hat{Y}/\partial a_2$
$d\hat{Y}_2 = (\partial\hat{Y}/\partial a_2)da_2$
a_2 axis

Figure 7.- Linear closeness in two dimensions.

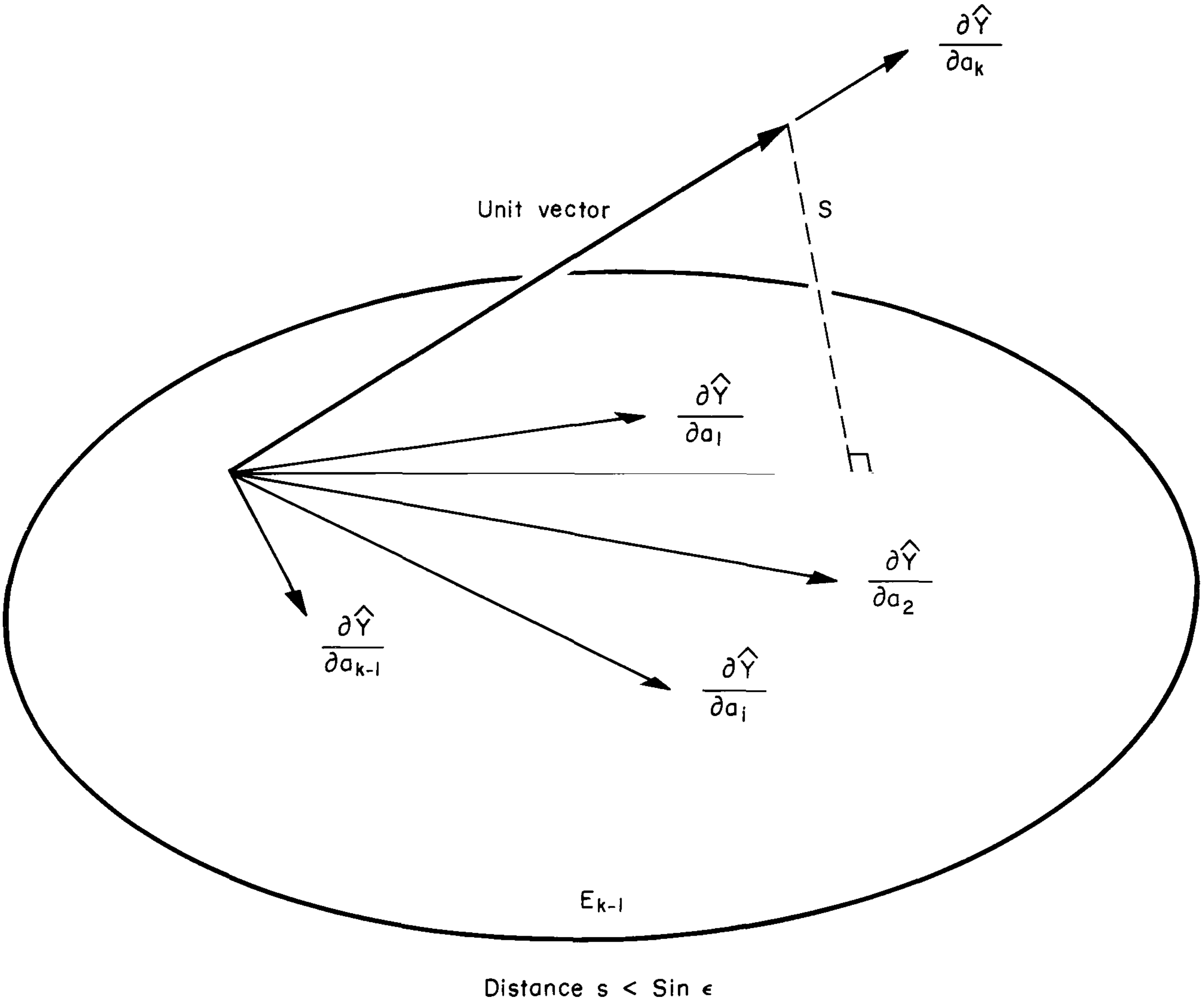
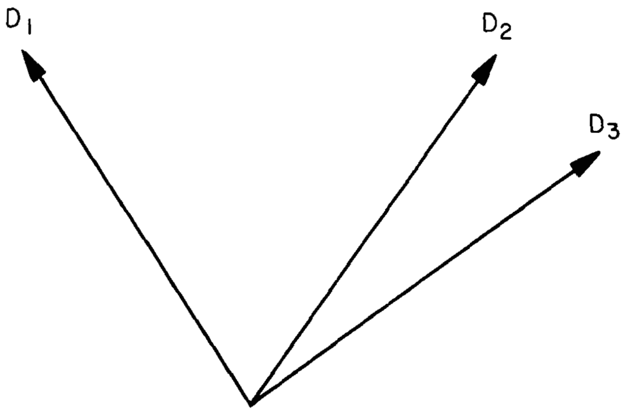

$\frac{\partial \hat{Y}}{\partial a_k}$
Unit vector
s
$\frac{\partial \hat{Y}}{\partial a_1}$
$\frac{\partial \hat{Y}}{\partial a_2}$
$\frac{\partial \hat{Y}}{\partial a_{k-1}}$
$\frac{\partial \hat{Y}}{\partial a_i}$
E_{k-1}
Distance s < Sin ϵ

