1972

Parameter identification of minimal realizations

David Passeri
Iowa State University

72-19,998

PASSERI, David Paul, 1940-
PARAMETER IDENTIFICATION OF MINIMAL REALIZATIONS.

Iowa State University, Ph.D., 1972
Engineering, electrical

University Microfilms, A XEROX Company, Ann Arbor, Michigan

Parameter identification of minimal realizations

by

David Passeri

A Dissertation Submitted to the

Graduate Faculty in Partial Fulfillment of

The Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Major Subject: Electrical Engineering

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Iowa State University
Ames, Iowa

1972

TABLE OF CONTENTS

1.0. INTRODUCTION

Since an accurate mathematical model is essential to any system design involving optimization or state estimation, the need to be able to correctly specify a germane mathematical model is well established. Formulation of a mathematical model can be divided into two broad categories, the situation in which there is no a priori information about the system structure so that the topology of the model must be defined and the case in which partial information is available that describes the system. In both cases complete specification of the model requires estimation of the unknown model parameters through use of experimental data. The most general mathematical model that in practice would represent most physical systems would be a nonlinear time varying multiple input/multiple output model. However, in many cases linearizied model can be used to approximate the physical system resulting in a simpler mathematical model for which the majority of modern control system theory has been developed. Consequently the brunt of recent research has concentrated on the formulation of linear nontime varying multiple input/multiple output models.

Two elegant solutions to the specification of a multiple input/multiple output mathematical model in which no a priori information is available have been proposed in recent years. The first is due to B. C. Ho (1) in 1965. Given the impulse response of a system Ho's algorithm generates a minimal realization of a system. The second method first proposed by Gopinath (2) and later extended by Budin (3) requires only knowledge of input/output observations in order to specify a minimal realization.

This document concerns itself with the second case, that in which partial information is available that describes the system and where the parameters that make up the mathematical model are to be estimated. A considerable amount of previous work has been accomplished in this area and an exhaustive literature search can be found in reference (4).

Parameter estimation of single input/single output systems has been the subject of many papers in recent years. The techniques used range from stochastic approximation to model reference. Steiglitz and McBride(5) propose a model reference scheme that identifies the linear system by minimizing the error between system and model. Saridis and Stein (6), Sakrison (7), and Holmes (8) all use stochastic approximation algorithms for determining the system model. Neal and Bekey (9) have investigated parameter estimation of sampled-data systems by stochastic approximation. Wong and Polak (10) have applied the instrumental variable method used in economics for on-line system identification. Approximation of the impluse response of a pulse-transfer function by linear combinations of orthonormal sequences has been proposed by Tretter (11). These techniques all suffer in that they cannot be easily extended to the multiple input/multiple output system.

Augmentation of the Kalman-Bucy filter (12) in which the unknown system parameters are included as state elements has been attempted by Kopp and Orford (13), however results using this method have not been favorable. Denery (14) considers the problem of identifying a multiple-output/single input system by using a gradient algorithm that combines the equation error method and the output error method. The above methods are suitable for multiple input/multiple output systems yet present computational difficulties.

This dissertation presents criteria that specify necessary requirements for parameter identification of multiple input/multiple output systems. A canonical form for multiple input/multiple output systems is proposed, along with the necessary transformation matrix, that is suitable for use in the identification algorithm used in this document. The identification algorithm is an on-line algorithm that estimates system parameters directly from noisy input/output observations, the only requirement being that the noise covariances be known. Computation is straight forward requiring only simple matrix operations.

Application of the identification algorithm has been made to estimating parameters in both the longitudinal and lateral equations of motion of an aircraft under simulated flight conditions.

2.0. STRUCTURE OF LINEAR SYSTEMS

2.1. External and Internal Descriptions

The structure of linear dynamical systems has been a popular subject for many years. The older literature on control theory described linear systems by means of transfer functions, e.g., the Laplace transforms of the differential equations relating the inputs to the outputs. Formalization of the mathematical definition of a linear dynamical system was first made by Kalman (15) in 1963. Recent endeavors, Kalman (16, 17, 18), Wymore (19), and Zeiger (20) have been centered in defining a dynamical system as an algebraic structure; the most elegant description being due to Kalman who describes a linear dynamical system Σ as having a module structure.

This dissertation will discuss the structure of a linear system from the viewpoint of a K-vector-space structure. In defining the structure of a dynamical system the discussion will be limited only to systems that are: discrete-time, linear, constant, having a finite number of inputs and outputs, and constructed with numbers from a fixed field K. Definitions of the terminology used in discussing K-vector spaces can be found in reference (21).

<u>Definition 2.1</u>:

A discrete-time, constant, linear, r-input, p-output dynamical system Σ over a field K is a composite concept $\Sigma = (F,G,H)$ where

$$F: X \rightarrow X$$
$$G: K^r \rightarrow X$$
$$H: X \rightarrow K^p$$

are abstract K-homomorphisms and X is an abstract vector space K^n (the state space). The triple (F,G,H) defines the <u>internal description</u> of a system via the equations

$$x(k+1) = Fx(k) + Gu(k)$$
$$y(k) = Hx(k) \qquad (2.1)$$

where $k \in Z^+ = \{0,1,2,\ldots\}$, $x \in X$, $u(k) \in K^r$, $y(k) \in K^p$. Here F,G,H are $n\times n$, $n\times r$, $p\times n$ matrices over the field K. The dimension of the system, dim Σ, is defined as the dim X.

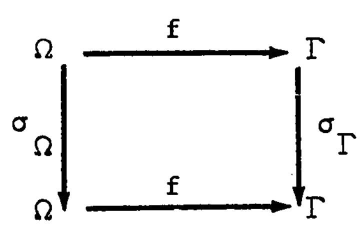
<u>Definition 2.2</u>:

The <u>external description</u> of the system Σ is described by the K-linear input/output map f, $f:\Omega\rightarrow\Gamma$, where $\Omega = \{u(0), u(1), \ldots\}$ and $\Gamma = \{y(1), y(2), \ldots\}$ vector spaces over a field K. The map f has the following properties:

a) f is defined as the zero-state impulsive response map $\{HF^iG;\ i=0,1,\ldots\}$. Given an arbitrary input sequence the output sequence is,

$$(HGu(0),\ \sum_{t=0}^{1} HF^{1-t}Gu(t),\ \sum_{t=0}^{2} HF^{2-t}Gu(t)\ \ldots) \qquad (2.2)$$

b) the map f is invariant under time translation in the sense that the diagram depicted below



commutes with respect to the shift operators σ_Ω and σ_Γ defined as

$$\sigma_\Omega: \ (u(0), u(1), \ldots) \to (u(1), u(2), \ldots)$$

$$\sigma_\Gamma: \ (y(0), y(1), \ldots) \to (y(1), y(2), \ldots)$$

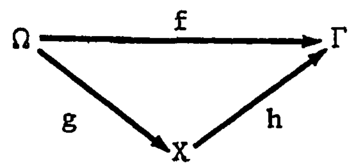
Note that the external description has the property of causality. The external description of a system can be related to the internal description through the concept of a realization.

2.2. Minimal Realizations

A linear dynamical system Σ defined by Equation 2.1 is called a realization of an input/output map f defined by Definition 2.2 if the input/output map f_Σ of Σ is equal to f.

Definition 2.3:

Let Ω and Γ be arbitrary K-vector spaces and $f: \Omega \to \Gamma$ an arbitrary K-homomorphism. We say that f is factored through a vector space X iff there exist a commutative diagram



where g is a surjective K-homomorphism and h an injective K-homomorphism. Such a factorization is called a realization.

Having an external description of the system Σ one would like to deduce the internal description in terms of the triple (F,G,H). In order to do so, some additional structure must be placed on the abstract vector space X; that is, given the K-homomorphism f what requirements need be placed on the state space X such that a factorization of f is possible via X.

Since the mapping g is required to be onto, every $x \in X$ is the image of at least one $\omega \in \Omega$. This is precisely the requirement that the system Σ be completely reachable. Recall that a system is completely reachable iff

$$\rho[G, FG, \cdots, F^{n-r} G] = n = \dim X \tag{2.3}$$

where $r = \rho[G] = \text{rank}\,[G]$.

Similarly the requirement that the map h be 1-1 is equivalent to saying that the system Σ be completely observable. That is,

$$\rho[H^T, F^T H^T, \cdots, (F^T)^{n-p} H^T] = n = \dim X \tag{2.4}$$

where $p = \rho[H]$.

<u>Proposition 2.4</u>:

If Σ is both completely reachable and completely observable then f_Σ is a realization of the input/output map f.

Proof: Since Σ is both completely reachable and completely observable the map $f_\Sigma = g \cdot h$ is a factorization of f; therefore, $f = f_\Sigma$.

<u>Definition 2.5</u>:

A realization Σ of f is called a canonical or minimal realization iff Σ is both completely reachable and completely observable.

<u>Theorem 2.6</u>:

Given an external description, f, of the system Σ, two minimal realizations $\Sigma = (F,G,H)$ and $\tilde{\Sigma} = (\mathcal{F},\mathcal{G},\mathcal{H})$ are isomorphic to each other if there exist a unique K-isomorphism $A: X \to \mathscr{X}$ such that the factorization remains commutative. The system isomorphism can be expressed by the following matrix notations:

$$\text{(a)} \quad \mathscr{F} = AFA^{-1}$$

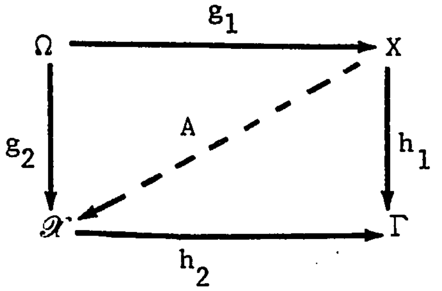
$$\text{(b)} \quad \mathscr{G} = AG \qquad (2.5)$$

$$\text{(c)} \quad \mathscr{H} = HA^{-1}$$

Proof: The proof requires the following lemma.

<u>Lemma 2.7</u>:

Let Ω, X, $\mathscr{X}$, Γ be arbitrary sets. Consider the commutative diagram



(2.6)

and suppose that g_1 and g_2 are onto mappings and h_1 and h_2 are one-to-one mappings. Then there is a unique map A such that the diagram remains commutative.

Proof: Any element $x \in X$ is the image of at least one element of Ω. Hence if $x \in X$ then there is at least one element $\omega \in \Omega$ such that the set $g_1^{-1}(x)$ is not void. For any $\omega \in g_1^{-1}(x)$ commutivity implies that

$$(h_2 \cdot g_2)(\omega) = \gamma = (h_1 \cdot g_1)(\omega) = h_1(x)$$

Since h_2 is 1 - 1, there is a unique $x = h_2^{-1}(\gamma)$ then for such an element $x \in \mathscr{X}$

$$g_2(\omega) = (h_2^{-1} \cdot (h_1 \cdot g_1)(\omega)) = (h_2^{-1} \cdot h_1) \cdot g_1)(\omega)$$

If the map A is defined as $A(x) = (h_2^{-1} \cdot h_1)(x)$ so that

$$g_2(\omega) = (A \circ g_1)(\omega)$$

then the triangle $X\mathcal{X}\Gamma$ commutes since

$$(h_2 \cdot A)(x) = (h_2 \circ (h_2^{-1} \circ h_1)(x)) = h_1(x)$$

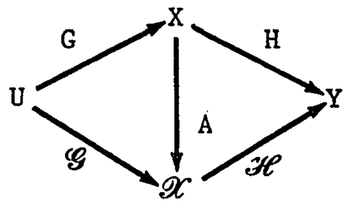
Similarity of the mappings in the commutative diagram implies the proof in the opposite direction. Now since h_1 and h_2 are both 1 - 1 mappings and the mapping A is unique then clearly A^{-1} exists.

The following corollary stated without proof is an immediate result.

Corollary 2.8:

If the commutative Diagram 2.6 involves K-vector spaces and K-homomorphisms then A is a K-homomorphism.

We now proceed with the proof of Theorem 2.6. Consider the following commutative diagram



(2.7)

Using Lemma 2.7, $A: X \rightarrow \mathcal{X}$ is a unique vector space isomorphism which is compatible with the commutativity of the factorization.

The matrix relationships (b) and (c) can be easily proved from the diagram,

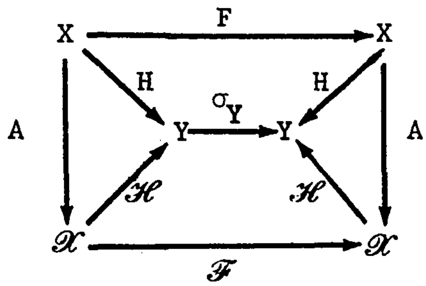
$$\mathcal{G}U = \mathcal{X} = AX = AGU$$

$$\mathcal{G} = AG$$

$$\mathcal{H}AX = \mathcal{H}\mathcal{X} = Y = HX$$

$$\mathcal{H} = HA^{-1}$$

To prove matrix relationship (a) consider the commutative diagram



(2.8)

From this diagram

$$\mathscr{X} = AX = AFX = \mathscr{F}\mathscr{X}$$
$$AFA^{-1} = \mathscr{F}$$

Theorem 2.6 states that all minimal realizations are equivalent to each other. The mapping or linear transformation A relating any two given minimal realizations can be readily found.

<u>Theorem 2.9</u>:

If (F,G,H) and $(\mathscr{F},\mathscr{G},\mathscr{H})$ are two given minimal realizations the linear transformation $AX = \mathscr{X}$ relating the two realizations can be computed as,

$$A = [\mathscr{G}, \mathscr{F}\mathscr{G}, \cdots, \mathscr{F}^{n-r}\mathscr{G}][G, \cdots, F^{n-r}G]^T \times \{[G, \cdots, F^{n-r}G][G, \cdots, F^{n-r}G]^T\}^{-1} \quad (2.9)$$

where $r = \text{rank}\,\mathscr{G} = \text{rank}\ G$

Proof: From Theorem 2.6 the linear transformation A can be related to the matrices G and F by Equation 2.5 a and b

$$\mathscr{G} = AG$$
$$\mathscr{F} = AFA^{-1} \quad (2.10)$$

Now
$$\mathcal{G} = AG$$
$$\mathcal{F}\mathcal{G} = AFA^{-1}AG = AFG$$
$$\mathcal{F}^2\mathcal{G} = \mathcal{F}\mathcal{F}\mathcal{G} = \mathcal{F}AFG = AFA^{-1}AFG = AF^2G$$
$$\vdots \qquad \vdots$$
$$\mathcal{F}^{n-r}\mathcal{G} = AF^{n-r}G$$

so that

$$[\mathcal{G}, \mathcal{F}\mathcal{G}, \cdots, \mathcal{F}^{n-r}\mathcal{G}] = A[G, FG, \cdots, F^{n-r}G] \tag{2.11}$$

Since Σ is completely reachable

$$\rho[\mathcal{G}, \mathcal{F}\mathcal{G}, \cdots, \mathcal{F}^{n-r}\mathcal{G}] = \rho[G, FG, \cdots, F^{n-r}G] = n$$

and the matrix

$$[G, FG, \cdots, F^{n-r}G][G, FG, \cdots, F^{n-r}G]^T \tag{2.12}$$

is nonsingular, one concludes that,

$$A = [\mathcal{G}, \mathcal{F}\mathcal{G}, \cdots, \mathcal{F}^{n-r}\mathcal{G}][G, \cdots, F^{n-r}G]^T$$
$$\times \{[G, FG, \cdots, F^{n-r}G][G, FG, \cdots, F^{n-r}G]^T\}^{-1}$$

The following observations can now be made with regard to the determination of an internal description from an external description:

(1) Given an external description of a system a factorization exists iff the system is completely observable and completely reachable. Practically this implies that one can identify only those states that are observable and with regard to controllability one can only hope to control controllable states. There is no need to consider any other situation.

(2) If any two systems Σ and $\tilde{\Sigma}$ are both completely reachable and completely observable and have the same input/output map f, then they differ only in the coordination of their state space.

(3) If a minimal realization Σ of f exists, then it is essentially uniquely determined by f, since coordination of states is irrelevant.

2.3. Minimal Realizations and Parameter Identification

Having an external description and the knowledge that Σ is a minimal realization of the map f, one may surmise that the n(n+r+p) parameters in the triple (F,G,H) can be uniquely determined. However, unless the structure of the triple (F,G,H) is constrained the n(n+r+p) parameters cannot be uniquely determined. The following simple example expounds the above statement. Given the external description of a single input/single output second order system that is completely reachable and completely observable the internal description has the form,

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} g_{11} \\ g_{12} \end{bmatrix} u(k), \tag{2.13}$$

$$y(k) = [h_{11} \quad h_{12}] \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}$$

while the external description is specified by the z-transfer function,

$$\frac{Y(z)}{U(z)} = \frac{(\beta_1 z + \beta_0)}{(z^2 + \alpha_1 z + \alpha_0)} \tag{2.14}$$

The internal description has eight unknown parameters, while the external description is completely specified by only four parameters, which in turn implies that only four parameters can be uniquely determined in the internal description via the external description. A possible constraint that can be placed on the triple (F,G,H) in the example is the well known single input/single output canonical form

$$F = \begin{bmatrix} 0 & 1 \\ f_{21} & f_{22} \end{bmatrix}, \quad G = \begin{bmatrix} g_{11} \\ g_{12} \end{bmatrix}, \quad H = [1 \quad 0] \tag{2.15}$$

If the triple (F,G,H) is known to have this canonical form then the following relationships exist between the parameters in the internal and external descriptions,

$$\begin{aligned} f_{21} &= -\alpha_0 \qquad & g_{11} &= \beta_1 \\ f_{22} &= -\alpha_1 \qquad & g_{12} &= \beta_0 - \alpha_1\beta_1 \end{aligned} \tag{2.16}$$

Analogously in multiple input/output systems all of the parameters in Equation 2.1 can not be uniquely determined.

