NPS ARCHIVE
1969
WERTHER, M.

RECURSIVE ALGORITHM FOR THE BEST APPROXIMATE SOLUTION OF LENEAR EQUATIONS WITH APPLICATIONS TO SYSTEM IDENTIFICATION AND STATE ESTIMATION

by

Manfred Herman Fritz Werther

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943-5101

United States Naval Postgraduate School

THESIS

RECURSIVE ALGORITHM FOR THE BEST APPROXIMATE SOLUTION OF LINEAR EQUATIONS WITH APPLICATIONS TO SYSTEM IDENTIFICATION AND STATE ESTIMATION

by

Manfred Hermann Fritz Werther

October 1969

This document has been approved for public release and sale; its distribution is unlimited.

Library
U.S. Naval ate School
Monterey, California 93940

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943-5101

Recursive Algorithm for the Best Approximate Solution of Linear Equations with Applications to System Identification and State Estimation

by

Manfred Hermann Fritz Werther
Lieutenant Commander, Federal German Navy

Submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

from the

NAVAL POSTGRADUATE SCHOOL
October 1969

VPS ARCHIVE
1969
WERTHER, M.

ABSTRACT

The sequential solution, in recursive form, of a growing set of linear equations, based upon the least-square-error and a weighted least-square-error criterion, are developed. For comparison these results are applied to the discrete-time solution of several estimation and identification problems. Recursive algorithms for pseudo inversion and the best approximate solution of a set of linear equations are included. Finally, efficient state estimation procedures for time-invariant systems using a sliding-window observer are presented.

Library
U.S. Naval Postgraduate School
Monterey, California 93940

TABLE OF CONTENTS

ACKNOWLEDGEMENT

The author wishes to express his appreciation to Dr. Sydney R. Parker for his encouragement, guidance and assistance in this work.

I. INTRODUCTION

In a broad sense this dissertation is concerned with the basic problem of solving the linear matrix equation*

$$A\underline{x} = \underline{b} \tag{1.1}$$

where A is an m x n matrix, $\underline{x}$ an n x 1 vector of unknowns, and $\underline{b}$ an m x 1 vector of constants. The solution, if A has rank r = n, is straightforward and reduces essentially to matrix inversion

$$\hat{\underline{x}} = [A^T A]^{-1} A^T \underline{b} \tag{1.2}$$

where A^T is the transpose of A, $[A^T A]^{-1}$ is the inverse of $A^T A$, and where $\hat{\underline{x}}$ denotes the solution of (1.1). However, when the rank of A is less than n, the set of equations (1.1) is underdetermined and infinitely many solutions exist. In order to select a unique solution out of all possible solutions further constraints are imposed. In the work presented here the minimum-norm solution, as defined by Penrose [5], is selected. If $\hat{\underline{x}}$ denotes the selected solution, and $\underline{x}_o$ any other possible solution, then

$$\|\hat{\underline{x}}\| \leq \|\underline{x}_o\| \tag{1.3}$$

where $\|\underline{x}\| = \text{trace } \underline{x}\underline{x}^T$. The solution of (1.1) is further

---

*Throughout this dissertation a bar under a lower case letter represents a column matrix or vector. Capital letters generally refer to matrices.

complicated when the set of equations is inconsistent so that there is no solution which satisfies all equations. In this case a compromise solution has to be accepted such that all equations are satisfied as close as possible according to some criterion. Usually this compromise solution is selected by minimizing an error criterion J.

$$J = f(\underline{e}) \tag{1.4}$$

where $\underline{e} = A\underline{x} - \underline{b}$. The most commonly used criterion is the least-square-error sum

$$J = \underline{e}^T\underline{e} = \text{trace } \underline{e}\,\underline{e}^T = \sum_i e_i^2 \tag{1.5}$$

where e_i is the i^{th} element of the vector $\underline{e}$. When this criterion is combined with the minimum-norm condition (1.3) a unique general solution results which Penrose [5] has defined as the best approximate solution and which is obtained using the concept of the pseudo inverse.

This general approach to the solution of (1.1) is particularly useful when applied to discrete-time system problems such as state estimation, parameter identification, and the limited memory observer problem as discussed in the body of this dissertation. As an example, consider the problem of estimating the states of a linear, dynamic system from noise-contaminated measurements. The dynamic system and the measurements are given by

$$\underline{x}_k = \Phi_{k,k-1}\underline{x}_{k-1} \tag{1.6a}$$

$$z_k = M_k \underline{x}_k + \nu_k \tag{1.6b}$$

where $\underline{x}_k$ is the system state vector, $\Phi_{k,k-1}$ the discrete, time-varying transition matrix, M_k a time-varying observation matrix of dimensions $1 \times n$, ν_k the measurement noise, and z_k a scalar observation. After k observations the data may be arranged as follows

$$\begin{bmatrix} z_1 \\ z_2 \\ \cdot \\ \cdot \\ \cdot \\ z_k \end{bmatrix} = \begin{bmatrix} M_1\Phi_{1,k} \\ M_2\Phi_{2,k} \\ \cdot \\ \cdot \\ \cdot \\ M_k \end{bmatrix} \underline{x}_k \tag{1.7}$$

Thus the state estimation problem for linear, discrete-time systems is reduced to the problem of solving (1.1). When the transition matrix is the identity matrix this problem reduces to estimating a constant but unknown vector $\underline{x}$. The estimation problem has been considered for many years by such famous mathematicians as Gauss [1], Penrose [4,5], Kalman [2], Deutsch [1] and many others. In spite of their differing approaches the underlying concept remains the solution of (1.1).