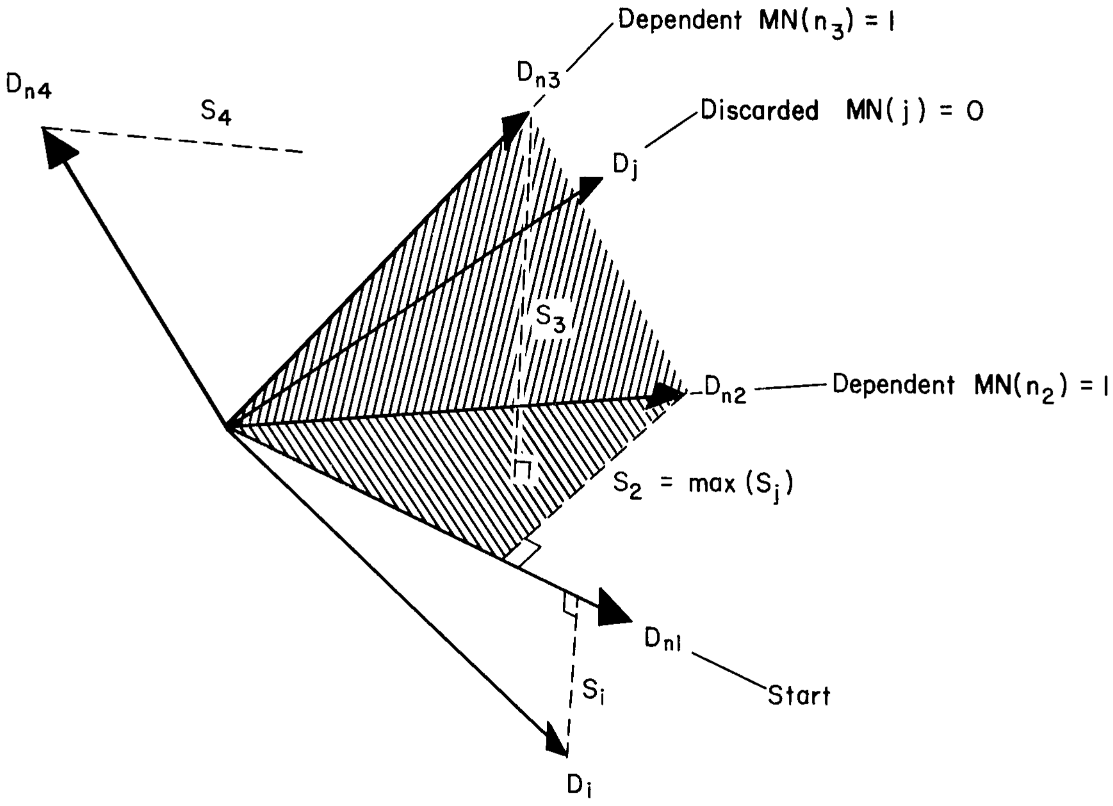
Figure 8.- Linear closeness in k dimensions.



D_2 or D_3 should be discarded

Figure 9.- Choice of optimal basis.

$$D_i = \frac{\delta \hat{Y}}{\delta a_i} \Big/ \left| \frac{\delta \hat{Y}}{\delta a_i} \right|$$



Basic vectors

D_{n1}

D_{n2}

D_{n3}

D_{n4}

⋮

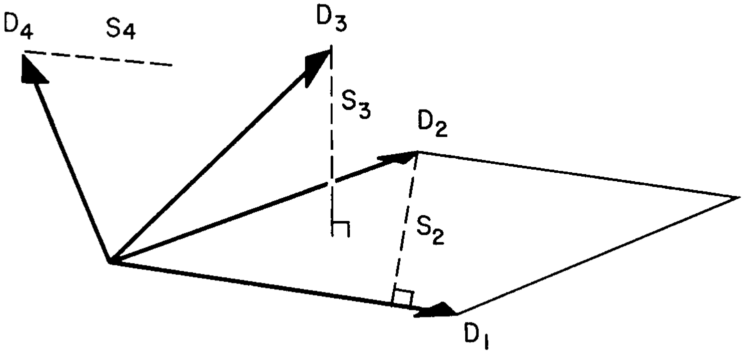
Dependent set a_{n2} , a_{n3} , a_j

Figure 10.- Dependence analysis.

$$G = D^T D$$

Gram determinant

$$DET(G) = V^2$$



Recursive computations of V^2

$$V^2_{k+1} = V^2_k \times S^2_{k+1}$$

$$DET_{k+1} = DET_k \times Sii$$

$$DET(G) = DET_k \times \left| \begin{array}{cccccccc} 1 & x & x & x & x & x & \cdots & \\ & 1 & x & x & x & x & \cdots & \\ & & \vdots & \vdots & \vdots & \vdots & & k \text{ First basis vectors} \\ & & & 1 & x & x & \cdots & \\ & & & & g_{ii} & x & \cdots & \\ & & & & \bullet & g_{jj} & \cdots & \\ & & & & \bullet & \bullet & \cdots & \end{array} \right|$$

$$Sii = S^2_{k+1}$$

Figure 11.- Computation technique.