Although many canonical forms for multiple input/output systems have been developed (22, 23) location of the identifiable parameters must fit the proposed parameter identification algorithm presented in Chapter 3 of this dissertation. Therefore canonical forms for minimal realizations will be presented as needed for use in the parameter identification algorithm.

2.4. Cyclic Spaces and Cyclic Systems

Although cyclic subspaces have been used in linear algebra (23), extension to the systems area has not been extensive. Cyclic spaces however frequently arise and the condition of a system not being cyclic is in general a singular case.

Let X be a finite abstract vector space over a field K and F a linear operator on X. If x is any vector in X, there is a smallest subspace of X which is invariant under F and contains x.

Definition 2.10:

If x is any vector in X, $x \neq 0$, the F-cyclic subspace generated by x is the subspace Z(x;F) spanned by the vectors $F^k x$, $k \geq 0$. If $Z(x;F) = X$ then x is called a cyclic vector for F.

In futher discussion our only concern will be the case where $Z(x;F) = X$ which induces the following definition.

Definition 2.11:

If x is a cyclic vector for F then the n-dimensional vector space X is said to be cyclic with respect to F if $\rho[x, Fx, \cdots, F^{n-1}x] = n$.

Definition 2.12:

If the state space X is cyclic with respect to F the system Σ is said to be cyclic.

Theorem 2.13:

If F is a linear operator on a finite n-dimensional vector space X, then F has a cyclic vector if and only if there is some ordered basis for X in which F is represented by the companion matrix of the minimal polynomial for F.

Proof: By definition if F has a cyclic vector then there is an ordered basis for X; namely, the vectors x, ... , $F^{n-1}x$. If $x_i = F^{i-1}x$, i = 1, ···, n then

$$Fx_i = x_{i+1} \; ; \; i = 1, \cdots, n-1$$
$$Fx_n = -c_0x_1 - c_1x_2 - \cdots - c_{n-1}x_n$$

By definition, the matrix of F in this basis is the transpose of the matrix of coefficeints of the above system of equations. Hence, the matrix has the form of a companion matrix of the minimal polynomial for F. Conversely if there exists a basis for X in which F is represented by the companion matrix of its minimal polynomial, then x is a cyclic vector for F.

The following remarks are direct results of Theorem 2.13. If x is a cyclic vector for F, then the minimal polynomial for F must have degree equal to the dimension of the space X; hence, by the Caley-Hamilton theorem the minimal and characteristic polynomials for F are identical. Conversely if the minimal and characteristic polynomials of F are identical then F has a cyclic vector.

The noncyclic condition occurs whenever the minimal and characteristic polynomials of F are not identical. This corresponds to the condition of two or more uncoupled Jordon blocks in the Jordon canonical form having the same minimal polynomial. This can easily be verified by considering the situation where two uncoupled Jordan blocks have the same minimal polynomial. From linear algebra the minimal polynomial of the Jordan canonical form of the operator F is the least common multiple of the minimum polynomials of the Jordan blocks. Consequently, if two or more minimum polynomials are identical then the characteristic polynomial and the minimum polynomial can not be identical.

Theorem 2.14:

If two or more uncoupled Jordan blocks in a Jordan canonical form have identical minimal polynomials the linear operator associated with the Jordan form is not cyclic.

Proof: The theorem has been proven in the preceding discussion.

Note that noncyclicity of a system corresponds to having two or more identical decoupled subsystems imbedded in a system.

Theorem 2.15:

If, given F, there is a G of rank 1 (G a vector denoted by g) such that the pair $\{F,g\}$ is completely reachable then F has a cyclic vector.

Proof: Let χ_F be the characteristic polynomial of the matrix F; i.e., $\chi_F = \det(zI - F)$. The minimal polynomial ψ_F of F is defined to be the monic polynomial of smallest degree such that $\psi_F(F) = 0$.

Suppose $\deg(\psi_F) = \ell < n$ is the degree of the minimal polynomial. Then by the Caley-Hamilton Theorem

$$\psi_F(F)g = F^{\ell}g + \sum_{i=1}^{\ell} \alpha_i F^{\ell-i} g = 0$$

which says that there exists a linear relationship between the powers of F. This is a contradiction to the premise that

$$\rho[g, Fg, \cdots, F^{n-1}g] = n$$

videlicit,

$$g + \alpha_2 Fg + \cdots + \alpha_{n-1}F^{n-1}g \neq 0 \qquad \forall \alpha_i \neq 0$$

Consequently, $\psi_F = \chi_F$; vebatim, F has a cyclic vector.

Theorem 2.15 states that any single input, completely reachable system is cyclic. Unfortunately there is no similar statement for a multiple input completely reachable system.

In our discussion of cyclic spaces and cyclic systems the cyclic vector $x \in X$ that generates a basis for the state space has been assumed to exist, yet it is not clear what x's are cyclic vectors.

Theorem 2.16 (Gopinath):

If Σ is cyclic then almost any $x \in X$ is a cyclic vector; i.e., P(x is a cyclic vector) = 1.

Proof: Let x be selected from a random distribution of vectors in X. In reference (24) it has been shown that there exist only a finite number of invariant subspaces of dim < n under F. (Recall that for a F-invariant subspace W, if $x \in W$ then $Fx \in W$). Now since Σ is cyclic and there exist only a finite number of F-invariant subspace the probability of selecting a x contained in an invariant subspace of dim < n = 0. Hence P(x is a cyclic vector) = 1.

Definition 2.17:

If Σ is a minimal realization and also cyclic, then Σ will be said to be identifiable.

In this section the work of Kalman (18) in the area of minimal realizations and Gopinath's work (25) with regard to cyclic systems has been utilized to determine the necessary criteria for system parameter identification from input/output data. To paraphase Definition 2.17 the necessary criteria for system parameter identification are:

1) The system must be a minimal realization in order to identify it from its input/output map.
2) The system can not assume an arbitrary form; i.e., a maximum of n(r+p) parameters can be identified.
3) The system must be cyclic.

3.0. A PARAMETER IDENTIFICATION ALGORITHM

3.1. External Description Via the Internal Description

In this section the external description of a system will be generated by means of the internal description

$$\begin{aligned} x(k+1) &= Fx(k) + Gu(k) \\ y(k) &= Hx(k) \end{aligned} \tag{3.1}$$

where x is a $n \times 1$ state vector, u is a $r \times 1$ input vector, y is a $p \times 1$ output vector, F is a $n \times n$ state transition matrix, G is a $n \times r$ input weighting matrix and H is a $p \times n$ output observation matrix.

The following requirements will be imposed on Σ;

a) Σ is both completely reachable and completely observable

b) dim Σ is known

c) the matrix F is cyclic

d) H has full rank.

Note that these requirements imply Σ is identifiable.

<u>Definition 3.1</u>:

A selector matrix S is a $\ell \times m$ matrix ($\ell \leq m$) such that multiplication of a $m \times n$ matrix M by S results in ℓ of the rows of M being identically reproduced. The definition implies that S is specified as

$$\begin{aligned} &S(i,j) = 0 \qquad \text{if } j \neq k_i \\ &S(i, k_j) = \delta_{ij} \\ &k_1 < k_2 < \cdots < k_\ell \end{aligned} \tag{3.2}$$

In the subsequent development, we will make use of two particular selector matrices, S_1 and S_2 of dimension $(pn^* \times p(n^*+1))$ defined as

$$S_1 = [I_{pn*} \; \vdots \; 0_{pn* \times p}]$$
$$S_2 = [0_{pn* \times p} \; \vdots \; I_{pn*}] \qquad (3.3)$$

where $n* = n - \rho(H) + 1$.

The following development parallels that of Gopinath (25) and Budin (26) and is repeated here for completeness. Consider iteration of the state equation and the observation equation given by Equation 3.1 so that

$$\begin{aligned} y(k) &= Hx(k) + 0 \\ y(k+1) &= HFx(k) + HGu(k) \\ &\vdots \\ y(k+n*-1) &= HF^{n*-1}x(k) + \sum_{t=0}^{n*-2} HF^{n*-2-t}Gu(k+t) \end{aligned} \qquad (3.4)$$

This can be expressed in matrix notation as

$$\bar{y}_{n*}(k) = \begin{bmatrix} H \\ HF \\ \vdots \\ HF^{n*-1} \end{bmatrix} x(k) + S_1 R_{n*} \bar{u}_{n*}(k) \qquad (3.5)$$

where,

$$\begin{aligned} \bar{y}^T_{n*}(k) &\triangleq [y^T(k),\; y^T(k+1),\; \cdots,\; y^T(k+n*-1)] \\ \bar{u}^T_{n*}(k) &\triangleq [u^T(k),\; u^T(k+1),\; \cdots,\; u^T(k+n*-1)] \end{aligned} \qquad (3.6)$$

and

$$R_{n^*} \triangleq \begin{bmatrix} 0 & 0 & \cdot & \cdot & \cdot & 0 & 0 \\ HG & 0 & \cdot & \cdot & \cdot & 0 & 0 \\ HFG & HG & \cdot & \cdot & \cdot & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ HF^{n^*-1}G & \cdot & \cdot & \cdot & \cdot & HFG & HG \end{bmatrix} \tag{3.7}$$

Since the system is completely observable

$$\rho[H^T, F^TH^T, \cdots, (F^T)^{n^*-1}H^T] = n$$

and from the definition of a selector matrix there exists a S such that,

$$S \begin{bmatrix} H \\ HF \\ \cdot \\ \cdot \\ \cdot \\ HF^{n^*-1} \end{bmatrix} = I_{n \times n} \tag{3.8}$$

Equation 3.8 follows from the fact that coordination of the state space is irrelevant; i.e., any basis may be chosen.

Multiplication of Equation 3.5 by S then results in,

$$S\bar{y}_{n^*}(k) = x(k) + SS_1R_{n^*}\bar{u}_{n^*}(k) \tag{3.9}$$

Iterating one more step,

$$S\bar{y}_{n^*}(k+1) = F[S\bar{y}_{n^*}(k) - SS_1R_{n^*}\bar{u}_{n^*}(k)] + Gu(k) + SS_1R_{n^*}\bar{u}_{n^*}(k+1) \tag{3.10}$$

which in turn implies,

$$S\bar{y}_{n*}(k+1) = FS\bar{y}_{n*}(k) - FSS_1R_{n*}\bar{u}_{n*}(k) + SS_2R_{n*}\bar{u}_{n*}(k)$$

$$\triangleq [F \,\vdots\, R] \begin{bmatrix} S\bar{y}_{n*}(k) \\ \bar{u}_{n*}(k) \end{bmatrix} \tag{3.11}$$

where $R = -FSS_1R_{n*} + SS_2R_{n*}$.

This is an external description of the system Σ which does not involve the state space X, and in principle can be solved for the triple (F,G,H).

Solution for the triple (F,G,H) is possible whenever the product matrix

$$\mathscr{P}B_{n*}(k) \triangleq \left[\begin{array}{c|c} S & 0 \\ \hline 0 & I_{rn*} \end{array}\right] \begin{bmatrix} \bar{y}_{n*}(k) & \cdots & \bar{y}_{n*}(k+n+rn*-1) \\ \bar{u}_{n*}(k) & \cdots & \bar{u}_{n*}(k+n+rn*-1) \end{bmatrix} \tag{3.12}$$

is row equivalent to a (n+rn*) identity matrix in the expression

$$SA_{n*}(k+1) \triangleq S[\bar{y}_{n*}(k+1) \;\ldots\; \bar{y}_{n*}(k+n+rn*)] = [F \,\vdots\, R]\mathscr{P}B_{n*}(k) \tag{3.13}$$

Here the definition of $\mathscr{P}$, $A_{n*}(k+1)$, and $B_{n*}(k)$ follows from Equations 3.12 and 3.13.

If $\mathscr{P}B_{n*}(k)$ is nonsingular one can solve for $[F \,\vdots\, R]$. As shown in Reference (25) pp. 36-38, G can be obtained from $[F \,\vdots\, R]$ using the relationship

$$G = R_0 + FR_1 + \cdots + F^{n*-1}R_{n*-1} \tag{3.14}$$

where R is partitioned as $[R_0 \,\vdots\, R_1 \,\vdots\, \cdots \,\vdots\, R_{n*-1}]$. R_i $n \times r$ $\forall_i$. The matrix H is found by solving the n equations resulting from Equation 3.8.

3.2. A Canonical Form For Single Input/Single Output Systems

Our attention is first focused on the simplest case, that of the single input/single output system. As pointed out in Section 2.3 all the parameters in triple (F,G,H) cannot be identified from knowledge of the external description. With this in mind, the following definition constrains the system to assume a certain canonical form.

Definition 3.2:

The single input/single output identification canonical form consists of the triple (F,G,H) specified by,

$$F = \begin{bmatrix} 0 & 1 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & 1 & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdot & \cdot & \cdot & 1 \\ -\alpha_n & -\alpha_{n-1} & -\alpha_{n-2} & \cdot & \cdot & \cdot & -\alpha_1 \end{bmatrix} \tag{3.15}$$

$$G^T = [\beta_1 \quad \beta_2 \quad \cdot \quad \cdot \quad \cdot \quad \beta_n]$$

$$H = [1 \quad 0 \quad \cdot \quad \cdot \quad \cdot \quad 0\,]$$

where the α_i's and β_i's are the 2n parameters to be identified. As shown in Theorem 2.9 any single input/single output minimal realization can be expressed with regard to the basis of the above identification canonical form.

The determination of an appropriate selector matrix can be readily ascertained by formulation of the observability matrix,

$$\mathcal{O} = \begin{bmatrix} H \\ HF \\ \cdot \\ \cdot \\ \cdot \\ HF^{n-1} \end{bmatrix} \triangleq I_n \tag{3.16}$$

Then from Equation 3.8 clearly S = I. Equation 3.11 reduces to,

$$[\bar{y}_n(k+1) \cdots \bar{y}_n(k+2n)] = [F \vdots R] \begin{bmatrix} \bar{y}_n(k) \cdots \bar{y}_n(k+2n-1) \\ \bar{u}_n(k) \cdots \bar{u}_n(k+2n-1) \end{bmatrix} \tag{3.17}$$

which has solution whenever $B_{n*}(k)$ is nonsingular.

3.3. A Canonical Form For Multiple Input/Multiple Output Systems

The discussion in Section 2.3 points out that all the parameters in the triple (F,G,H) cannot be deduced from the external description. Unlike the phase-variable canonical form for the single input/single output system, the corresponding canonical forms for multivariable systems are not unique. In this section a canonical form for a multiple input/multiple output system will be presented.

Definition 3.3:

The multiple input/multiple output identification canonical form consisting of the triple (F,G,H) is specified by,

$$F = \left[\begin{array}{ccccc|ccccc} 0 & \cdot & \cdot & \cdot & 0 & 1 & \cdot & \cdot & \cdot & 0 \\ \cdot & & & & \cdot & 0 & 1 & \cdot & \cdot & 0 \\ \cdot & & & & \cdot & \cdot & & \cdot & & \cdot \\ \cdot & & & & \cdot & \cdot & & & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & 0 & 0 & \cdot & \cdot & \cdot & 1 \\ \hline & & F_1 & & & & & F_2 & & \end{array}\right] \quad \begin{array}{l} p \text{ columns (left)}, \; m \text{ columns (right)} \\ m \text{ rows} \end{array} \tag{3.18}$$

$$G = \begin{bmatrix} \beta_{11} & \cdot & \cdot & \cdot & \beta_{1r} \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ \beta_{n1} & \cdot & \cdot & \cdot & \beta_{nr} \end{bmatrix} \qquad H = [I_p \mid 0]$$

where m = n-p. Observe that the m columns and rows form a m x m identity matrix; i.e.,

$$F = \left[\begin{array}{c|c} 0 & I_m \\ \hline F_1 & F_2 \end{array}\right] \tag{3.19}$$

Note that the canonical form is devised so that the first n rows of the observability matrix are not only linearly independent but identically equal to a $n \times n$ identity matrix,

$$\mathcal{O} = \begin{bmatrix} H \\ HF \\ \cdot \\ \cdot \\ \cdot \\ HF^{n-p} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & 1 & 0 & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdot & \cdot & \cdot & 1 \\ \hline x & x & x & x & x & x & x \\ x & x & x & x & x & x & x \end{bmatrix} \quad (3.20)$$

The free parameters are then the nr parameters of the input weighting matrix and the np parameters associated with the matrices F_1 and F_2. The corresponding selector matrix needed for application of the algorithm can be easily computed from Equation 3.7 as

$$S = [I_n \mid 0] \quad (3.21)$$

The required linear transformation, A, needed to put a triple (F,G,H) into the identifiable canonical form $(\mathcal{F},\mathcal{G},\mathcal{H})$ can be easily calculated. Consider the two observability matrices where the first n rows on the right hand side are linearly independent,

$$\begin{bmatrix} \mathcal{H} \\ \mathcal{H}\mathcal{F} \\ \cdot \\ \cdot \\ \cdot \\ \mathcal{H}\mathcal{F}^{n-p} \end{bmatrix} = \begin{bmatrix} H \\ HF \\ \cdot \\ \cdot \\ \cdot \\ HF^{n-p} \end{bmatrix} A^{-1} \quad (3.22)$$

Since the first n rows of the left hand matrix are equal to the $n \times n$ identity matrix one has

$$\begin{bmatrix} A_{n \times n} \\ \hline x \quad x \quad x \\ x \quad x \quad x \end{bmatrix} = \begin{bmatrix} H \\ HF \\ \vdots \\ \hline HF^{n-p} \end{bmatrix} \tag{3.23}$$

Here again Equation 3.13 has solution whenever $\mathscr{S} B_{n*}(k)$ is nonsingular. $\mathscr{S} B_{n*}(k)$ is nonsingular whenever $B_{n*}(k)$ has rank n+rn* since $\mathscr{S}$ always has rank n+rn* for both the single input/single output canonical form and the multiple input/multiple output canonical form.