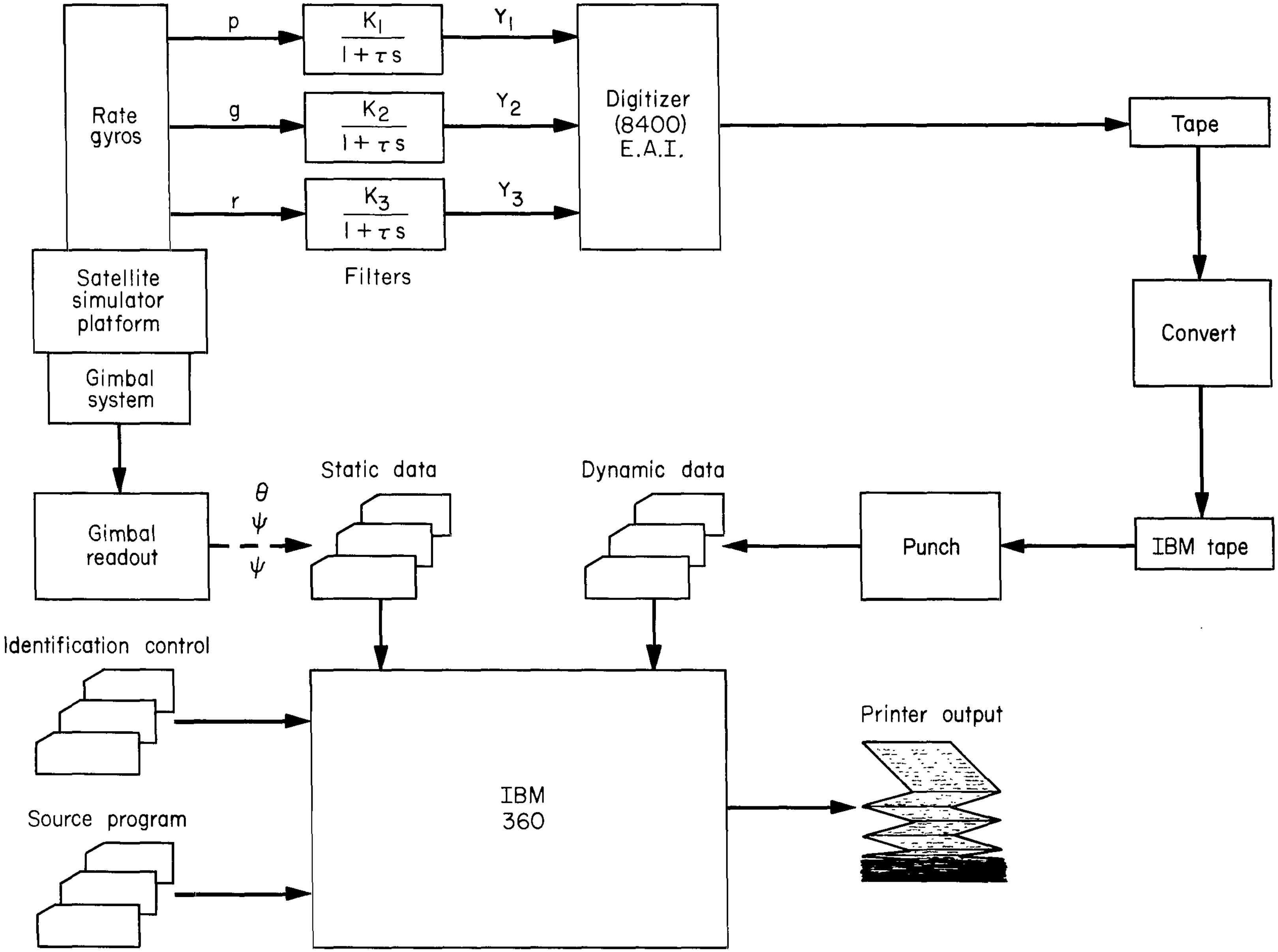


Figure 12.- General arrangement for the identification of the satellite simulator.

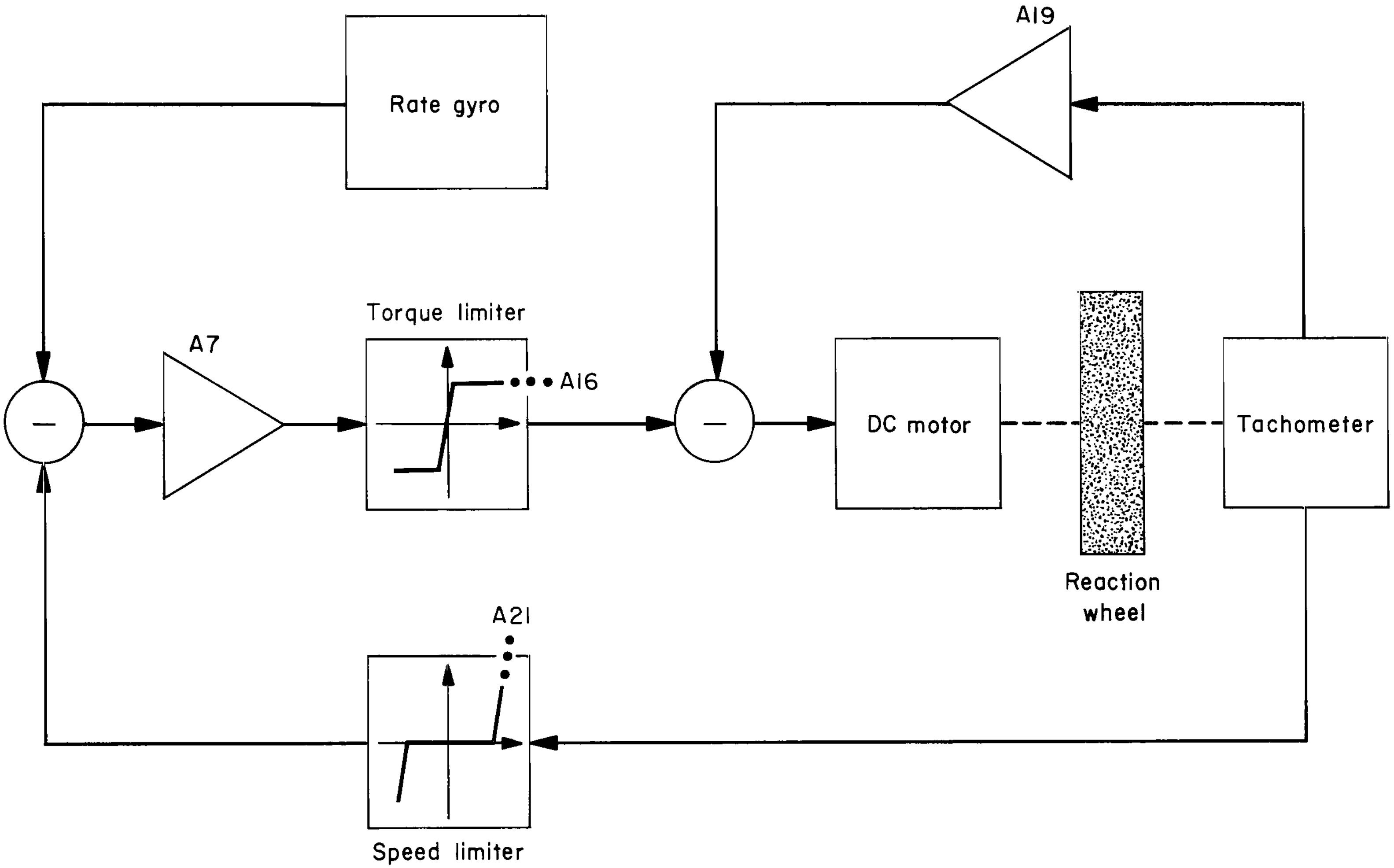


Figure 13.- Damping system (roll axis).