Theorem 3.4:

The matrix $\mathscr{S} B_{n*}(k)$ is nonsingular almost surely if Σ is cyclic and the u(i)'s are independent random variables taken from a nonlattice distribution.

Proof: The matrix $\mathscr{S} B_{n*}(k)$ can be written as,

$$\left[\begin{array}{c|c} S & 0 \\ \hline 0 & I_{rn*} \end{array}\right] \cdot \left[\begin{array}{c|c} \begin{matrix} H \\ HF \\ \vdots \\ HF^{n*-1} \end{matrix} & 0 \\ \hline 0 & I_{rn*} \end{array}\right] \cdot \begin{bmatrix} x(k) & \cdots & x(k \underline{+} n + rn* - 1) \\ \bar{u}_{n*}(k) & \cdots & \bar{u}_{n*}(k+n+rn*-1) \end{bmatrix}$$

$$= \left[\begin{array}{c|c} I_n & 0 \\ \hline 0 & I_{rn*} \end{array}\right] \cdot \begin{bmatrix} x(k) & \cdots & x(k+n+rn*-1) \\ \bar{u}_{n*}(k) & \cdots & \bar{u}_{n*}(k+n+rn*-1) \end{bmatrix} \tag{3.24}$$

If

$$\rho \begin{bmatrix} x(k) \ . \ . \ . \ x(k+n+rn^*-1) \\ \bar{u}_{n^*}(k) \ . \ . \ . \ \bar{u}_{n^*}(k+n+rn^*-1) \end{bmatrix} = \rho[D_{n^*}(k)] = n + rn^* \qquad (3.25)$$

almost surely then $\mathscr{P}B_{n^*}(k)$ is nonsingular. Partition the matrix $D_{n^*}(k)$ so that

$$D_{n^*}(k) = \left[\begin{array}{c} x(k) \ . \ . \ . \ x(k+n+rn^*-1) \\ \hline \bar{u}_{n^*}(k) \ . \ . \ . \ \bar{u}_{n^*}(k+n+rn^*-1) \end{array}\right] \qquad (3.26)$$

If Σ is cyclic then by Theorem 2.16 almost any x is a cyclic vector so that

$$\rho[x(k) \ . \ . \ . \ x(k+n+rn^*-1)] = n \quad \text{almost surely.}$$

Suppose the u(i)'s are taken from a nonlattice distribution, consider the matrix

$$U = \begin{bmatrix} u(k) & u(k+1) & \cdots & u(k+n+r-1) & \cdots & u(k+n+rn+-1) \\ u(k+1) & u(k+2) & \cdots & u(k+n^*r) & \cdots & u(k+n+rn^*) \\ \cdot & \cdot & \cdots & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot & \cdots & \cdot \\ u(k+n^*-1) & u(k+n^*) & \cdots & \cdot & \cdots & u(k+n+n^*(1+r)-2) \end{bmatrix}$$

(3.27)

then $\rho[U] = rn^*$ almost surely.

Since the u(i)'s are independent random variables taken from a nonlattice distribution the $P(u(i)=u(j)) = 0 \quad \forall i \neq j$. If any partitioned $(n^*r) \times (n+rn^*)$ matrix has rank n^*r then $\rho[U] = n^*r$. Consider the $(n^*r) \times (n^*r)$ matrix

$$U_{n^*} = \begin{bmatrix} u(k) & \cdots & u(k+n^*-1) \\ u(k+1) & \cdots & u(k+n^*) \\ \vdots & & \vdots \\ u(k+n^*-1) & \cdots & u(k+2n^*-2) \end{bmatrix} \tag{3.28}$$

Suppose $\rho[U_{n^*}] < n^*r$ then two or more columns of the matrix are linearly dependent. Let the last column be made up of linear combinations of the first n^*-1 columns; viz,

$$\begin{bmatrix} u(k+n^*-1) \\ \vdots \\ u(k+2n^*-2) \end{bmatrix} = \sum_{i=1}^{n^*-1} \alpha_i \cdot \begin{bmatrix} u(k+n^*-1-i) \\ \vdots \\ u(k+2n^*-2-i) \end{bmatrix} \tag{3.29}$$

The α_i's can be uniquely determined by solving the first n^*-1 equations. The remaining term $u(k+2n^*-2)$ then can be expressed as a unique linear combination of the α_i's so that

$$u(k+2n^*-2) = \sum_{i=1}^{n^*-1} \alpha_i \cdot u(k+2n^*-2-i) \tag{3.30}$$

which implies that $u(k+2n^*-2)$ is dependent on the previous n^*-1 random variables which contradicts the premise that the random variables be independent. Consequently, $\rho[U_{n^*}] = \rho[U] = n^*r$ almost surely. The rank of $B_{n^*}(k)$ then is $n+n^*r$ which implies that $\mathscr{P}B_{n^*}(k)$ is nonsingular almost surely.

3.4. Identification From Noisy Input/Output Observations

In the previous section we have shown that given input and output observations one can identify the corresponding unknown parameters in the F and G matrices. The more realistic situation, depicted in Figure 3.1, is that where the measurements are corrupted with noise.

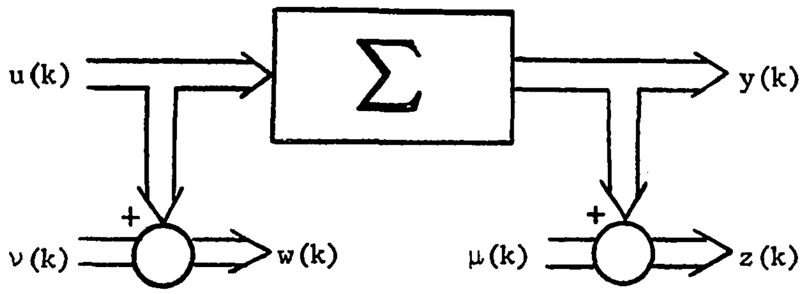


Figure 3.1. Discrete linear system measurement model

The available measurements are defined as

$$\begin{aligned} w(k) &= u(k) + \nu(k) \\ z(k) &= y(k) + \mu(k) \end{aligned} \tag{3.31}$$

where $\nu(k)$ is a r-dimensional zero mean gaussian white sequence with covariance

$$E\{\nu(j)\nu^T(k)\} = Q(k)\delta_{jk} \quad \forall\ j,k = 0, 1, \ldots$$

and $\mu(k)$ is a p-dimensional zero mean gaussian white sequence with covariance

$$E\{\mu(j)\mu^T(k)\} = R(k)\delta_{jk} \quad \forall\, j,k = 0, 1, \ldots$$

We allow the possibility that $\{\nu(k),\ k \in Z^+\}$ and $\{\mu(k),\ k \in Z^+\}$ may be correlated with each other, the cross-covariance matrix being

$$E\{\mu(k)\nu^T(k)\} = P(k)\delta_{jk} \quad \forall\, j, k = 0, 1, \ldots$$

The noise processes $\nu(k)$ and $\mu(k)$ are assumed to be independent of the input and output.

Our objective is to determine an estimate of the triple (F,G,H) namely $(\hat{F},\hat{G},\hat{H})$ from the available measurements.

<u>Definition 3.5</u>:

An estimate $\hat{\theta}_N$ of a parameter θ is said to be consistent if

$$P(\lim_{N\to\infty} \hat{\theta}_N = \theta) = 1 \tag{3.32}$$

In the following discussion we will make use of the following definitions,

$$\begin{aligned}
\bar{z}_{n*}^T(k) &\triangleq [z^T(k)\quad z^T(k+1)\quad \ldots\quad z^T(k+n*-1)] \\
\bar{w}_{n*}^T(k) &\triangleq [w^T(k)\quad w^T(k+1)\quad \ldots\quad w^T(k+n*-1)] \\
\bar{\nu}_{n*}^T(k) &\triangleq [\nu^T(k)\quad \nu^T(k+1)\quad \ldots\quad \nu^T(k+n*-1)] \\
\bar{\mu}_{n*}^T(k) &\triangleq [\mu^T(k)\quad \mu^T(k+1)\quad \ldots\quad \mu^T(k+n*-1)]
\end{aligned} \tag{3.33}$$

The following covariance matrices are also utilized,

$$E\{\bar{\nu}_{n*}(i)\ \bar{\nu}_{n*}^T(j)\} \triangleq \bar{Q}_{n*}(i-j)$$
$$E\{\bar{\mu}_{n*}(i)\ \bar{\mu}_{n*}^T(j)\} \triangleq \bar{R}_{n*}(i-j) \tag{3.34}$$
$$E\{\bar{\mu}_{n*}(i)\ \bar{\nu}_{n*}^T(j)\} \triangleq \bar{P}_{n*}(i-j)$$

Note that

$$\bar{Q}_{n*}(1) = \bar{\nu}_{n*}(k+i+1)\bar{\nu}_{n*}^T(k+i) \quad \forall\ i, k = 0, 1, 2, \ldots$$
$$\bar{R}_{n*}(0) = \bar{\mu}_{n*}(k)\bar{\mu}_{n*}^T(k) \quad \forall\ k = 0, 1, 2, \ldots$$

Motivation for using a consistent estimator follows. Suppose $\mathscr{S}B_{n*}(k)$ is nonsingular in the noise free case. Then

$$S[\bar{y}_{n*}(k+1)\ldots\bar{y}_{n*}(k+n+rn*)]\left\{\mathscr{S}\begin{bmatrix}\bar{y}_{n*}(k) & \ldots & \bar{y}_{n*}(k+n+rn*-1)\\ \bar{u}_{n*}(k) & \ldots & \bar{u}_{n*}(k+n+rn*-1)\end{bmatrix}\right\}^{-1} = [F, R] \tag{3.35}$$

for any k. Consequently,

$$\left\{\frac{1}{N}\sum_{k=1}^{N} S\ A_{n*}(k+1)\right\}\left\{\frac{1}{N}\sum_{k=1}^{N}\mathscr{S}B_{n*}(k)\right\}^{-1} = [F, R] \tag{3.36}$$

and

$$\left\{\frac{1}{N}\sum_{k=1}^{N} S\ A_{n*}(k+1)B_{n*}^T(k)\mathscr{S}^T\right\}\left\{\frac{1}{N}\sum_{k=1}^{N}\mathscr{S}B_{n*}(k)B_{n*}^T(k)\mathscr{S}^T\right\}^{-1} = [F, R] \tag{3.37}$$

It is reasonable to expect similar results in the noisy observation case where the $\bar{y}_{n*}(k)$'s are replaced by $\bar{z}_{n*}(k)$'s and the $\bar{u}_{n*}(k)$'s are replaced with $\bar{w}_{n*}(k)$'s.

<u>Proposition 3.6</u>:

The estimate

$$\left\{ \frac{1}{N}\sum_{k=1}^{N} SA_{n*}(k+1)B_{n*}^{T}(k)\mathscr{P}^{T} - (n+rn*)S[\bar{R}_{n*}(1) \mid \bar{P}_{n*}(1)]\mathscr{P}^{T} \right\}$$

$$\left\{ \frac{1}{N}\sum_{k=1}^{N} \mathscr{P}B_{n*}(k)B_{n*}^{T}(k)\mathscr{P}^{T} - (n+rn*)\mathscr{P}\left[\begin{array}{c|c} \bar{R}_{n*}(0) & \bar{P}_{n*}(0) \\ \hline \bar{P}_{n*}^{T}(0) & \bar{Q}_{n*}(0) \end{array}\right]\mathscr{P}^{T} \right\}^{-1} = [\hat{F}_N, \hat{R}_N] \tag{3.38}$$

is an unbiased consistent estimate as $N \to \infty$.

Proof: Rewriting Equation 3.38 in terms of the noisy observations one has,

$$\left\{ \frac{1}{N}\sum_{k=1}^{N} S\left[\sum_{i=1}^{n+rn*} \bar{z}_{n*}(k+i)\bar{z}_{n*}^{T}(k+i-1) \;\Big|\; \sum_{i=1}^{n+rn*} \bar{z}_{n*}(k+i)\bar{w}_{n*}^{T}(k+i-1) \right]\mathscr{P}^{T} - (n+rn*)S[\bar{R}_{n*}(1) \mid \bar{P}_{n*}(1)]\mathscr{P}^{T} \right\}$$

$$\left\{ \frac{1}{N}\sum_{k=1}^{N} \mathscr{P}\left[\begin{array}{c|c} \sum_{i=1}^{n+rn*} \bar{z}_{n*}(k+i-1)\bar{z}_{n*}^{T}(k+i-1) & \sum_{i=1}^{n+rn*} \bar{z}_{n*}(k+i-1)\bar{w}_{n*}^{T}(k+i-1) \\ \hline \sum_{i=1}^{n+rn*} \bar{w}_{n*}(k+i-1)\bar{z}_{n*}^{T}(k+i-1) & \sum_{i-1}^{n+rn*} \bar{w}_{n*}(k+i-1)\bar{w}_{n*}^{T}(k+i-1) \end{array}\right]\mathscr{P}^{T} \right.$$

$$- (n+rn^*)\mathscr{S}\left[\begin{array}{c|c} \bar{R}_{n^*}(0) & \bar{P}_{n^*}(0) \\ \hline \bar{P}_{n^*}^T(0) & \bar{Q}_{n^*}(0) \end{array}\right]\mathscr{S}^T \Bigg\}^{-1} = [\hat{F}_N, \hat{R}_N] \qquad (3.39)$$

Taking the limit as $N \to \infty$ (the expectation of both sides) results in

$$E\left\{ S[\bar{y}_{n^*}(k+1)\ldots\bar{y}_{n^*}(k+n+rn^*)] \begin{bmatrix} \bar{y}_{n^*}^T(k) & \bar{u}_{n^*}^T(k) \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \bar{y}_{n^*}^T(k+n+rn^*-1) & \bar{u}_{n^*}^T(k+n+rn^*-1) \end{bmatrix} \mathscr{S}^T \right\}$$

$$E\left\{ \mathscr{S}\begin{bmatrix} \bar{y}_{n^*}(k)\ldots\bar{y}_{n^*}(k+n+rn^*-1) \\ \bar{u}_{n^*}(k)\ldots\bar{u}_{n^*}(k+n+rn^*-1) \end{bmatrix} \begin{bmatrix} \bar{y}_{n^*}^T(k) & \bar{u}_{n^*}^T(k) \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \bar{y}_{n^*}^T(k+n+rn^*-1) & \bar{u}_{n^*}^T(k+n+rn^*-1) \end{bmatrix} \mathscr{S}^T \right\}^{-1}$$

$$= E\{SA_{n^*}(k+1)B_{n^*}^T(k)\mathscr{S}^T\}E\{\mathscr{S}B_{n^*}(k)B_{n^*}^T(k)\mathscr{S}^T\}^{-1} = [\hat{F}_\infty \ , \ \hat{R}_\infty] \qquad (3.40)$$

By Equation 3.37 this implies that

$$[\hat{F}_\infty \ , \ \hat{R}_\infty] = [F \ , \ R] \qquad (3.41)$$

so the estimator is consistent.

4.0. APPLICATION OF THE PARAMETER IDENTIFICATION ALGORITHM

This section will investigate the application of the algorithm developed in Chapter 3 with regard to the identification of the parameters that make up the equations of motion of an aircraft. Flight parameter data for the F-105B were used to determine the behavior of the algorithm under simulated flight conditions. The following tables taken from reference (27) tabulate the necessary parameters needed to specify both the linearized longitudinal and lateral equations of motion for the F-105B.