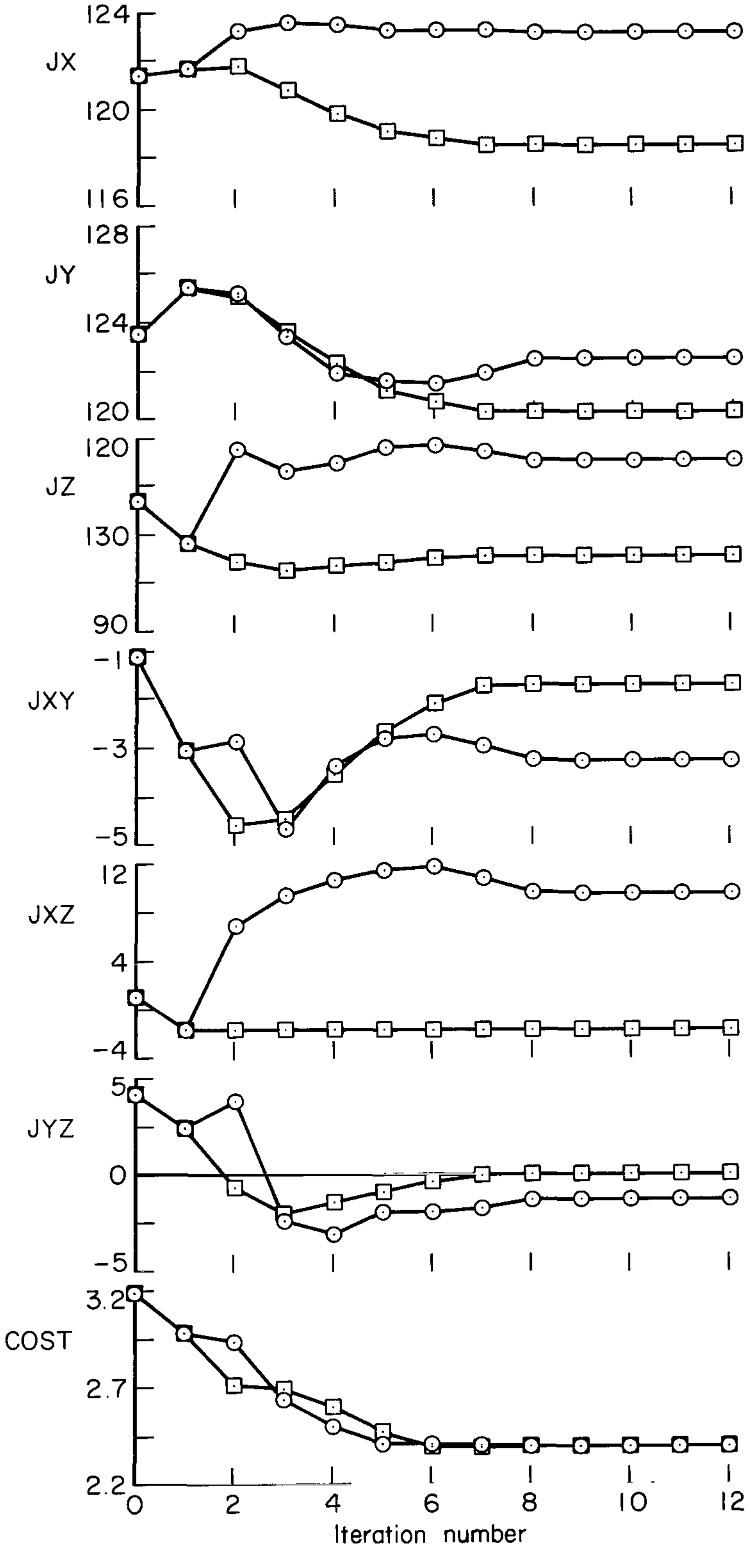


Figure 14.- Evolution of the parameters during two different identification runs (Case A).

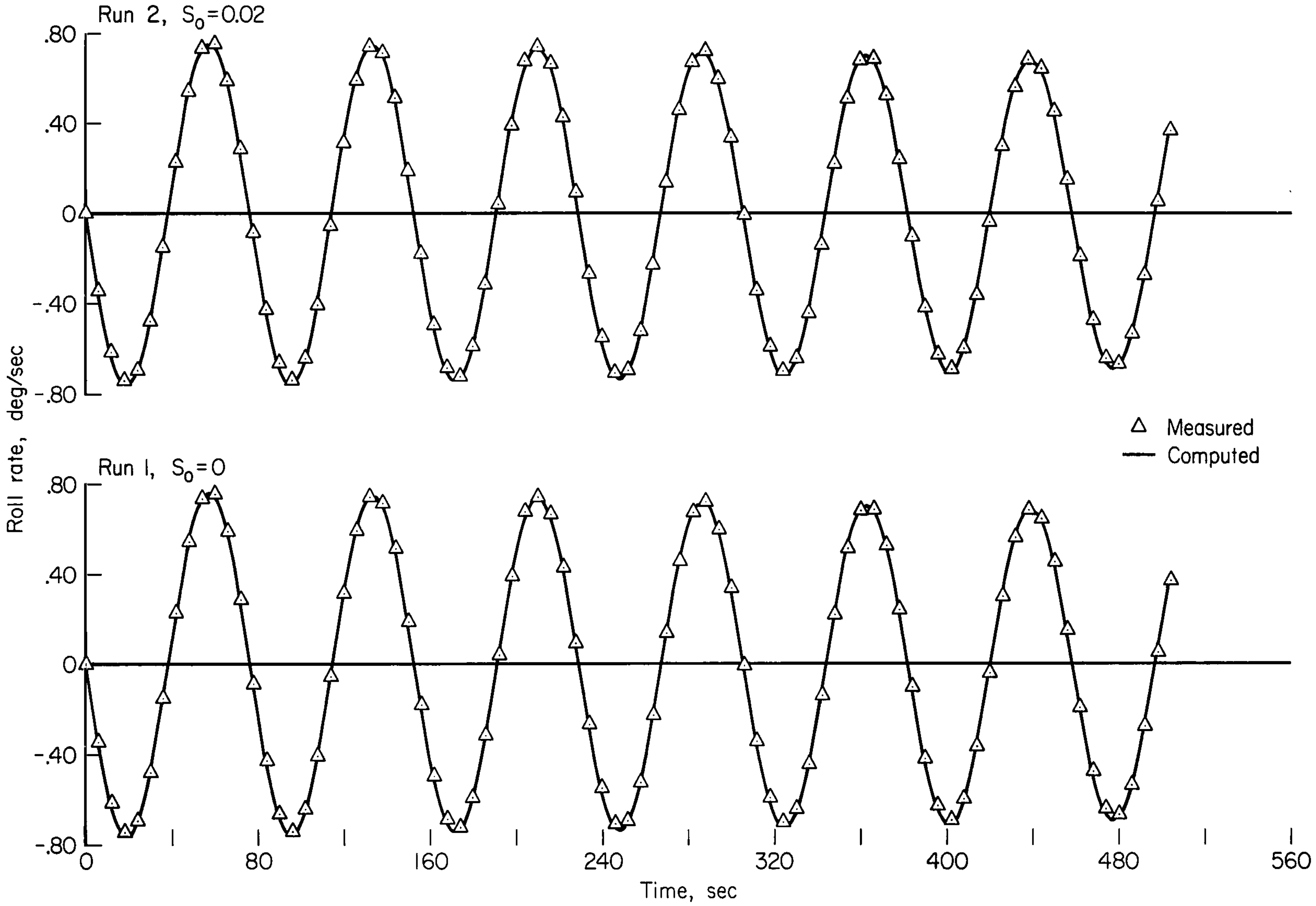


Figure 15.- Measured and computed time histories of the roll rate of the platform at the end of two different identification runs (Case A).