4.1. Identification of the Longitudinal Equations of Motion of an Aircraft

The longitudinal equations of motion of an aircraft can be expressed as

$$\begin{bmatrix} \dot{\theta} \\ \dot{u} \\ \dot{w} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ -g\cos\theta_o & X_u & X_w & -W_o \\ \dfrac{-g\sin\theta_o}{1-Z_{\dot{w}}} & \dfrac{Z_u}{1-Z_{\dot{w}}} & \dfrac{Z_w}{1-Z_{\dot{w}}} & \dfrac{U_o}{1-Z_{\dot{w}}} \\ \dfrac{-M_{\dot{w}}\, g\sin\theta_o}{1-Z_{\dot{w}}} & M_u + \dfrac{M_{\dot{w}} Z_u}{1-Z_{\dot{w}}} & \dfrac{M_{\dot{w}} Z_w}{1-Z_{\dot{w}}} + M_w & \dfrac{M_{\dot{w}} U_o}{1-Z_{\dot{w}}} + M_q \end{bmatrix} \begin{bmatrix} \theta \\ u \\ w \\ q \end{bmatrix} + \begin{bmatrix} 0 \\ X_{\delta_e} \\ \dfrac{Z_{\delta_e}}{1-Z_{\dot{w}}} \\ \dfrac{M_{\dot{w}} Z_{\delta_e}}{1-Z_{\dot{w}}} + M_{\delta_e} \end{bmatrix} \delta_e \quad (4.1)$$

Table 4.1. Longitudinal dimensional derivatives for the F-105B

Note: Data for body-fixed stability axis

	1 Takeoff	2 Start Cruise
h(ft)	Sea Level	35,000
M(-)	0.261	0.9
u_o(ft/sec)	288.57631	868.10036
W_o(ft/sec)	37.47952	109.66658
α_o(deg)	7.4	7.2
γ_o(deg)	10.0	0.0
θ_o(deg)	17.4	7.2
X_u(1/sec)	-0.029	-0.00582
X_w(1/sec)	0.0793	0.00693
X_{δ_e}[(ft/sec^2)/rad]	0.0	0.0
Z_u(1/sec)	-0.1585	-0.01386
$Z_{\dot{w}}$(-)	0.0	0.0
Z_w(1/sec)	-0.311	-0.4
Z_{δ_e}[(ft/sec^2)/rad]	-17.3	-65.19
M_u(1/sec-ft)	-0.0000119	0.0
$M_{\dot{w}}$(1/ft)	-0.000259	-0.000117
M_w(1/sec-ft)	-0.00575	-0.00819
M_q(1/sec)	-0.345	-0.485
M_{δ_e}(1/sec^2)	-2.60	-12.03

FLIGHT CONDITION		
3 End Cruise	4 Power Approach	5 Vmax Clean
35,000	Sea Level	40,000
0.9	0.241	2.1
868.47788	267.89290	2026.21364
106.63568	24.38016	123.92854
7.0	5.2	3.5
0.0	-5.0	0.0
7.0	0.2	3.5
-0.00565	-0.0263	-0.00751
0.0264	0.086	0.0132
0.0	0.0	0.0
-0.0527	-0.1719	-0.0265
0.0	0.0	0.0
-0.466	-0.406	-0.590
-75.19	-19.88	-135.9
0.0	-0.0000101	-0.0000198
-0.000117	-0.000259	-0.00000535
-0.00468	-0.00324	-0.01252
-0.485	-0.319	-0.303
-12.03	-2.703	-21.0

Table 4.2. Lateral dimensional derivatives for the F-105B

Note: Lateral data not available for flight conditions 1 and 2

	FLIGHT CONDITION		
	3 End Cruise	4 Power Approach	5 Vmax Clean
h(ft)	35,000	Sea Level	40,000
V_{T_o}(ft/sec)	875.0	269.0	2030.0
Y_V(1/sec)	-0.1497	-0.1878	-0.213
$Y^*_{\delta_a}$ [(1/sec)/rad]	-0.00173	-0.0021	-0.00221
$Y^*_{\delta_r}$ [(1/sec)/rad]	0.0234	0.0241	0.0837
L'_β(1/sec^2)	-41.1	-21.5	-139.8
L'_p(1/sec)	-2.8	-1.185	-3.14
L'_r(1/sec)	1.709	1.251	1.966
L'_{δ_a} (1/sec^2)	10.71	3.72	26.5
L'_{δ_r} (1/sec^2)	14.37	2.86	12.97
N'_β(1/sec)	12.39	4.38	18.81
N'_p(1/sec)	0.324	0.0725	0.1341
N'_r(1/sec)	-0.382	-0.242	-0.386
N'_{δ_a} (1/sec^2)	-1.086	-0.277	-1.339
N'_{δ_r} (1/sec^2)	-4.71	-0.975	-1.989

where θ is the pitch angle of the aircraft, u is the perturbed linear velocity along the X-axis, w is the perturbed linear velocity along the Z-axis, q is the pitch rate (angular velocity about the Y-axis) and δ_e is the elevator surface deflection. See Figure 4.1.

The measurement parameters are θ, u, w so that the measurement equation assumes the form

$$\begin{bmatrix} \theta \\ u \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ u \\ w \\ q \end{bmatrix} \tag{4.2}$$

The system described by the above equations is an identifiable system in terms of Definition 2.17.

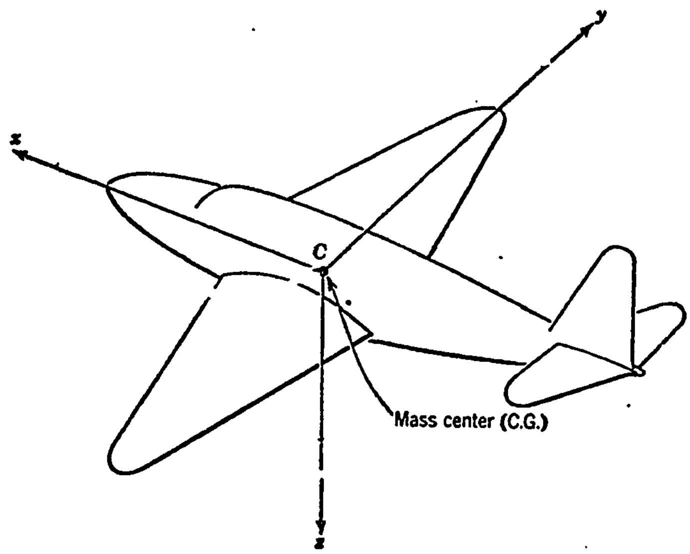


Figure 4.1. Aircraft axes definition

Example 4.1: Flight Condition 1, Takeoff

In this example use of the algorithm specified by Equation 3.38 will be explained step by step. Equations 4.1 and 4.2 are continuous equations of motion, consequently it will be necessary to discretize the model and place the resulting discrete equations in the canonical form gives by Equation 3.18. The continuous longitudinal equations have the form,

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \tag{4.3}$$

For flight condition 1 the A and B matrices are,

$$A = \begin{bmatrix} 0.0 & 0.0 & 0.0 & 1.0 \\ -0.3072654E\ 02 & -0.2900000E\text{-}01 & 0.7930000E\text{-}01 & -0.3747952E\ 02 \\ -0.9629114E\ 01 & -0.1585000E\ 00 & -0.3110000E\ 00 & 0.2885763E\ 03 \\ 0.2590000E\text{-}03 & 0.2900000E\text{-}04 & -0.5750000E\text{-}02 & -0.4197400E\ 00 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.0 \\ 0.0 \\ -0.1730000E\ 02 \\ -0.2595519E\ 01 \end{bmatrix} \tag{4.4}$$

There are two modes of oscillation in the continuous case; the phugoid mode, which is a low frequency lightly damped mode and the short period mode, which is a high frequency heavily damped mode.

Choice of a sampling time of 0.783 seconds places the continuous poles in the z-plane so that there is approximately 12 samples per cycle of the short period mode. For a sampling time of 0.783 seconds the discrete equations have the following F and G matrices,

$$F = \begin{bmatrix} 0.1003290\text{E } 01 & 0.6727690\text{E-}04 & -0.1344116\text{E-}02 & 0.5665131\text{E } 00 \\ -0.2410219\text{E } 02 & 0.9714125\text{E } 00 & 0.1078716\text{E } 00 & -0.2342473\text{E } 02 \\ -0.4428109\text{E } 01 & -0.9057302\text{E-}01 & 0.4260424\text{E } 00 & 0.1417087\text{E } 03 \\ 0.1102219\text{E-}01 & 0.2275202\text{E-}03 & -0.2834095\text{E-}02 & 0.3751003\text{E } 00 \end{bmatrix}$$

$$G = \begin{bmatrix} -0.1133112\text{E } 01 \\ 0.4614477\text{E } 02 \\ -0.2939646\text{E } 03 \\ -0.7239227\text{E } 00 \end{bmatrix} . \qquad (4.5)$$

Using the linear transformation matrix A specified by Equation 3.23 the matrices F and G given in Equation 4.5 can be put into the canonical form required by the algorithm. The linear transformation matrix A is given by,

$$A = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.1003290\text{E } 01 & 0.6727690\text{E-}04 & -0.1344116\text{E-}02 & 0.5665131\text{E } 00 \end{bmatrix} \qquad (4.6)$$

For a sampling time of 0.783 seconds, the canonical system matrices for flight condition 1 are,

$$\mathscr{F} = \begin{bmatrix} 0.0 & 0.0 & 0.0 & 1.0 \\ 0.1738291E\ 02 & 0.9741944E\ 00 & 0.5229372E\text{-}01 & -0.4134906E\ 02 \\ -0.2553931E\ 03 & -0.1074018E\ 00 & 0.7622623E\ 00 & 0.2501421E\ 03 \\ -0.2564270E\text{-}01 & 0.3135588E\text{-}03 & -0.2122424E\text{-}02 & 0.1039389E\ 01 \end{bmatrix}$$

$$\mathscr{G} = \begin{bmatrix} -0.1133112E\ 01 \\ 0.4614477E\ 02 \\ -0.2937646E\ 03 \\ -0.1148994E\ 01 \end{bmatrix} \qquad \mathscr{H} = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 \end{bmatrix} \qquad (4.7)$$

Application of the parameter identification procedure will be made with respect to identifying the parameters in Equation 4.7.

In order to implement Equation 3.38 the dimensions of the matrices, S, $A_{n*}(k+1)$, $B_{n*}(k)$, $\mathscr{S}$, $\bar{R}_{n*}(1)$, $\bar{P}_{n*}(1)$, and $\bar{Q}_{n*}(0)$ must be known. In terms of the known system dimensions n, p, r, n* the matrices have the following dimensions,

S(n, pn*)

$A_{n*}(k+1)$ (pn*, n+rn*)

$B_{n*}(k)$ ((p+r)n*, n+rn*)

$\mathscr{S}$(n+rn*, (p+r)n*)

$\bar{R}_{n*}(1)$ (pn*, pn*)

$\bar{P}_{n*}(1)$ (pn*, rn*)

$\bar{Q}_{n*}(0)$ (rn*, rn*)

In order to start the algorithm the matrices $A_{n*}(k+1)$ and $B_{n*}(k)$ must first be constructed from the first n+n*(r+1) input and output observations. Thereafter the algorithm is recursive in the sense that each additional observation updates $A_{n*}(k+1)$ and $B_{n*}(k)$ resulting in a sliding data window.

Flight conditions at takeoff were simulated using an IBM 360-65 computer. The initial state of the system was taken to be zero. Note that the choice of initial conditions is immaterial in the application of the algorithm. The computer simulation (see the Appendix) generated the output vector given a gaussian input having zero mean with variance of 0.05 radians squared as an elevator input via the triple $(\mathscr{F},\mathscr{G},\mathscr{H})$. The steady state convariance matrix of the output was computed from the equation

$$P(k+1) = \mathscr{F}P(k)\mathscr{F}^T + \mathscr{G}R(k)\mathscr{G}^T$$
$$P(0) = 0$$

and was used to calculate the amount of noise to be added to each output observation, the ratio of the variance of the observation to the variance of the added noise being the measure of S/N ratio. In evaluating the performance of the estimator with various signal to noise ratios, the normalized state error in properly identifying the system will be defined as

$$\text{State Error} = \frac{\|x - \hat{x}\|}{\|x\|} \qquad (4.8)$$

where x is the state vector and $\hat{x}$ is the state vector resulting from the estimated triple $(\hat{\mathscr{F}},\hat{\mathscr{G}},\hat{\mathscr{H}})$ being driven by the identical input driving the triple $(\mathscr{F},\mathscr{G},\mathscr{H})$. The absolute parameter errors in the elements of $(\hat{\mathscr{F}},\hat{\mathscr{G}})$ are defined by the following equations

$$\begin{aligned} \tilde{f}_{ij} &= |f_{ij} - \hat{f}_{ij}| \\ \tilde{g}_{ij} &= |g_{ij} - \hat{g}_{ij}| \end{aligned} \tag{4.9}$$

Given the input observations and output observations Equation 3.38 and Equation 3.14 were implemented in order to arrive at the estimated matrices $\hat{\mathscr{F}}$ and $\hat{\mathscr{G}}$. Figure 4.2 shows the convergence rate of the state error for various signal to noise ratios. Tables 4.3, 4.4, and 4.5 tabulate the absolute parameter errors as a function of number of iterations.

4.2. Identification of the Lateral Equations of Motion of an Aircraft

The lateral equations of motion of an aircraft can be expressed as

$$\begin{bmatrix} \dot{\phi} \\ \dot{p} \\ \dot{r} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & \tan\theta_o & 0 \\ 0 & L'_p & L'_r & L'_\beta \\ 0 & N'_p & N'_r & N'_\beta \\ \dfrac{-g\cos\theta_o}{V_{T_o}} & \dfrac{W_o}{V_{T_o}} & -\dfrac{U_o}{V_{T_o}} & Y_V \end{bmatrix} \begin{bmatrix} \varphi \\ p \\ r \\ \beta \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ L'_{\delta_a} & L'_{\delta_r} \\ N'_{\delta_a} & N'_{\delta_r} \\ Y^*_{\delta_a} & Y^*_{\delta_r} \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \tag{4.10}$$

Here φ is the roll angle of the aircraft, p is the roll rate about the X-axis, r is the yaw rate about the Z-axis, β is the sideslip angle of the aircraft, δ_a is the aileron control surface deflection, and δ_r is the rudder deflection. Measureable parameters in the lateral case are φ, p, and r so that the measurement equation becomes,

$$\begin{bmatrix} \varphi \\ p \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \varphi \\ p \\ r \\ \beta \end{bmatrix} \tag{4.11}$$

The lateral equations of motion represent an identifiable multiple input/multiple output system.

Example 4.2: Flight Condition 4, Power Approach

Using a sampling time of 0.318 seconds Equations 4.10 and 4.11 have the discrete time canonical triple,

$$\mathscr{F} = \begin{bmatrix} 0.0 & 0.0 & 0.0 & 1.0 \\ 0.1802022\text{ E }00 & 0.1014800\text{ E }01 & 0.1268955\text{ E }-02 & -0.2352869\text{ E }00 \\ 0.1416865\text{ E }00 & -0.6111120\text{ E }-01 & 0.9926234\text{ E }00 & 0.1836178\text{ E }00 \\ -0.2762777\text{ E }00 & 0.1624717\text{ E }01 & 0.1921763\text{ E }00 & 0.1071259\text{ E }01 \end{bmatrix}$$

$$\mathscr{G} = \begin{bmatrix} 0.6551825\text{ E }00 & 0.1850717\text{ E }00 \\ -0.3037358\text{ E }-01 & -0.1942935\text{ E }00 \\ 0.2932957\text{ E }00 & 0.1817910\text{ E }00 \\ 0.2319299\text{ E }00 & -0.5567527\text{ E }00 \end{bmatrix}$$

$$\mathcal{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{4.12}$$

Again our objective is to identify the unknown parameters in the triple $(\mathcal{F},\mathcal{G},\mathcal{H})$. Proceeding as in Example 4.1, Figure 4.3 shows the convergence rate of the states error while Tables 4.6, 4.7, and 4.8 tabulate the parameter matrix error as a function of S/N ratio and number of iterations. Here both δ_a and δ_r are zero mean gaussian sequences having variances equal to 0.05 rad. squared.

In both the lateral and longitudinal simulations both the states error and parameter error show good correspondence with the actual states and the true parameters. The greater the signal to noise ratio the closer the identification with regard to number of iterations. The convergence rate for the noisy input/output observations compares favorably with the algorithms of Saridis and Stein (6) and Holmes (8). It should be noted that in the noise free case only one iteration is necessary in order to identify the unknown parameters.

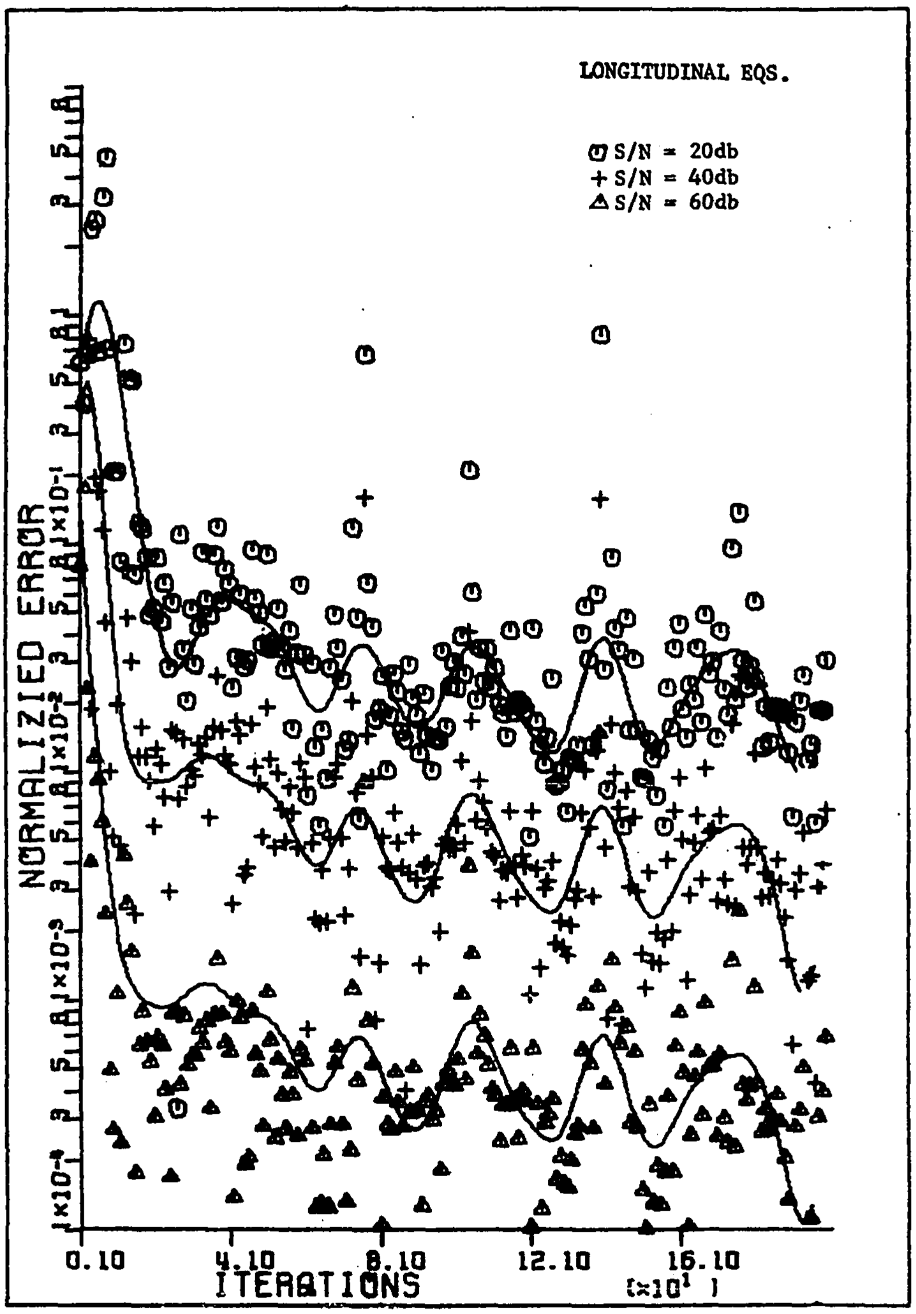


Figure 4.2. Convergence rate for state error

Table 4.3. Absolute parameter error S/N = 20 db
(Estimated parameter)

Parameter/Iteration		1	50
f_{21} =	0.1738291E 02	0.4873559E 02	0.1783069E 02
f_{22} =	0.9741994E 00	0.9424786E 00	0.9798733E 00
f_{23} =	0.5229372E-01	0.1035570E 01	0.7490249E-01
f_{24} =	-0.4134906E 02	-0.4276985E 03	-0.4233175E 02
f_{31} =	-0.2553931E 03	-0.1710916E 03	-0.2662779E 03
f_{32} =	-0.1074018E 00	-0.3063951E 00	-0.1123203E 00
f_{33} =	0.7622623E 00	-0.3661576E 00	0.7694729E 00
f_{34} =	0.2501421E 03	0.4935574E 03	0.263837E 03
f_{41} =	-0.2564270E-01	-0.9594353E 00	-0.5888821E-01
f_{42} =	0.3135588E-03	-0.2934971E-02	0.2891652E-03
f_{43} =	-0.2122424E-02	-0.6706337E-02	-0.2081408E-02
f_{44} =	0.1039389E 01	0.2805414E 01	0.1079613E 01
g_{11} =	-0.1133112E 01	-0.1164553E 01	-0.1160064E 01
g_{21} =	0.4614477E 02	0.1949300E 03	0.5818665E 02
g_{31} =	-0.2937646E 03	-0.2946374E 03	-0.3134640E 03
g_{41} =	-0.1148994E 01	-0.6775904E 00	-0.1059018E 01

100	150	200
0.1271905E 02	0.1058821E 02	0.1177326E 02
0.9810518E 00	0.9749529E 00	0.9758507E 00
0.7150866E-01	0.6823299E-01	0.5943013E-01
-0.3556995E 02	-0.3401416E 02	-0.3498680E 02
-0.2623206E 03	-0.2548507E 03	-0.2608132E 03
-0.1189027E 00	-0.1121118E 00	-0.1151139E 00
0.7660782E 00	0.7552372E 00	0.7621397E 00
0.2576531E 03	0.2501775E 03	0.2561238E 03
-0.9667312E-01	-0.1012379E 00	-0.1002283E 00
0.3117786E-03	0.3019081E-03	0.3029536E-03
-0.1959381E-02	-0.2070169E-02	-0.2083909E-02
0.1102145E 01	0.1114738E 01	0.1113321E 01
-0.1152311E 01	-0.1174831E 01	-0.1165389E 01
0.5405302E 02	0.4982184E 02	0.4860137E 02
-0.3026560E 03	-0.3038248E 03	-0.3014289E 03
-0.1055417E 01	-0.1128305E 01	-0.1127484E 01

Table 4.3. Absolute parameter error S/N = 20 db

(Absolute error)

Parameter/Iteration	1	50
f_{21} = 0.1738291E 02	0.3135268	0.0044778
f_{22} = 0.9741994E 00	0.0317208	0.0056739
f_{23} = 0.5229372E-01	9.8327628	0.2260877
f_{24} = -0.4134906E 02	3.8634944	0.0098269
f_{31} = -0.2553931E 03	0.0843015	0.0108848
f_{32} = -0.1074018E 00	0.1989933	0.0049185
f_{33} = 0.7622623E 00	0.3961047	0.0072106
f_{34} = 0.2501421E 03	0.2464153	0.0136966
f_{41} = -0.2564270E-01	9.3379260	0.3324551
f_{42} = 0.3135588E-03	2.6214122	0.0243936
f_{43} = -0.2122424E-02	0.4583913	0.0041016
f_{44} = 0.1039389E 01	0.1766025	0.0040224
g_{11} = -0.1133112E 01	0.0031441	0.0026952
g_{21} = 0.4614477E 02	1.4878523	0.1204178
g_{31} = -0.2937646E 03	0.0008728	0.0186994
g_{41} = -0.1148994E 01	0.0471403	0.0089976

100	150	200
0.0466386	0.0679470	0.0560965
0.0068524	0.0007535	0.0016513
0.1921494	0.1593927	0.0713641
0.0577911	0.0733490	0.0636226
0.0069275	0.0005424	0.0054201
0.0115009	0.0047100	0.0077121
0.0038159	0.0070251	0.0001226
0.0075110	0.0000354	0.0059817
0.7103042	0.7573630	0.7458560
0.0017802	0.0116507	0.0106052
0.0163043	0.0052255	0.0038515
0.0062756	0.0075349	0.0073932
0.0019199	0.0041719	0.0032277
0.0790825	0.0367707	0.0245660
0.0088914	0.0100602	0.0076643
0.0093577	0.0020689	0.0021510

Table 4.4. Absolute parameter error S/N = 40 db (Estimated parameter)

Parameter/Iteration		1	50
f_{21} =	0.1738291E 02	-0.1055713E 03	0.1688423E 02
f_{22} =	0.9741994E 00	0.7752862E 00	0.9742421E 00
f_{23} =	0.5229372E-01	0.4738272E 00	0.5491297E-01
f_{24} =	-0.4134906E 02	-0.8556041E 02	-0.4091193E 02
f_{31} =	-0.2553931E 03	-0.1970058E 03	-0.2571707E 03
f_{32} =	-0.1074018E 00	-0.6831815E-01	-0.1080211E 00
f_{33} =	0.7622623E 00	0.5267756E 00	0.7642420E 00
f_{34} =	0.2501421E 03	0.2685217E 03	0.2521441E 03
f_{41} =	-0.2564270E-01	-0.8300148E 00	-0.2927423E-01
f_{42} =	0.3135588E-03	-0.9852491E-03	0.3090531E-03
f_{43} =	-0.2122424E-02	0.2564200E-03	-0.2115498E-02
f_{44} =	0.1039389E 01	0.8698114E 00	0.1043099E 01
g_{11} =	-0.1133112E 01	-0.1039588E 01	-0.1135216E 01
g_{21} =	0.4614477E 02	0.9335898E 02	0.4758389E 02
g_{31} =	-0.2937646E 03	-0.2716615E 03	-0.2954451E 03
g_{41} =	-0.1148994E 01	-0.8216950E 00	-0.1139294E 01

100	150	200
0.1662231E 02	0.1649466E 02	0.1664903E 02
0.9745724E 00	0.9740636E 00	0.9741390E 00
0.5448320E-01	0.5397441E-01	0.5308778E-01
-0.4047732E 02	-0.4039655E 02	-0.4054026E 02
-0.2568398E 03	-0.2553732E 03	-0.2554422E 03
-0.1081146E 00	-0.1075876E 00	-0.1077103E 00
0.7631284E 00	0.7613840E 00	0.7616247E 00
0.2517447E 03	0.2502263E 03	0.2502906E 03
-0.3240891E-01	-0.3025507E-01	-0.2976996E-01
0.3124617E-03	0.3133196E-03	0.3132415E-03
-0.2105139E-02	-0.2119320E-02	-0.2121351E-02
0.1044951E 01	0.1043865E 01	0.1043382E 01
-0.1134556E 01	-0.1137140E 01	-0.1136320E 01
0.4708161E 02	0.4664582E 02	0.4648771E 02
-0.2943747E 03	-0.2946602E 03	-0.2944324E 03
-0.1139659E 01	-0.1147680E 01	-0.1147318E 01

Table 4.4. Absolute parameter error S/N = 40 db
(Absolute error)

Parameter/Iteration	1	50
f_{21} = 0.1738291E 02	0.8818839	0.0049868
f_{22} = 0.9741994E 00	0.1989132	0.0000427
f_{23} = 0.5229372E-01	4.2153348	0.0261925
f_{24} = -0.4134906E 02	0.4421135	0.0043713
f_{31} = -0.2553931E 03	0.0583873	0.001776
f_{32} = -0.1074018E 00	0.0390837	0.0006193
f_{33} = 0.7622623E 00	0.2384867	0.0019797
f_{34} = 0.2501421E 03	0.0183796	0.0020020
f_{41} = -0.2564270E-01	8.0437210	0.0363153
f_{42} = 0.3135588E-03	0.6716903	0.0045057
f_{43} = -0.2122424E-02	0.1866004	0.0006926
f_{44} = 0.1039389E 01	0.0169578	0.0003710
g_{11} = -0.1133112E 01	0.0093524	0.0002104
g_{21} = 0.4614477E 02	0.4721421	0.0143912
g_{31} = -0.2937646E 03	0.0221031	0.0016805
g_{41} = -0.1148994E 01	0.0327304	0.0009700

100	150	200
0.0076060	0.0093629	0.0073388
0.0003730	0.0001358	0.0000604
0.0218948	0.0168069	0.0079406
0.0087174	0.0095251	0.0084646
0.0014467	0.0000199	0.0000491
0.0007128	0.0001858	0.0003085
0.0008661	0.0008783	0.0006376
0.0016026	0.0000842	0.0001485
0.0676571	0.0461237	0.0412726
0.0010971	0.0002392	0.0003173
0.0017285	0.0003104	0.0001073
0.0005562	0.0004476	0.0003993
0.0001444	0.0004028	0.0003208
0.0093684	0.0050105	0.0034294
0.0006101	0.0008956	0.0006678
0.0009335	0.0001314	0.0001676

Table 4.5. Absolute parameter error S/N = 60 db

(Estimated parameter)

Parameter/Iteration	1	50
f_{21} = 0.1738291E 02	0.4731068E 02	0.1732810E 02
f_{22} = 0.9741994E 00	0.1012240E 01	0.9741938E 00
f_{23} = 0.5229372E-01	-0.4628977E-01	0.5255852E-01
f_{24} = -0.4134906E 02	-0.3494458E 02	-0.4130054E 02
f_{31} = -0.2553931E 03	-0.1973942E 03	-0.2555774E 03
f_{32} = -0.1074018E 00	-0.3449265E-01	-0.1074597E 00
f_{33} = 0.7622623E 00	0.5629380E 00	0.7624722E 00
f_{34} = 0.2501421E 02	0.2651073E 03	0.2503482E 03
f_{41} = -0.2564270E-01	0.1217372E 00	-0.2600948E-01
f_{42} = 0.3135588E-03	0.4974591E-03	0.3130874E-03
f_{43} = -0.2122424E-02	-0.2640043E-02	-0.2121701E-02
f_{44} = 0.1039389E 01	0.1079924E 01	0.1039757E 01
g_{11} = -0.1133112E 01	-0.1143858E 01	-0.1133160E 01
g_{21} = 0.4614477E 02	0.3903571E 02	0.4629178E 02
g_{31} = -0.2937646E 03	-0.3104274E 03	-0.2939301E 03
g_{41} = -0.1148994E 01	-0.1177808E 01	-0.1148017E 01

100	150	200
0.1730428E 02	0.1728920E 02	0.1730771E 02
0.9742291E 00	0.9741779E 00	0.9741867E 00
0.5251513E-01	0.5246514E-01	0.5237398E-01
-0.4125944E 02	-0.4124832E 02	-0.4126639E 02
-0.2555446E 03	-0.2553899E 03	-0.2553932E 03
-0.1074691E 00	-0.1074158E 00	-0.1074282E 00
0.7623534E 00	0.7621761E 00	0.7621925E 00
0.2503099E 03	0.2501493E 03	0.2501526E 03
-0.2631496E-01	-0.2607573E-01	-0.2602280E-01
0.3134396E-03	0.3135430E-03	0.3135338E-03
-0.2120686E-02	-0.2122136E-02	-0.2122344E-02
0.1039937E 01	0.1039810E 01	0.1039755E 01
-0.1133252E 01	-0.1133522E 01	-0.1133434E 01
0.4624008E 02	0.4619871E 02	0.4618012E 02
-0.2938227E 03	-0.2938539E 03	-0.2938308E 03
-0.1148062E 01	-0.1148854E 01	-0.1148832E 01

Table 4.5. Absolute parameter error S/N = 60 db (Absolute error)

Parameter/Iteration	1	50
f_{21} = 0.1738291E 02	0.2992777	0.0005481
f_{22} = 0.9741994E 00	0.0380406	0.0000056
f_{23} = 0.5229372E-01	0.0600395	0.0026480
f_{24} = -0.4134906E 02	0.0640448	0.0004852
f_{31} = -0.2553931E 03	0.0579989	0.0001843
f_{32} = -0.1074018E 00	0.0729091	0.0000579
f_{33} = 0.7622623E 00	0.1993243	0.0002099
f_{34} = 0.2501421E 03	0.0149652	0.0002061
f_{41} = -0.2564270E-01	0.9609450	0.0036678
f_{42} = 0.3135588E-03	0.1839003	0.0004714
f_{43} = -0.2122424E-02	0.0517619	0.0000723
f_{44} = 0.1039389E 01	0.0040535	0.0000368
g_{11} = -0.1133112E 01	0.0010746	0.0000048
g_{21} = 0.4614477E 02	0.0710906	0.0014701
g_{31} = -0.2937646E 03	0.0166628	0.0001655
g_{41} = -0.1148994E 01	0.0028814	0.0000977

100	150	200
0.0007863	0.0009371	0.0007520
0.0000297	0.0000215	0.0000127
0.0022141	0.0017142	0.0008026
0.0008962	0.0010074	0.0008267
0.0001515	0.0000032	0.0000010
0.0000673	0.0000140	0.0000264
0.0000911	0.0000862	0.0000699
0.0001678	0.0000072	0.0000105
0.0067226	0.0043303	0.0038010
0.0001192	0.0000158	0.0000250
0.0001738	0.0000288	0.0000080
0.0000548	0.0000421	0.0000366
0.0000140	0.0000410	0.0000322
0.0009531	0.0005394	0.0003535
0.0000581	0.0000893	0.0000662
0.0000932	0.0000140	0.0000162

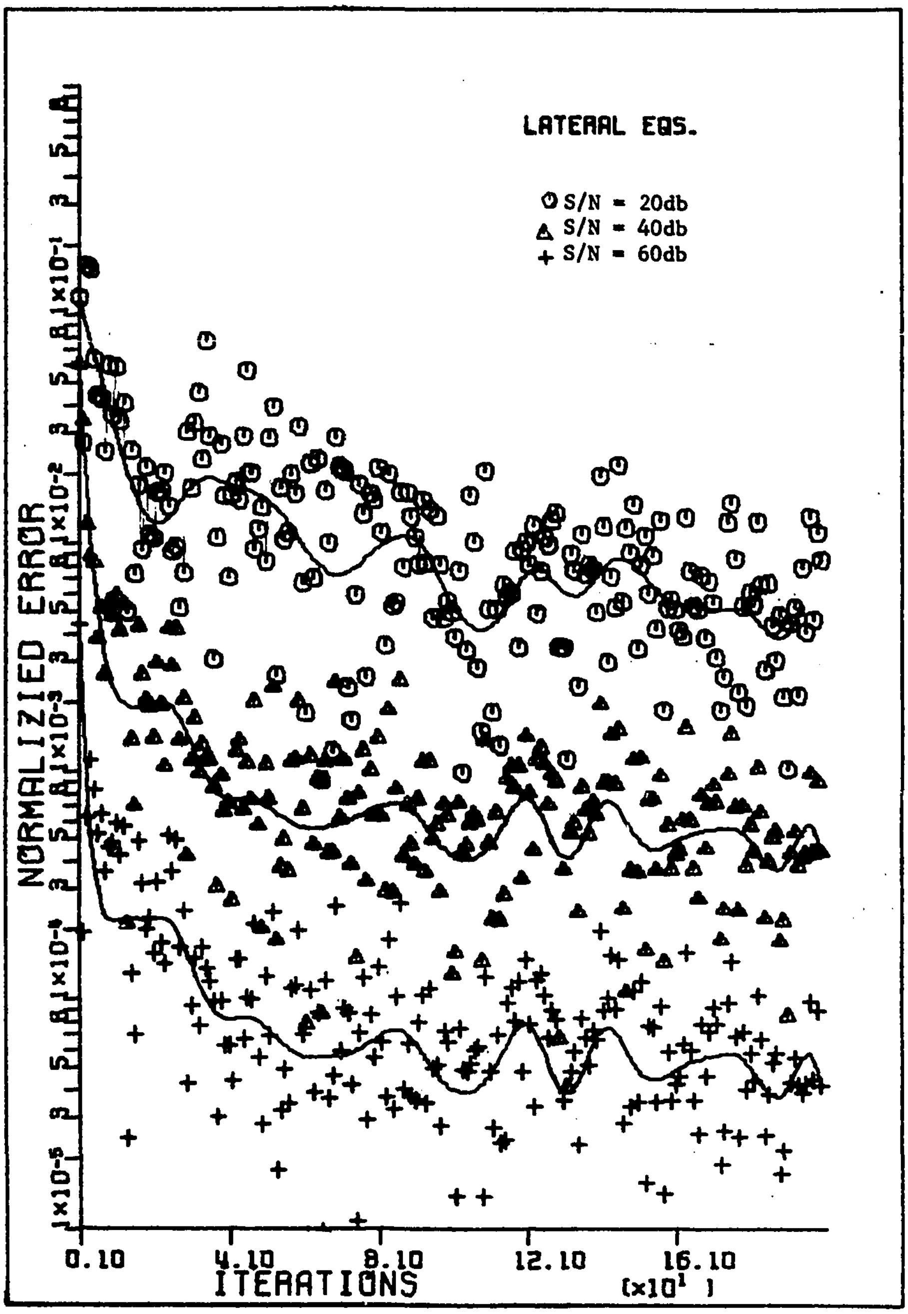


Figure 4.3. Convergence rate for state error

Table 4.6. Absolute parameter error S/N = 20 db
(Estimated parameter)