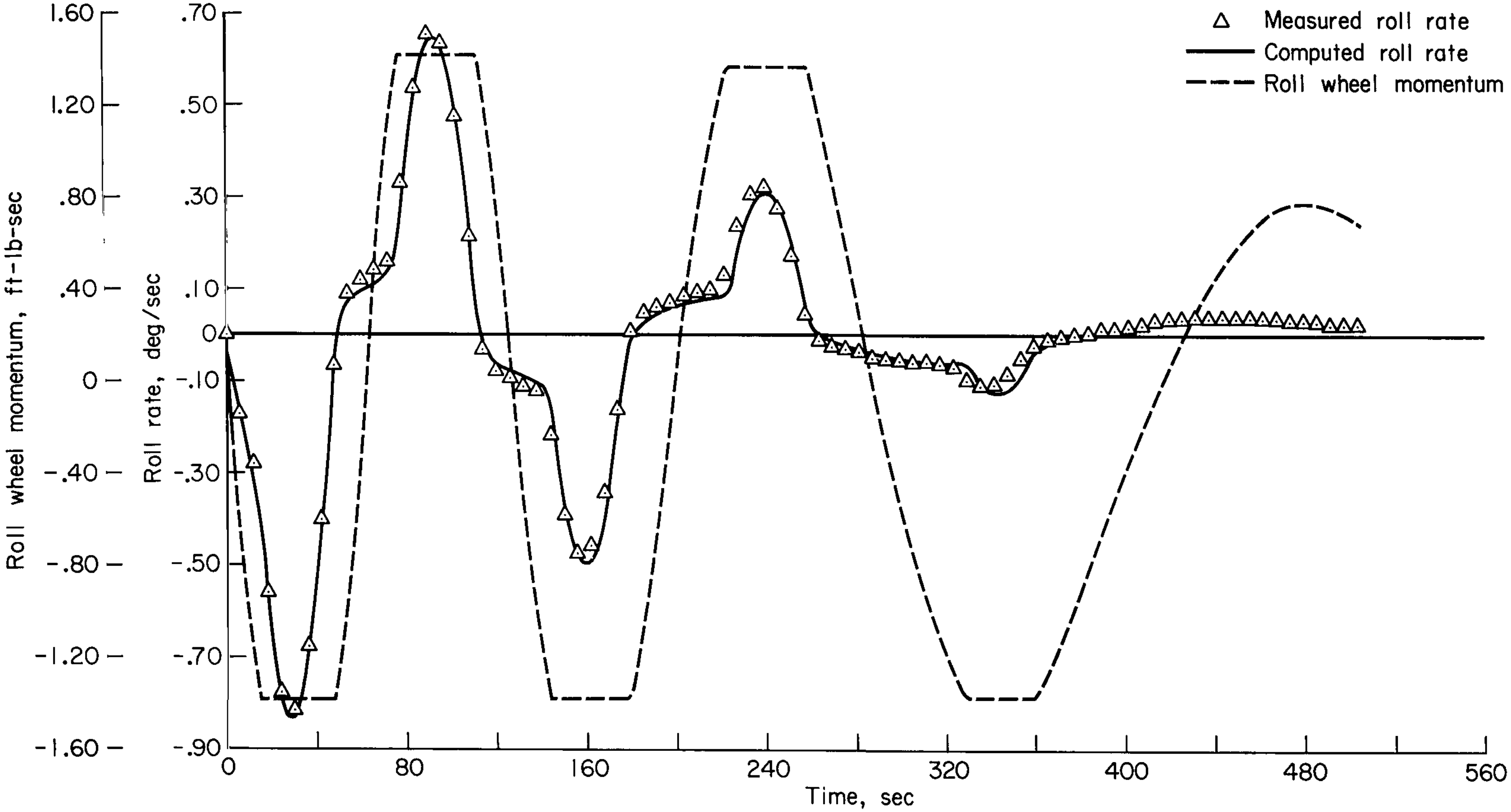
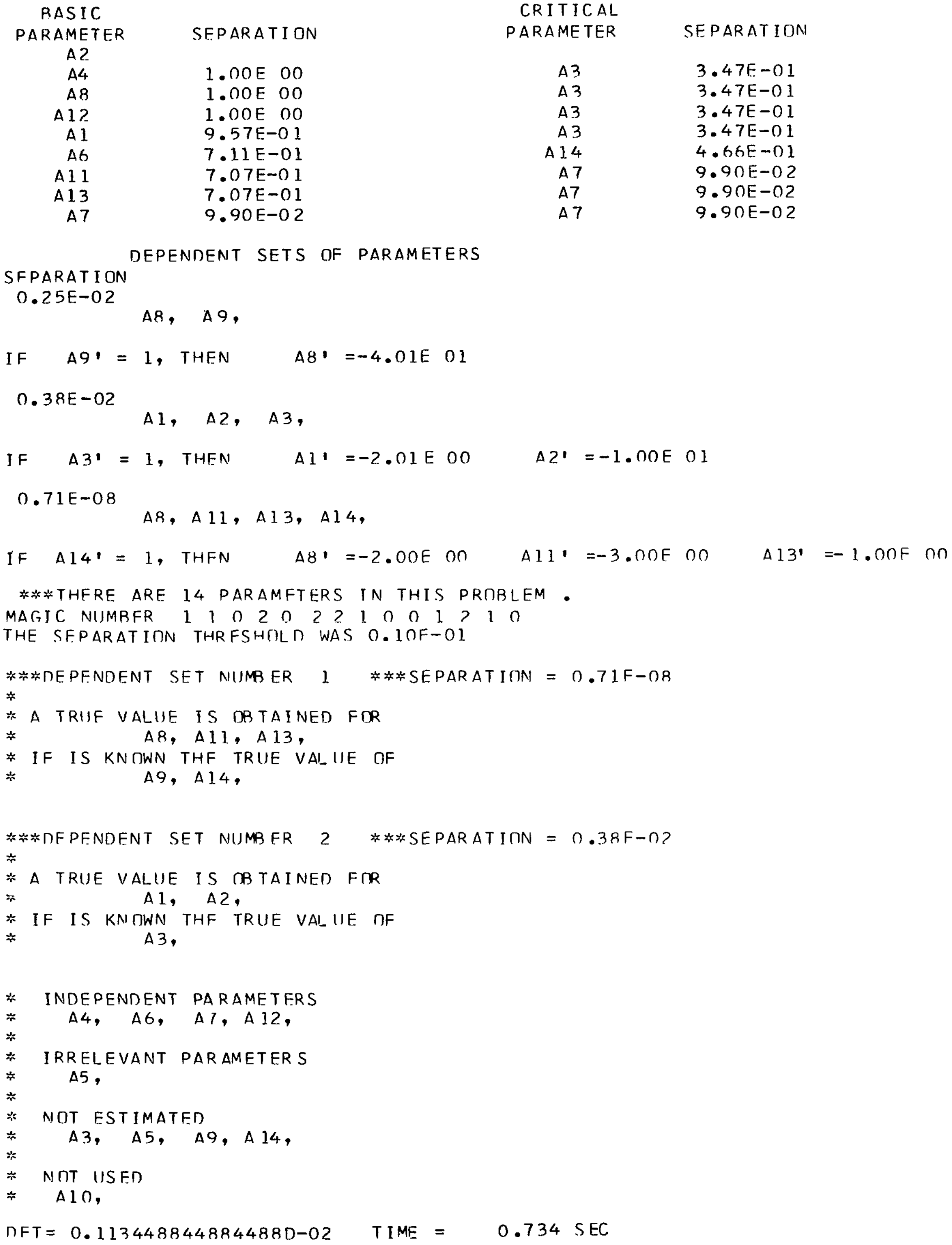
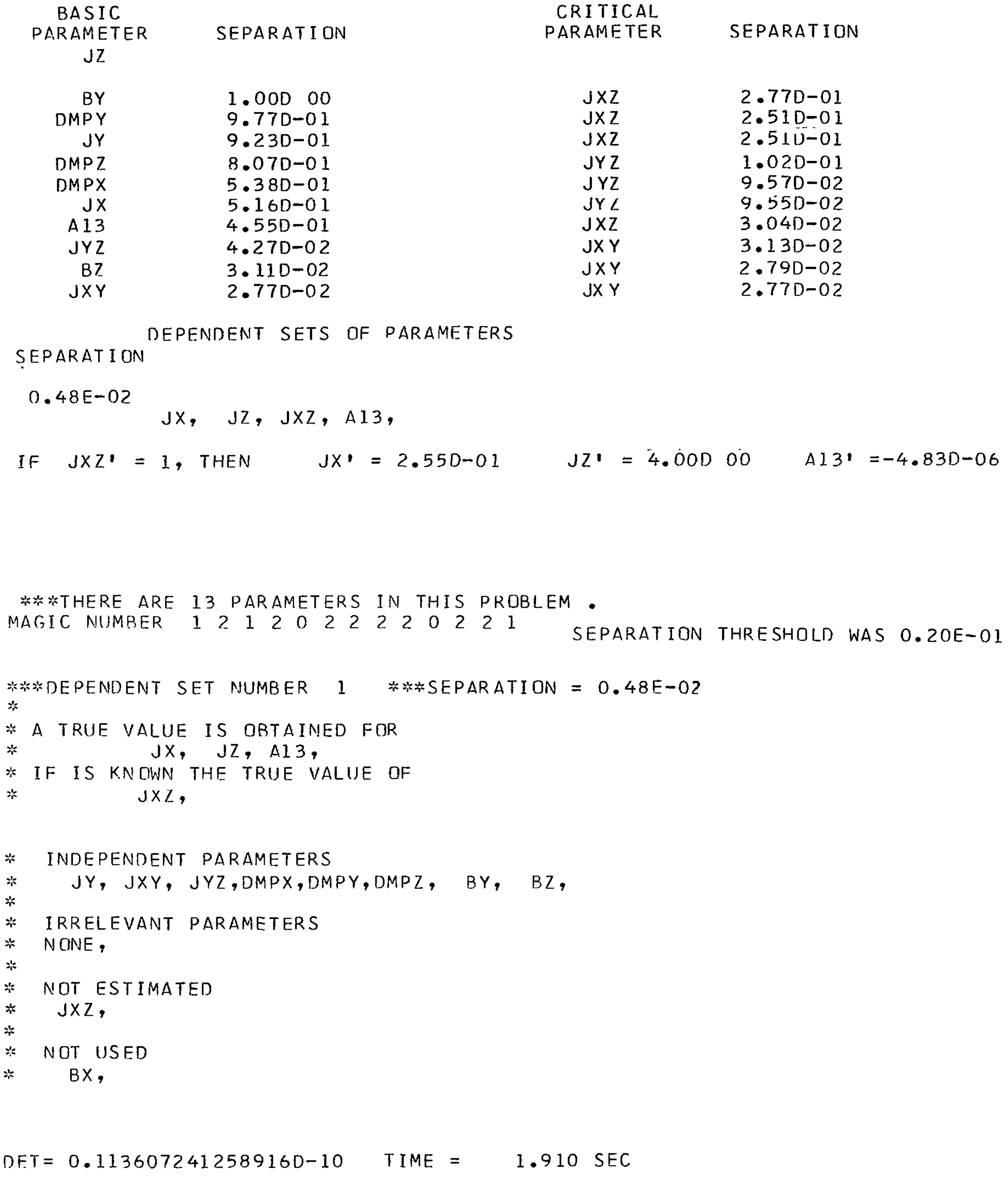


Figure 16.- Measured and computed time histories of the roll rate (Case B).



```
 BASIC                                          CRITICAL
PARAMETER        SEPARATION                     PARAMETER        SEPARATION
    A2
    A4             1.00E 00                        A3              3.47E-01
    A8             1.00E 00                        A3              3.47E-01
   A12             1.00E 00                        A3              3.47E-01
    A1             9.57E-01                        A3              3.47E-01
    A6             7.11E-01                       A14              4.66E-01
   A11             7.07E-01                        A7              9.90E-02
   A13             7.07E-01                        A7              9.90E-02
    A7             9.90E-02                        A7              9.90E-02

            DEPENDENT  SETS  OF  PARAMETERS
SEPARATION
 0.25E-02
            A8,  A9,

IF   A9' = 1, THEN      A8' =-4.01E 01

 0.38E-02
            A1,  A2,  A3,

IF   A3' = 1, THEN      A1' =-2.01E 00      A2' =-1.00E 01

 0.71E-08
            A8, A11, A13, A14,

IF  A14' = 1, THEN      A8' =-2.00E 00     A11' =-3.00E 00     A13' =-1.00E 00

 ***THERE ARE 14 PARAMETERS IN THIS PROBLEM .
MAGIC NUMBER  1 1 0 2 0 2 2 1 0 0 1 2 1 0
THE SEPARATION THRESHOLD WAS 0.10E-01

***DEPENDENT SET NUMBER  1    ***SEPARATION = 0.71E-08
*
* A TRUE VALUE IS OBTAINED FOR
*           A8, A11, A13,
* IF IS KNOWN THE TRUE VALUE OF
*           A9, A14,


***DEPENDENT SET NUMBER  2    ***SEPARATION = 0.38E-02
*
* A TRUE VALUE IS OBTAINED FOR
*           A1,  A2,
* IF IS KNOWN THE TRUE VALUE OF
*           A3,


*  INDEPENDENT PARAMETERS
*    A4,   A6,  A7, A12,
*
*  IRRELEVANT PARAMETERS
*    A5,
*
*  NOT ESTIMATED
*    A3,   A5,  A9, A14,
*
*  NOT USED
*   A10,

DFT= 0.113448844884488D-02    TIME =     0.734 SEC
```