Parameter/Iteration	1	50
f_{21} = 0.1802022E 00	0.9197672E-01	0.6337377E-01
f_{22} = 0.1014800E 01	0.5695839E 00	0.6851740E 00
f_{23} = 0.1268955E-02	-0.6116086E-01	-0.3855333E-01
f_{24} = -0.2352869E 00	-0.4977766E-01	-0.5494304E-01
f_{31} = 0.1416865E 00	-0.1866323E 01	0.4182648E 00
f_{32} = -0.6111120E-01	-0.6286525E 01	0.4934371E 01
f_{33} = 0.9926340E 00	0.834515E 00	0.1596312E 01
f_{34} = 0.1836178E 00	-0.1033437E 00	-0.8359954E 00
f_{41} = -0.2767700E 00	-0.8851244E 00	-0.3701748E 00
f_{42} = 0.1624717E 01	-0.1219565E 00	0.3106321E 01
f_{43} = 0.1921763E 00	0.5058761E-01	0.3735340E 00
f_{44} = 0.1071259E 01	0.1472204E 01	0.9236054E 00
g_{11} = 0.6551825E 00	0.1836602E 00	0.6995185E 00
g_{12} = 0.1850717E 00	0.5960326E 00	0.4686740E 00
g_{21} = 0.3037358E-01	-0.7973780E-01	-0.6282792E-01
g_{22} = -0.1942935E 00	0.1154847E 01	-0.2869273E 00
g_{31} = 0.2932957E 00	0.3587494E 00	0.6119337E 00
g_{32} = 0.1817910E 00	-0.2362654E 02	0.1812687E 01
g_{41} = 0.2319299E 00	-0.2189306E 00	0.5208593E 00
g_{42} = -0.5567527E 00	-0.2302824E 01	0.4149459E-02

100	150	200
0.1128948E 00	0.1297694E 00	0.1382647E 00
0.7984117E 00	0.6594588E 00	0.6428129E 00
-0.2491588E-01	-0.4132076E-01	-0.4316155E-01
-0.1289739E 00	-0.1445538E 00	-0.1662999E 00
0.4558469E 00	0.1109714E 00	0.7904864E-01
0.2286009E 01	0.1164403E 01	0.7821305E 00
0.1275229E 01	0.1140713E 01	0.1093942E 01
-0.4263071E 00	0.6404233E-01	0.1909365E 00
-0.2712713E 00	-0.3512144E 00	-0.3456622E 00
0.2126775E 01	0.1933759E 01	0.1797702E 01
0.2534050E 00	0.2300993E 00	0.2137816E 00
0.1001318E 01	0.1114797E 01	0.1117543E 01
0.6645506E 00	0.7273659E 00	0.7420682E 00
0.2456186E 00	0.2519700E 00	0.2869519E 00
-0.1317074E-01	-0.1690940E-01	-0.2546823E-01
-0.2818560E 00	-0.2305601E 00	-0.2148296E 00
0.2612117E 00	0.5419333E 00	0.6107843E 00
0.8786002E 00	0.7355217E 00	0.6814212E 00
0.3729071E 00	0.4350285E 00	0.4575615E 00
-0.5470779E 00	-0.4903706E 00	-0.4545567E 00

Table 4.6. Absolute parameter error S/N = 20 db
(Absolute error)

Parameter/Iteration	1	50
f_{21} = 0.1802022E 00	0.0882254	0.1168284
f_{22} = 0.1014800E 01	0.0445216	0.0329626
f_{23} = 0.1268955E-02	5.9891905	3.7284375
f_{24} = -0.2352869E 00	0.1855092	0.1803438
f_{31} = 0.1416865E 00	1.7246365	0.2765783
f_{32} = -0.6111120E-01	62.2541380	48.7325980
f_{33} = 0.9926340E 00	0.1581190	0.6036780
f_{34} = 0.1836178E 00	0.0802741	0.6523776
f_{41} = -0.2767700E 00	0.6083544	0.0934048
f_{42} = 0.1624717E 01	0.1502760	0.1481604
f_{43} = 0.1921763E 00	0.1415886	0.1813577
f_{44} = 0.1071259E 01	0.040945	0.0147653
g_{11} = 0.6551825E 00	0.4715223	0.0443360
g_{12} = 0.1850717E 00	0.4109609	0.2836023
g_{21} = -0.3037358E-01	0.4936422	0.324534
g_{22} = -0.1942935E 00	0.9605535	0.4812205
g_{31} = 0.2932957E 00	0.0654537	0.3186380
g_{32} = 0.1817910E 00	23.444749	1.6308960
g_{41} = 0.2319299E 00	0.0136239	0.2889294
g_{42} = -0.5567527E 00	1.7460713	0.5526032

100	150	200
0.0673074	0.0504328	0.0419375
0.0216388	0.0355341	0.0371987
2.3646925	4.0051805	4.1892595
0.1063130	0.0907331	0.0689870
0.3141604	0.0307151	0.0626378
22.2489780	11.0329180	7.2101930
0.2825950	0.1480790	0.1013080
0.2426893	0.1195554	0.0073187
0.0054987	0.0744444	0.0688922
0.0502058	0.0309042	0.0172985
0.0612287	0.0379230	0.0216053
0.0069941	0.0043538	0.0046284
0.0093681	0.0721834	0.0868857
0.0605469	0.0668983	0.1018802
0.1720284	0.1346418	0.0490535
0.0875625	0.0362666	0.0205361
0.0320840	0.2486376	0.3174886
0.6968092	0.5537307	0.1996302
0.1409772	0.2030986	0.2256316
0.0096748	0.0663821	0.1021960

Table 4.7. Absolute parameter error S/N = 40 db (Estimated parameter)

Parameter/Iteration	1	50
f_{21} = 0.1802022E 00	-0.9352959E-01	0.1712311E 00
f_{22} = 0.1014800E 01	0.5065932E-01	0.1010618E 01
f_{23} = 0.1268955E-02	-0.1066611E 00	0.7583325E-03
f_{24} = -0.2352869E 00	0.9207415E-01	-0.2241354E 00
f_{31} = 0.1416865E 00	-0.1866290E 01	0.1517157E 00
f_{32} = -0.6111120E-01	-0.7050973E 01	-0.1564184E 00
f_{33} = 0.9926340E 00	0.2353890E 00	0.9816096E 00
f_{34} = 0.1836178E 00	0.2450219E 01	0.1719362E 00
f_{41} = -0.2767700E 00	-0.8279577E 00	-0.2830616E 00
f_{42} = 0.1624717E 01	-0.2784200E 00	0.1644505E 01
f_{43} = 0.1921763E 00	-0.1867689E-01	0.1948524E 00
f_{44} = 0.1071259E 01	0.1712254E 01	0.1069848E 01
g_{11} = 0.6551825E 00	0.2754267E 00	0.6538140E 00
g_{12} = 0.1850717E 00	0.6496938E 00	0.2032160E 00
g_{21} = -0.3037358E-01	-0.1266129E 00	-0.3090370E-01
g_{22} = -0.1942935E 00	-0.2992409E 00	-0.1991860E 00
g_{31} = 0.2932957E 00	-0.1136373E 01	0.3084099E 00
g_{32} = 0.1817910E 00	-0.1195643E 01	0.2934887E 00
g_{41} = 0.2319299E 00	-0.7085839E 00	0.2509113E 00
g_{42} = -0.5567527E 00	-0.5079149E-01	-0.5228165E 00

100	150	200
0.1768231E 00	0.1767030E 00	0.1770769E 00
0.1015528E 01	0.1005660E 01	0.1010445E 01
0.1334709E-02	0.1735225E-03	0.7534535E-03
-0.2307895E 00	-0.2302438E 00	-0.2316678E 00
0.1688386E 00	0.1474903E 00	0.1328972E 00
-0.6290429E-01	-0.5250417E-01	-0.6057030E-01
0.9925670E 00	0.9938119E 00	0.9891710E 00
0.1491558E 00	0.1696958E 00	0.1932473E 00
-0.2760488E 00	-0.283323E 00	-0.2836822E 00
0.1620396E 01	0.1619791E 01	0.1629985E 01
0.1917800E 00	0.1916885E 00	0.1928972E 00
0.1068915E 01	0.1077265E 01	0.1076626E 01
0.6545367E 00	0.6605098E 00	0.6624719E 00
0.1902039E 00	0.1909232E 00	0.1948902E 00
-0.2882288E-01	-0.2643396E-01	-0.2610679E-01
-0.2012346E 00	-0.1961063E 00	-0.1936863E 00
0.2899998E 00	0.3092488E 00	0.3145242E 00
0.2448891E 00	0.2329836E 00	0.2229467E 00
0.2437884E 00	0.2502200E 00	0.2545182E 00
-0.5568893E 00	-0.5501817E 00	-0.5452804E 00

Table 4.7. Absolute parameter error S/N = 40 db
(Absolute error)

Parameter/Iteration	1	50
f_{21} = 0.1802022E 00	0.0866726	0.0089711
f_{22} = 0.1014800E 01	0.0964140	0.0004182
f_{23} = 0.1268955E-02	10.5392145	0.0510622
f_{24} = -0.2352869E 00	0.1432127	0.0111515
f_{31} = 0.1416865E 00	1.7246035	0.0100292
f_{32} = -0.6111120E-01	69.8986100	0.9530640
f_{33} = 0.9926340E 00	0.7572450	0.0110244
f_{34} = 0.1836178E 00	2.2666012	0.0116816
f_{41} = -0.2767700E 00	0.5511877	0.0062916
f_{42} = 0.1624717E 01	0.1346297	0.0019788
f_{43} = 0.1921763E 00	0.1734994	0.0026761
f_{44} = 0.1071259E 01	0.0640995	0.0001411
g_{11} = 0.6551825E 00	0.3797558	0.0013685
g_{12} = 0.1850717E 00	0.4646211	0.0181443
g_{21} = -0.3037358E-01	0.9623932	0.0053012
g_{22} = -0.1942935E 00	0.1049474	0.0048925
g_{31} = 0.2932957E 00	0.8430773	0.0151142
g_{32} = 0.1817910E 00	0.9377733	0.1116977
g_{41} = 0.2319299E 00	0.4766540	0.0189814
g_{42} = -0.5567527E 00	0.5059612	0.0339362

100	150	200
0.0033791	0.0034992	0.0031253
0.0000728	0.0009140	0.0004355
0.0065754	0.1095432	0.0515501
0.0044974	0.0050431	0.0036191
0.0271521	0.0058038	0.0087893
0.0179229	0.0860783	0.0054090
0.0000670	0.0011779	0.0034630
0.0344620	0.0139220	0.0096295
0.0007212	0.0065530	0.0069122
0.0004321	0.0004926	0.0005268
0.0003963	0.0004878	0.0007209
0.0002344	0.0006006	0.0005367
0.0008155	0.0053273	0.0072894
0.0051322	0.005815	0.0098185
0.0155070	0.0393962	0.0426679
0.0069411	0.0018128	0.0006072
0.0032977	0.0159531	0.0212285
0.0630981	0.0511926	0.0411557
0.0118585	0.0182901	0.0225883
0.0001366	0.0065710	0.0114723

Table 4.8. Absolute parameter error S/N = 60 db
(Estimated parameter)

Parameter/Iteration	1	50
f_{21} = 0.1802022E 00	0.1915455E 00	0.1793402E 00
f_{22} = 0.1014800E 01	0.1055080E 01	0.1014551E 01
f_{23} = 0.1268955E-02	0.5881634E-02	0.1238046E-02
f_{24} = -0.2352869E 00	-0.2493294E 00	-0.2342266E 00
f_{31} = 0.1416865E 00	0.2188832E 00	0.1422643E 00
f_{32} = -0.6111120E-01	0.2207499E 00	-0.7447994E-01
f_{33} = 0.9926340E 00	0.1027257E 01	0.9910787E 00
f_{34} = 0.1836178E 00	0.7543070E-01	0.1831969E 00
f_{41} = -0.2767700E 00	-0.2787657E 00	-0.2774199E 00
f_{42} = 0.1624717E 01	0.1622667E 01	0.1625993E 01
f_{43} = 0.1921763E 00	0.1922904E 00	0.1923605E 00
f_{44} = 0.1071259E 01	0.1071183E 01	0.1071205E 01
g_{11} = 0.6551825E 00	0.6559781E 00	0.6550538E 00
g_{12} = 0.1850717E 00	0.1803694E 00	0.1868948E 00
g_{21} = -0.3037358E-01	-0.1037306E-01	-0.3043761E-01
g_{22} = -0.1942935E 00	-0.2024156E 00	-0.1947940E 00
g_{31} = 0.2932957E 00	0.4239553E 00	0.2952656E 00
g_{32} = 0.1817910E 00	-0.1374497E-01	0.1933669E 00
g_{41} = 0.2319299E 00	0.2637431E 00	0.2338970E 00
g_{42} = -0.5567527E 00	-0.6156793E 00	-0.5533030E 00

100	150	200
0.1799141E 00	0.1798899E 00	0.1799234E 00
0.1015147E 01	0.1014219E 01	0.1014722E 01
0.1308154E-02	0.1199069E-02	0.1259977E-02
-0.2349169E 00	-0.2348485E 00	-0.2349812E 00
0.1442162E 00	0.1422630E 00	0.1407295E 00
-0.6415616E-01	-0.6148079E-01	-0.6512843E-01
0.9922864E 00	0.9926044E 00	0.9921603E 00
0.1806001E 00	0.1823330E 00	0.1847126E 00
-0.2767245E 00	-0.2774603E 00	-0.2774664E 00
0.1623640E 01	0.1623[illegible]67E 01	0.1625138E 01
0.1920601E 00	0.1920843E 00	0.1922357E 00
0.1071103E 01	0.1071941E 01	0.1071807E 01
0.6551207E 00	0.6556878E 00	0.6559048E 00
0.1855783E 00	0.1855941E 00	0.1860533E 00
-0.3023841E-01	-0.2997727E-01	-0.2993886E-01
-0.1949744E 00	-0.1944526E 00	-0.1942225E 00
0.2931283E 00	0.2948250E 00	0.2954313E 00
0.1880716E 00	0.1867296E 00	0.1858837E 00
0.2331341E 00	0.2337134E 00	0.2341834E 00
-0.5567703E 00	-0.5562173E 00	-0.5555951E 00

Table 4.8. Absolute parameter error S/N = 60 db (Absolute error)

Parameter/Iteration	1	50
f_{21} = 0.1802022E 00	0.0113433	0.0008620
f_{22} = 0.1014800E 01	0.0040280	0.0000249
f_{23} = 0.1268955E-02	0.4612679	0.0030909
f_{24} = -0.2352869E 00	0.0140425	0.0010603
f_{31} = 0.1416865E 00	0.0771967	0.0005478
f_{32} = -0.6111120E-01	1.5963870	0.1336874
f_{33} = 0.9926340E 00	0.0346230	0.0015553
f_{34} = 0.1836178E 00	0.1081871	0.0004209
f_{41} = -0.2767700E 00	0.0019957	0.0006499
f_{42} = 0.1624717E 01	0.0002380	0.0001276
f_{43} = 0.1921763E 00	0.0001141	0.0001842
f_{44} = 0.1071259E 01	0.0000076	0.0000054
g_{11} = 0.6551825E 00	0.0007956	0.0001287
g_{12} = 0.1850717E 00	0.0047023	0.0018231
g_{21} = -0.3037358E-01	0.2000052	0.0006403
g_{22} = -0.1942935E 00	0.0081221	0.0005005
g_{31} = 0.2932957E 00	0.1306596	0.0019699
g_{32} = 0.1817910E 00	0.1680460	0.0115759
g_{41} = 0.2319299E 00	0.0318132	0.0019671
g_{42} = -0.5567527E 00	0.0589266	0.0034497

100	150	200
0.0002881	0.0003123	0.0002788
0.0000347	0.0000581	0.0000078
0.0039199	0.0069886	0.0008978
0.0003700	0.0004384	0.0003057
0.0025267	0.0005765	0.0009570
0.0304496	0.0036959	0.0401723
0.0003476	0.0000296	0.0004737
0.0030177	0.0012848	0.0010948
0.0000455	0.0006903	0.0006964
0.0001077	0.0000850	0.0000421
0.0001162	0.0000920	0.0000594
0.0000156	0.0000682	0.0000548
0.0000618	0.0005053	0.0007223
0.0005066	0.0005224	0.0009816
0.0013517	0.0039631	0.0043472
0.0006809	0.0001591	0.0000710
0.0001674	0.0015293	0.0021356
0.0062806	0.0049386	0.0040927
0.0012042	0.0017835	0.0022535
0.0000176	0.0005354	0.0011576

5.0. CONCLUSIONS

Through analysis of the structure of linear systems criteria have been obtained that specify requirements for the parameter identification of multiple input/multiple output systems from input/output observations. It has been shown that the compostion of a multiple input/multiple output system can be uniquely determined if the system is a minimal realization and is cyclic.