Figure 17.- Computer analysis in the test case of 14 parameters.



```
      BASIC                                        CRITICAL
    PARAMETER       SEPARATION                    PARAMETER       SEPARATION
       JZ

        BY           1.00D 00                        JXZ            2.77D-01
      DMPY           9.77D-01                        JXZ            2.51D-01
        JY           9.23D-01                        JXZ            2.51D-01
      DMPZ           8.07D-01                        JYZ            1.02D-01
      DMPX           5.38D-01                        JYZ            9.57D-02
        JX           5.16D-01                        JYZ            9.55D-02
       A13           4.55D-01                        JXZ            3.04D-02
       JYZ           4.27D-02                        JXY            3.13D-02
        BZ           3.11D-02                        JXY            2.79D-02
       JXY           2.77D-02                        JXY            2.77D-02

             DEPENDENT SETS OF PARAMETERS
  SEPARATION

   0.48E-02
               JX,  JZ, JXZ, A13,

  IF  JXZ' = 1, THEN    JX' = 2.55D-01     JZ' = 4.00D 00    A13' =-4.83D-06




   ***THERE ARE 13 PARAMETERS IN THIS PROBLEM .
  MAGIC NUMBER  1 2 1 2 0 2 2 2 2 0 2 2 1
                                                  SEPARATION THRESHOLD WAS 0.20E-01

  ***DEPENDENT SET NUMBER  1    ***SEPARATION = 0.48E-02
  *
  * A TRUE VALUE IS OBTAINED FOR
  *          JX,  JZ, A13,
  * IF IS KNOWN THE TRUE VALUE OF
  *         JXZ,


  *  INDEPENDENT PARAMETERS
  *    JY, JXY, JYZ,DMPX,DMPY,DMPZ,  BY,  BZ,
  *
  *  IRRELEVANT PARAMETERS
  *  NONE,
  *
  *  NOT ESTIMATED
  *   JXZ,
  *
  *  NOT USED
  *    BX,


  DET= 0.113607241258916D-10   TIME =    1.910 SEC
```

Figure 18.- Computer analysis of the identification of the parameters of the platform in the free oscillation case (Case A, run 2).

NASA-Langley, 1971 —— 19 A-3807