A canonical form for multiple input/multiple output systems has been developed that allows for direct parameter identification solely from input/output data. The transformation matrix associated with the multiple input/multiple output canonical form has also been derived for a large class of systems.

In the case of noise corrupted input/output observations an algorithm is presented that yields consistent estimates of the system parameters. Application of the algorithm has been made in estimating parameters in both the longitudinal and lateral equations of motion of an aircraft under simulated flight conditions.

The advatages in using the developed algorithm are:

1) input/output data are readily available
2) no a priori estimates of the identifiable parameters are needed
3) state estimation is not required nor any parameter covariance matrix needed
4) computation can be on-line

Further work in the area of identification of linear time-varying systems may prove fruitful. Another area of further endeavor is that of parameter estimation from noisy observations where removal of the requirement of knowledge of the noise statistics in the proposed algorithm is extremely desirable.

6.0. LITERATURE CITED

1. Ho, B. L., "On Effective Construction of Realizations from Input-Output Description," Ph.D. Dissertation, Stanford University, Stanford, Calif. 1965.

2. Gopinath, B., "On the Identification of Linear Time-Invariant Systems From Input-Output Data," Bell System Technical Journal, Vol. 48, May-June 1969, pp. 1101-1113.

3. Budin, M. A., "Minimal Realizations of Discrete Linear Systems from Input-Output Observations," IEEE Transactions on Automatic Control, Vol. AC-16, No. 5, Oct. 1971, pp. 395-401.

4. Nieman, R. E., Fisher, D. G., and Seborg, D. E., "A Review of Process Identification and Parameter Estimation Techniques," International Journal of Control, Vol. 13, No. 2, 1971, pp. 209-264.

5. Steiglitz, K., and McBride, L. E., "Technique for the Identification of Linear Systems," IEEE Transactions on Automatic Control, Vol. AC-10, No. 4, October 1965, pp. 461-464.

6. Saridis, G. N., and Stein, G., "Stochastic Approximation Algorithms for Linear Discrete-Time System Identification," IEEE Transactions on Automatic Control, Vol. AC-13, No. 5, October 1968, pp. 515-523.

7. Sakrison, David J., "The Use of Stochastic Approximation to Solve the System Identification Problem," IEEE Transactions on Automatic Control, Vol. AC-12, No, 5, October 1967, pp.563-567.

8. Holmes, Jack K., "Two Stochastic Approximation Procedures to Identifying Linear Systems," IEEE Transactions on Automatic Control, Vol. AC-14, No. 3, June 1969, pp. 292-295.

9. Neal, C. B., and Bekey, G. A., "Estimation of the Parameters of Sampled-Data Systems by Means of Stochastic Approximation," International Journal of System Science, 1970, Vol. 1, No. 1, pp. 63-74.

10. Wong, Kwan Y., and Polak, Elijah, "Identification of Linear Discrete Time Systems Using the Instrumental Variable Method," IEEE Transactions on Automatic Control, Vol. AC-12, No. 6, December 1967, pp. 707-718.

11. Tretter, Steve A., "Pulse-Transfer Function Identification Using Discrete Orthonormal Sequences," IEEE Transactions on Audio and Electroacoustics, Vol. AU-18, No. 2, June 1970, pp. 184-187.

12. Kalman, R. E., and Bucy, R. S., "New Results in Linear Filtering and Prediction Theory," Transactions ASME, 83D, March 1961, pp. 95-108.

13. Kopp, R. E., and Orford, R. J., "Linear Regression Applied to System Identification for Adaptive Control Systems," AIAA Journal, October 1963, pp. 2300-2306.

14. Denery, Dallas G., "An Identification Algorithm That is Insensitive to Initial Parameter Estimates," AIAA Journal, Vol. 9, No. 3, March 1971, pp. 371-377.

15. Kalman, R. E., "Mathematical Description of Linear Dynamical Systems," J.S.I.A.M. Control, Vol. 1, No. 2, 1963, pp. 152-192.

16. Kalman, R. E., "Algebraic Aspects of the Theory of Dynamical Systems," in Differential Equations and Dynamical Systems, J. K. Hale and J. P. LaSalle (eds.), Academic Press, New York, pp. 133-146.

17. Kalman, R. E., "Algebraic Structure of Linear Dynamical Systems, I. The Module of Σ," Proceedings of the National Academy of Sciences, Vol. 54, 1965, pp. 1503-1508.

18. Kalman, R. E., "Advanced Theory of Linear Systems," Part Four in Topics in Mathematical System Theory, Kalman, R. E., Falb, P. I., and Arbib, M. A., McGraw-Hill, New York, 1969.

19. Wymore, A. Wayne., "A Mathematical Theory of System Engineering - The Elements," John Wiley and Sons, Inc. New York, 1967.

20. Zeiger, Paul, "Ho's Algorithm, Commutative Diagrams, and the Uniqueness of Minimal Linear Systems," Information and Control, Vol. 11, 1967, pp. 71-79.

21. Lipschutz, Seymour, Linear Algebra, Mc-Graw Hill, New York, 1968.

22. Luenberger, D. G., "Canonical Forms for Linear Multivariable Systems," IEEE Transactions on Automatic Control, Vol. AC-12, No. 3, June 1967, pp. 290-293.

23. Vogt, W. G., and Gunderson, R. W., "Canonical Realizations," Proceedings of the First Asilomar Conference on Circuits and Systems, Vol. 1, 1967, pp. 782-790.

24. Hoffman, K., and Kange, R., Linear Algebra Second Edition, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1971.

25. Gopinath, B., "On the Identification and Control of Linear Systems", Ph.D. Dissertation, Stanford University, Stanford, California, June 1968.

26. Budin, M. A., "Parameter Identification and State Estimation for Linear Systems," Ph.D. Dissertation, Stanford University, Stanford, California, January 1970.

27. Teper, G. L., "Aircraft Stability and Control Data," NASA CR-96008, April 1969.

7.0. ACKNOWLEDGMENTS

The author would like to acknowledge the help and guidance given to him by his major professor and co-chairman, Dr. R. G. Brown and Dr. C. J. Herget, respectively. The comments and encouragement of Professor A. N. Michel are especially appreciated. Special thanks goes to Dr. L. K. Seversike for his suggestions in the field of aeronautics.

The author expresses his gratitude to the Honeywell Corporation for providing him with a fellowship and financial support while preparing this document. Financial support for the computer computation was provided by the Office of Naval Research through the funds allocated to Project Themis under contract N00014-68-A-0162.

Finally, my wife, Margaret, whose infinite patience and encouragement surely is without peer.

8.0. APPENDIX

This computer program has been written in Fortran IV with all needed subroutines contained within the program. The program directly implements Equation 3.38 and outputs the estimated matrices F and G. In addition the program can be used to generate input/output data via the subroutine Model. If input/output data are available the subroutine Model can be modified in order to allow for direct utilization of these data.

In order to use the program appropriate matrix arrays must be specified. Dimension statements must be placed in the Main, Block Data and Model routines, all other subroutines utilize object-time dimensions.

The experienced programmer will have little difficulty in determining the appropriate matrix dimensions given the program listing for the simulation of the lateral equations of motion given in Section 4 of this dissertation once he has reviewed the program.

```
      IMPLICIT INTEGER*4(I-R),REAL*8(A-H,S-Z)
      DIMENSION U(2),W(4),X(4),Y(3),Z(8),NSIG(2),IN(2)
     1,F(4,4),G(4,2),H(3,4),A(4,4),B(4,2),C(4,2),D(4,4)
     2,ASTAR(6,8),BSTAR(10,8),BSTAR1(6,8),BSTAR2(4,8),FR(4,8)
     3,YTEMP(48),UTEMP(80),USTORE(4),YSTORE(6),S(4,6),SS(8,10),SST(10,8)
     4,BSTART(8,10),BBSTAR(10,10),ABSTAR(6,10),ASTORE(6,10),BSTORE(10,10
     5),CSTORE(4,10),DSTORE(4,8),ESTORE(8,10),FSTORE(8,8)
      DIMENSION GSTORE(4,8),HSTORE(6,8),FI(4,4),GI(4,2)
      DIMENSION COV1(8,8),COV2(4,8)
      REAL*8 MU,NU,XI(4),ERR(4)
      REAL*4 SIGMA1(2),SIGMA2(3)
      EQUIVALENCE (UTEMP(1),BSTAR(1),SST(1)),(YTEMP(1),HSTORE(1))
     1,(BSTART(1),SS(1)),(ASTORE(1),CSTORE(1),B(1))
     2,(BSTORE(1),ESTORE(1),C(1)),(FSTORE(1),GSTORE(1),D(1)),(S(1),A(1))
      COMMON /LC1/U,W,X,Y,Z,NSIG,IN/LC2/TT,K
      NAMELIST/DIMEN/N,P,R,RANKH,T/PARAM/F,G,H,NSIG,IN
      NAMELIST/STAT/SIGMA1,SIGMA2,COV1,COV2,NCOUNT
      DATA P1,P2/2*1/,XI/4*0.0D0/
      READ(5,DIMEN)
      READ(5,PARAM)
      READ(5,STAT)
      WRITE(6,100)
100   FORMAT('1  ..... THE STATE TRANSITION MATRIX IS .....')
      DO 101 I=1,N
101   WRITE(6,102) (F(I,J),J=1,N)
102   FORMAT(' ',8(D14.7,2X))
      WRITE(6,103)
103   FORMAT('0  ..... THE INPUT MATRIX IS .....')
      DO 104 I=1,N
104   WRITE(6,102) (G(I,J),J=1,R)
      WRITE(6,105)
105   FORMAT('0  ..... THE OUTPUT MATRIX IS  ..... ')
      DO 106 I=1,P
106   WRITE(6,102) (H(I,J),J=1,N)
C     ..... CALCULATE MATRIX DIMENSIONS .....
      PADDR = P + R
      NSTAR = N - RANKH + 1
```

```
      PNSTAR = P*NSTAR
      RNSTAR = R*NSTAR
      PRSTAR = PNSTAR + RNSTAR
      NRSTAR = N + RNSTAR
      L= N + NSTAR*(R+1)
      LM1 = L - 1
      LP=L*P
      LR=L*R
      CALL MODEL(F,G,H,N,P,R)
C     .....  FORM VECTOR OF INPUT & OUTPUT OBSERVATIONS .....
      DO 111 K=1,L
      TT=(K-1)*T
      CALL MODEL(F,G,H,N,P,R)
      DO 109 I=1,P
      YTEMP(P1)=Y(I) + MU(SIGMA2,I)
109   P1=P1 + 1
      DO 110 I=1,R
      UTEMP(P2)=U(I) + NU(SIGMA1,I)
110   P2 = P2 + 1
111   CONTINUE
      CALL MATFOR(YTEMP,ASTAR,PNSTAR,NRSTAR,P,P,LP)
      CALL MATFOR(YTEMP,BSTAR1,PNSTAR,NRSTAR,P,0,LP)
      CALL MATFOR(UTEMP,BSTAR2,RNSTAR,NRSTAR,R,0,LR)
      DO 112 I=1,PNSTAR
      J=LP-PNSTAR+I
112   YSTORE(I)=YTEMP(J)
      DO 113 I=1,RNSTAR
      J=LR-RNSTAR+I
113   USTORE(I)=UTEMP(J)
      CALL BFORM(BSTAR1,BSTAR2,BSTAR,PNSTAR,NRSTAR,RNSTAR,PRSTAR)
      DO 200 ICRM=1,NCOUNT
      WRITE(6,114)
114   FORMAT('0 ..... THE MATRIX OF OUTPUT OBSERVATIONS IS .....')
      DO 115 I=1,PNSTAR
115   WRITE(6,102) (ASTAR(I,J),J=1,NRSTAR)
      WRITE(6,116)
116   FORMAT('0 ..... THE MATRIX OF INPUT/OUTPUT OBSERVATIONS IS ....')
```

```
      DO 117 I
117   WRITE(6,          R(I,J),J=1,NRSTAR)
C     ..... IL           N ALGORITHM .....
      CALL TRN           TART,PRSTAR,NRSTAR)
      CALL MAT           TART,BBSTAR,PRSTAR,NRSTAR,PRSTAR)
      CALL MAT           TART,ABSTAR,PNSTAR,NRSTAR,PRSTAR)
      SCA = 1.           RM))
      IF(ICRM.           1181
      CALL MATA          ,ABSTAR,ABSTAR,PNSTAR,PRSTAR)
      CALL SCAMA        ABSTAR,ABSTAR,PNSTAR,PRSTAR)
      CALL MATADD(BSTORE,BBSTAR,BBSTAR,PRSTAR,PRSTAR)
      CALL SCAMAT(SCA,BBSTAR,BBSTAR,PRSTAR,PRSTAR)
1181  CALL SMATRX(S,SS,N,PNSTAR,NRSTAR,PRSTAR)
      CALL TRNSPZ(SS,SST,NRSTAR,PRSTAR)
      CALL MATMLT(S,ABSTAR,CSTORE,N,PNSTAR,PRSTAR)
      CALL MATMLT(CSTORE,SST,DSTORE,N,PRSTAR,NRSTAR)
      CALL MATSUB(DSTORE,COV2,DSTORE,N,NRSTAR)
      CALL MATMLT(SS,BBSTAR,ESTORE,NRSTAR,PRSTAR,PRSTAR)
      CALL MATMLT(ESTORE,SST,FSTORE,NRSTAR,PRSTAR,NRSTAR)
      CALL MATSUB(FSTORE,COV1,FSTORE,NRSTAR,NRSTAR)
      CALL MATINV(FSTORE,NRSTAR,NRSTAR,1.0D-20,DETER)
      IF(DETER.EQ.0.0) GO TO 200
      CALL MATMLT(DSTORE,FSTORE,FR,N,NRSTAR,NRSTAR)
      WRITE(6,118)
118   FORMAT('0 ..... THE IDENTIFIED MATRIX (F,R) IS .....')
      DO 119 I=1,N
119   WRITE(6,102) (FR(I,J),J=1,NRSTAR)
C     ..... DETERMINATION OF F,G .....
      WRITE(6,120)
120   FORMAT('0 ..... THE IDENTIFIED STATE TRANSITION MATRIX IS .....')
      DO 121 I=1,N
121   WRITE(6,102) (FR(I,J),J=1,N)
      CALL MATFG(A,B,C,D,FI,GI,FR,N,R,NRSTAR,RNSTAR)
      WRITE(6,122)
122   FORMAT('0 ..... THE IDENTIFIED INPUT MATRIX IS .....')
      DO 123 I=1,N
123   WRITE(6,102) (GI(I,J),J=1,R)
```

```
      IF(ICRM.EQ.NCOUNT) STOP
      SCA = 1.0/SCA
      CALL SCAMAT(SCA,ABSTAR,ABSTAR,PNSTAR,PRSTAR)
      CALL SCAMAT(SCA,BBSTAR,BBSTAR,PRSTAR,PRSTAR)
      CALL MATEQ(ABSTAR,ASTORE,PNSTAR,PRSTAR)
      CALL MATEQ(BBSTAR,BSTORE,PRSTAR,PRSTAR)
C     ..... UPDATE BSTAR2 .....
      CALL SORTKY(Z,NRSTAR,1)
      CALL CSORT(BSTAR2,Z,GSTORE,RNSTAR,NRSTAR)
      CALL MATEQ(GSTORE,BSTAR2,RNSTAR,NRSTAR)
      CALL EXCOL(BSTAR2,USTORE,NRSTAR,RNSTAR,NRSTAR)
C     ..... UPDATE BSTAR1 .....
      CALL MATEQ(ASTAR,BSTAR1,PNSTAR,NRSTAR)
C     ..... UPDATE BSTAR .....
      CALL BFORM(BSTAR1,BSTAR2,BSTAR,PNSTAR,NRSTAR,RNSTAR,PRSTAR)
C     ..... UPDATE YSTORE & USTORE .....
      K=K + 1
      TT=(K-1)*T
      CALL MATEQ(X,XI,N,1)
      CALL MODEL(F,G,H,N,P,R)
      CALL MATVEC(FI,XI,Z,N,N)
      CALL MATVEC(GI,U,W,N,R)
      CALL VECADD(Z,W,XI,N,N)
      CALL MATSUB(X,XI,ERR,N,1)
      CALL VECLEN(ERR,ERROR,N)
      CALL VECLEN(X,XLEN,N)
      ERROR=ERROR/XLEN
      XICRM=DFLOAT(ICRM)
      WRITE(6,300) ICRM,TT,ERROR
300   FORMAT('0 AT ITERATION#',I3,' AND REAL TIME =',F10.5,' THE NORMALI
     1ZIED ERROR IS',D15.7)
      WRITE(7,102) ERROR,XICRM,TT
      CALL SORTKY(Z,PNSTAR,P)
      CALL RSORT(YSTORE,Z,HSTORE,PNSTAR,1)
      CALL VECEQ(HSTORE,YSTORE,PNSTAR)
      CALL SORTKY(Z,RNSTAR,R)
      CALL RSORT(USTORE,Z,GSTORE,RNSTAR,1)
```

```
      CALL VECEQ(GSTORE,USTORE,RNSTAR)
      DO 124 I=1,P
      J=PNSTAR-P+I
124   YSTORE(J)=Y(I) + MU(SIGMA2,I)
      DO 125 I=1,R
      J=RNSTAR-R+I
125   USTORE(J)=U(I) + NU(SIGMA1,I)
C     ..... UPDATE ASTAR .....
      CALL SORTKY(Z,NRSTAR,1)
      CALL CSORT(ASTAR,Z,HSTORE,PNSTAR,NRSTAR)
      CALL MATEQ(HSTORE,ASTAR,PNSTAR,NRSTAR)
      CALL EXCOL(ASTAR,YSTORE,NRSTAR,PNSTAR,NRSTAR)
200   CONTINUE
129   STOP
      END
      BLOCK DATA
      IMPLICIT INTEGER*4(I-R),REAL*8(A-H,S-Z)
      DIMENSION U(2),W(4),X(4),Y(3),Z(8),NSIG(2),IN(2)
      COMMON /LC1/U,W,X,Y,Z,NSIG,IN
      DATA U,W/6*0.0D 00/,X/-0.906007E-01,0.1201847,-2.100486,-0.3127066
     1/,Y,Z/11*0.0D 00/,NSIG,IN/4*0/
      END
      SUBROUTINE MODEL (F,G,H,N,P,R)
C     ..... MODEL GENERATES X(K+1),Y(K) AND U(K) WHERE,.....
C     .....       X(K+1) = FX(K) + GU(K)      .....
C     .....         Y(K) = HX(K)             .......
C
      IMPLICIT INTEGER*4(I-R),REAL*8(A-H,S-Z)
      DIMENSION U(2),W(4),X(4),Y(3),Z(8),NSIG(2),IN(2)
     1,F(N,N),G(N,R),H(P,N)
      REAL*8 INPUT
      COMMON /LC1/U,W,X,Y,Z,NSIG,IN/LC2/TT,K
C     ..... GENERATE U(K) .....
      DO 10 J=1,R
      NSIGJ=NSIG(J)
      INJ = IN(J)
10    U(J) = INPUT(TT,K,NSIGJ,INJ)
```

```
C     ..... CALCULATE Y(K) .....
      CALL MATVEC(H,X,Y,P,N)
C     ..... CALCULATE X(K+1)  .....
      CALL MATVEC(F,X,Z,N,N)
      CALL MATVEC(G,U,W,N,R)
      CALL VECADD(Z,W,X,N,N)
      RETURN
      END
      REAL FUNCTION INPUT*8(TT,K,NSIGJ,INJ)
C     .....  INPUT GENERATES THE FOLLOWING SYSTEM INPUTS; .....
C     .....  UNIT STEP, UNIT RAMP, UNIT ACCELERATION, WHITE GAUSSIAN
C     .....  NOISE N(0,1), STEP+NOISE, RAMP+NOISE, ACCELERATION+NOISE
      REAL*8 TT
      Y1=0.0
      GO TO (10,20,30,40),NSIGJ
10    INPUT=1.0D0 + Y1
      RETURN
20    INPUT = TT + Y1
      RETURN
30    INPUT = TT*TT + Y1
      RETURN
40    CALL NOISE(Y1,INJ,&10,&20,&30)
      INPUT = Y1
      RETURN
      END
      SUBROUTINE NOISE(Y1,INJ,*,*,*)
      CALL GAUSS(X1,X2,0.0,0.0500,Y1,Y2)
      GO TO (1,2,3,4),INJ
1     RETURN 1
2     RETURN 2
3     RETURN 3
4     RETURN
      END
      SUBROUTINE GAUSS(X1,X2,MU,SIGMA,Y1,Y2)
C     GENERATE A PAIR OF RANDOM NUMBERS NORMALLY DISTRIBUTED
C     ACCORDING TO N(MU,SIGMA)
      REAL MU
```

```
      DATA PHI/6.2834/
      CALL RANDOM(X1)
      CALL RANDOM(X2)
      Y1=SIGMA*SQRT(-2.0*ALOG(X1))*COS(PHI*X2) + MU
      Y2=SIGMA*SQRT(-2.0*ALOG(X1))*SIN(PHI*X2) + MU
      RETURN
      END
      SUBROUTINE RANDOM(Z)
C     GENERATE A UNIFORM DISTRIBUTED RANDOM NUMBER U(0,1) BY THE
C     MIXED MULTIPLICATIVE CONGRUENTIAL METNOD
      DATAI/1/
      INTEGER A,X
      IF(I.EQ.0) GO TO 1
      I = 0
      M = 2**20
      FM = M
      X=566387
      A = 2**10 + 3
1     X = MOD(A*X,M)
      FX = X
      Z = FX/FM
      RETURN
      END
      REAL FUNCTION NU*8(SIGMA1,I)
      REAL*4 SIGMA1(2)
      SIGMA=SIGMA1(I)
      CALL GAUSS(X1,X2,0.0,SIGMA,Y1,Y2)
      NU=Y1
      RETURN
      END
      REAL FUNCTION MU*8(SIGMA2,I)
      REAL*4 SIGMA2(3)
      SIGMA=SIGMA2(I)
      CALL GAUSS(X1,X2,0.0,SIGMA,Y1,Y2)
      MU=Y2
      RETURN
      END
```

```
      SUBROUTINE MATFOR(A,B,M,N,P,O,L)
      IMPLICIT INTEGER*4(I-R),REAL*8(A-H,S-Z)
      DIMENSION A(L),B(M,N)
      KK= 0
      DO 10 J=1,N
      K=(J-1)*P
      DO 10 I=1,M
      KK= K + I + O
10    B(I,J) = A(KK)
      RETURN
      END
      SUBROUTINE BFORM(A,B,C,M,N,O,P)
      IMPLICIT INTEGER*4(I-R),REAL*8(A-H,S-Z)
      DIMENSION A(M,N),B(O,N),C(P,N)
      DO 10 I=1,M
      DO 10 J=1,N
10    C(I,J)=A(I,J)
      DO 20 I=1,O
      DO 20 J=1,N
      II=M+I
20    C(II,J)=B(I,J)
      RETURN
      END
      SUBROUTINE SMATRX(S,SS,N,PNSTAR,NRSTAR,PRSTAR)
      IMPLICIT INTEGER*4(I-R),REAL*8(A-H,S-Z)
      DIMENSION S(N,PNSTAR),SS(NRSTAR,PRSTAR)
      K=0
      L=N + 1
      DO 10 I=1,N
      DO 10 J=1,PNSTAR
      S(I,J) = 0.0D0
      IF(I.EQ.J) S(I,J) = 1.0D0
10    CONTINUE
      DO 20 I=1,N
      DO 20 J=1,PRSTAR
      SS(I,J) = 0.0D0
      IF(I.EQ.J) SS(I,J) = 1.0D0
```

```
 20     CONTINUE
        DO 30 I=L,NRSTAR
        K = K + 1
        DO 30 J=1,PRSTAR
        SS(I,J) = 0.0D0
        IF(J.EQ.(PNSTAR + K)) SS(I,J) = 1.0D0
 30     CONTINUE
        RETURN
        END
        SUBROUTINE  MATFG(A,B,C,D,F,G,FR,N,R,NRSTAR,RNSTAR)
        IMPLICIT INTEGER*4(I-R),REAL*8(A-H,S-Z)
        DIMENSION A(N,N),B(N,R),C(N,R),D(N,N),F(N,N),G(N,R),FR(N,NRSTAR)
        RADD1 = R + 1
        DO 10 I=1,N
        DO 10 J=1,N
 10     F(I,J)=FR(I,J)
        CALL MATEQ(F,A,N,N)
        DO 60 M=1,RNSTAR,R
        K=N + M
        L= K - 1 + R
        Q=0
        DO 20 J=K,L
        Q = Q + 1
        DO 20 I=1,N
 20     C(I,Q)=FR(I,J)
        IF(M-RADD1) 60,30,40
 30     CALL MATMLT(F,C,B,N,N,R)
        GO TO 50
 40     CALL MATMLT(F,A,D,N,N,N)
        CALL MATEQ(D,A,N,N)
        CALL MATMLT(A,C,B,N,N,R)
 50     CALL MATADD(G,B,C,N,R)
 60     CALL MATEQ(C,G,N,R)
        RETURN
        END
        SUBROUTINE SORTKY(Z,M,P)
        IMPLICIT INTEGER*4(I-R),REAL*8(A-H,S-Z)
```

```
      DIMENSION Z(M)
      DO 10 I=1,P
10    Z(I)=DFLOAT(M+I-P)
      PP=P+1
      DO 20 I=PP,M
20    Z(I)=DFLOAT(I-P)
      RETURN
      END
      SUBROUTINE MATRX(A,B,C,T,UU,V,X,Y,Z,M,N,P)
C     SUBROUTINES FOR MANIPULATIONS OF MATRICES AND VECTORS.
      IMPLICIT INTEGER*4(I-R),REAL*8(A-H,S-Z)
      DIMENSION A(M,N),B(M,N),C(M,N),T(M,P),UU(N,P),V(N,M),X(N),Y(N)
     1,Z(M),JORD(10),JCOL(10),IROW(10),YY(10)
      REAL*8 PIVOT
      RETURN
C
C     ..... T(M,P) = A(M,N)*UU(N,P) .....
      ENTRY MATMLT(A,UU,T,M,N,P)
      DO 1 I=1,M
      DO 1 J=1,P
1     T(I,J) =0.0
      DO 2 I=1,M
      DO 2 J=1,P
      DO 2 K=1,N
2     T(I,J) = A(I,K)*UU(K,J) + T(I,J)
      RETURN
C
C  .....C(M,N) = A(M,N) + B(M,N) .....
      ENTRY MATADD (A,B,C,M,N)
      DO 3 I=1,M
      DO 3  J=1,N
3     C(I,J) = A(I,J) + B(I,J)
      RETURN
C
C  .....C(M,N) = A(M,N) - B(M,N)......
      ENTRY MATSUB (A,B,C,M,N)
      DO 4 I=1,M
```

```
      DO 4  J=1,N
4     C(I,J) = A(I,J) - B(I,J)
      RETURN
C
C  .....Z(M) = A(M,N) * X(N) .....
      ENTRY MATVEC (A,X,Z,M,N)
      DO 5 I =1,M
5     Z(I) =0.0
      DO 6 I=1,M
      DO 6 J=1,N
6     Z(I) = A(I,J)*X(J) + Z(I)
      RETURN
C
C  .....Y(N) = S * X(N) .....
      ENTRY SCAVEC (S,X,Y,N)
      DO 7 I=1,N
7     Y(I) = S*X(I)
      RETURN
C
C  ..... B(M,N) = S*A(M,N) .....
      ENTRY SCAMAT (S,A,B,M,N)
      DO 8 I=1,M
      DO 8 J=1,N
8     B(I,J) = S*A(I,J)
      RETURN
C
C  .....S = X(N) * Y(N) .....
      ENTRY VECVEC (X,Y,S,N)
      S=0.0
      DO 9 I=1,N
9     S = S + X(I)*Y(I)
      RETURN
C
C  ..... X(N) = Z(M) * A(M,N) .....
      ENTRY VECMAT(Z,A,X,M,N)
      DO 10 J =1,N
      X(J) = 0.0
```

```
      DO 10 K =1,M
10    X(J) = X(J) + Z(K)*A(K,J)
      RETURN
C
C  ..... B(M,N) = A(M,N) .....
      ENTRY MATEQ (A,B,M,N)
      DO 11 I =1,M
      DO 11 J =1,N
11    B(I,J) = A(I,J)
      RETURN
C
C  ......VECTOR LENGTH S, OF X(N) .....
      ENTRY VECLEN (X,S,N)
      SUMSQX = 0.0
      DO 12 I =1,N
12    SUMSQX = SUMSQX + X(I)*X(I)
      S = DSQRT(SUMSQX)
      RETURN
C
C  ..... Y(N) = X(N) .....
      ENTRY VECEQ (X,Y,N)
      DO 13 I =1,N
13    Y(I) = X(I)
      RETURN
C
C  .....V(N,M) = A(M,N) TRANSPOSED .....
      ENTRY TRNSPZ (A,V,M,N)
      DO 14 I =1,M
      DO 14 J =1,N
14    V(J,I) = A(I,J)
      RETURN
C     .....   MATINV   .....
C     CALCULATION OF AN INVERSE MATRIX IN PLACE.  THIS SUBROUTINE READS
C     AND COMPUTES THE INVERSE OF A MATRIX 'A' IN PLACE.  EPS IS THE
C     MINIMUM PIVOT MAGNITUDE PERMITTED BY MATINV.  SHOULD NO ACCEPTABLE
C     PIVOT BE FOUND MATINV RETURNS A TRUE ZERO AS ITS VALUE FOR DETER.
C     DETER, THE DETERMINANT OF THE MATRIX OF COEFFICIENTS IS RETURNED
```

```
C      AS THE VALUE OF MATINV AS WELL AS THE INVERSE.
       ENTRY MATINV(A,M,N,EPS,DETER)
C      ..... BEGIN ELIMINATION PROCEDURE .....
       DETER = 1.0
       DO 22 K=1,N
       KM1 = K - 1
C      ..... SEARCH FOR THE PIVOT ELEMENT .....
       PIVOT = 0.0
       DO 18 I = 1,N
       DO 18 J =1,N
C      ..... SCAN IROW AND JCOL ARRAYS FOR INVALID PIVOT .....
       IF(K.EQ.1) GO TO 17
       DO 16 ISCAN =1,KM1
       DO 16 JSCAN =1,KM1
       IF(I.EQ.IROW(ISCAN)) GO TO 18
       IF(J.EQ.JCOL(JSCAN)) GO TO 18
16     CONTINUE
17     IF(DABS(A(I,J)).LE.DABS(PIVOT)) GO TO 18
       PIVOT = A(I,J)
       IROW(K) = I
       JCOL(K) = J
18     CONTINUE
       IF(DABS(PIVOT) .GT. EPS) GO TO 19
       WRITE(6,29)
       RETURN
C      ..... UPDATE THE DETERMINANT VALUE .....
19     IROWK = IROW(K)
       JCOLK = JCOL(K)
       DETER = DETER*PIVOT
C      ..... NORMALIZE THE PIVOT ROW ELEMENTS .....
       DO 20 J =1,N
20     A(IROWK,J) = A(IROWK,J)/PIVOT
C      ..... CARRY OUT ELIMINATION & DEVELOP INVERSE .....
       A(IROWK,JCOLK) = 1.0/PIVOT
       DO 22 I=1,N
       AIJCK = A(I,JCOLK)
       IF(I.EQ.IROWK) GO TO 22
```

```
      A(I,JCOLK) = -AIJCK/PIVOT
      DO 21 J=1,N
21    IF(J.NE.JCOLK) A(I,J) = A(I,J)-AIJCK*A(IROWK,J)
22    CONTINUE
C     ..... CREATE JORDAN ARRAY .....
      DO 23 I=1,N
      IROWI = IROW(I)
      JCOLI = JCOL(I)
23    JORD(IROWI) = JCOLI
C     ..... ADJUST SIGN  OF DETERMINANT .....
      INTCH = 0
      NM1 = N - 1
      DO 24 I=1,NM1
      IP1 = I + 1
      DO 24 J=IP1,N
      IF(JORD(J).GE.JORD(I))  GO TO 24
      JTEMP = JORD(J)
      JORD(J) =JORD(I)
      JORD(I) = JTEMP
      INTCH = INTCH + 1
24    CONTINUE
      IF(INTCH/2*2.NE.INTCH) DETER=-DETER
C     ..... UNSCRAMBLE THE INVERSE .....
      DO 26 J=1,N
      DO 25 I=1,N
      IROWI=IROW(I)
      JCOLI=JCOL(I)
25    YY(JCOLI)=A(IROWI,J)
      DO 26 I=1,N
26    A(I,J) = YY(I)
C     ..... THEN BY COLUMNS .....
      DO 28 I=1,N
      DO 27 J=1,N
      IROWJ=IROW(J)
      JCOLJ=JCOL(J)
27    YY(IROWJ) = A(I,JCOLJ)
      DO 28 J=1,N
```

```
28     A(I,J) = YY(J)
29     FORMAT('0      ***** THE MATRIX IS SINGULAR ***** ')
       RETURN
C
C  ..... Z(I) = X(I) + Y(I) .....
       ENTRY VECADD(X,Y,Z,N,M)
       DO 30 I=1,N
30     Z(I) = X(I) + Y(I)
       RETURN
C
C      ..... RSORT .....
C      ..... RSORT EXCHANGES THE ROWS OF MATRIX A(M,N) FORMING MATRIX
C      B(M,N) OCCURING TO THE SORTING KEY SPECIFIED IN Z(M).  THE SORT
C      KEY PLACES ROW 'I' IN A IN THE CORRESPONDING ROW POSITION
C      SPECIFIED IN Z('I').  NUMBERS IN THE SORT KEY MUST RANGE
C      FROM 1 TO M.
       ENTRY RSORT(A,Z,B,M,N)
       DO 31 K=1,M
       I = IDINT(Z(K))
       DO 31 J = 1,N
31     B(I,J) = A(K,J)
       RETURN
C      ..... CSORT .....
C      CSORT EXCHANGES THE COLUMNS OF MATRIX A(M,N) FORMING MATRIX
C      B(M,N) ACCORDING TO THE SORTING KEY SPECIFIED IN Y(N),THE SORT
C      KEY PLACES COLUMN 'I' IN A IN THE CORRESPONDING COLUMN POSITION
C      SPECIFIED IN Y('I').   NUMBERS IN Y(N) MUST RANGE FROM 1 TO N.
       ENTRY CSORT(A,Y,B,M,N)
       DO 32 K=1,N
       J=IDINT(Y(K))
       DO 32 I=1,M
32     B(I,J)=A(I,K)
       RETURN
C      ..... EXCOL .....
C      EXCOL REPLACES COLUMN Q IN THE MATRIX A(M,N) WITH THE VALUES
C      IN Z(M).  Q ANY INTEGER FROM 1 TO N
       ENTRY EXCOL(A,Z,Q,M,N)
```

```
      DO 33 I=1,M
33    A(I,Q) = Z(I)
      RETURN
      END
```