The problem of identifying the coefficients of the recursive difference equation describing a time-invariant, linear system from a sequence of noisy response measurements can also be reduced to the solution of (1.1). Previous investigators have solved this identification problem

using different methods such as correlation functions, deconvolution techniques, adaptive model techniques, and others as discussed by Mishkin and Braun [17], Eveleigh [18], Eykkoff [19] and many others. The approach using the problem formulation of (1.1) has been considered by R. C. K. Lee [3].

Thus, it has been well established that it is possible to solve discrete estimation or identification problems using the concept of the best approximate solution of (1.1). For real-time computation, as required for example in some self-adaptive control systems, it is desired to obtain numerical results sequentially with a minimum of computation time and data storage. Since the dimension, m, of matrix A grows at each step in time as additional data is acquired, it is desirable to formulate a sequential solution to (1.1) in recursive form. Almost all known algorithms are based upon the assumption that matrix A has rank $r = n$ whenever $m \geq n$, which might not be true in general. The algorithm developed in this dissertation solves (1.1) sequentially for the best approximate solution [5] without any assumptions as to the rank of A. The normalized least-square-error solution and an algorithm, an alternate formulation based upon a different error criterion, are developed and applied to an estimation and an identification problem. For completeness, recursive non-sequential forms for obtaining the Penrose inverse and the best approximate solution of (1.1), when the matrix A has constant dimensions,

are included. Finally, efficient procedures for sequential state estimation of time-invariant systems are presented, where the state estimation is obtained from a finite but continuously updated set of observations (sliding-window observer). These results fall into the category of linear observer theory as discussed by Luenberger [11] and Bona [12].

The development and discussion of the foregoing results, including geometric interpretation and examples, are presented as follows. In Chapter II, Eqs. (1.1) are considered to be a set of overdetermined equations. The closed form solution, as well as the recursive solution (Kalman type filter), are well known. However, a geometric interpretation of the known results suggests an alternate way of selecting the compromise solution using a weighted least-square-error criterion. This solution is defined here as the normalized least-square-error solution. The normalized least-square-error solution in recursive form, which requires only a slight modification of the least-square-error algorithm, is applied to a specific estimation problem as well as to an identification problem. The latter consists of identifying the coefficients of a difference equation describing a dynamic, linear, time-invariant system from system response data. The experimental results are quite favorable for the normalized least-square-error solutions. For the identification problem it is shown that with sinusoidal excitation the normalized least-square-error

solution results in a smaller bias error. However, these results cannot be generalized and whether the normalized least-square-error algorithm should be used depends entirely upon the specific problem under consideration.

In Chapter III the general solution of (1.1) when the rank of A is less than or equal to n is discussed. First the definition and some properties of the pseudo inverse and the best approximate solution according to Penrose [4,5] are stated and some alternate expressions for the pseudo inverse are discussed. Then a complete recursive algorithm for the best approximate solution for the general case is developed for use as a sequential-estimation procedure. The algorithm presented here has the advantage over previously published results in that the dimensions of the matrices involved in the algorithm remain constant irrespective of the dimension, m, and rank of A. It is interesting to note that the form of the resulting filter equation

$$\underline{\hat{x}}_k = \underline{\hat{x}}_{k-1} + \underline{g}\,(z_k - \underline{a}^T\,\underline{\hat{x}}_{k-1}) \qquad (1.4)$$

evolves naturally by solving the necessary equations and is not assumed à priori. Finally, the algorithm is adapted for state estimation of linear, time-varying dynamic systems.

The solution of a fixed set of equations (1.1) is considered in Chapter IV. Although it is possible to obtain the solution with the algorithm of Chapter III, this method

is not very efficient because only the final solution, without intermediate sequential estimates, is required. More efficient methods are obtained here by combining an infinite iterative error correcting method, as developed by Noda and Nagumo [10], and the Gram-Schmidt orthogonalization process [9]. The results are finite-step algorithms for the solution of matrix equation (1.1), matrix inversion (when the matrix is nonsingular), and matrix pseudo inversion for the general case.

In Chapter V the sequential solution of a growing set of equations (1.1) is considered again. However it is assumed à priori that the set of equations is consistent and that any subsequent sets of n equations in (1.1) are independent. The problem is then solved with the ultimate goal of developing finite-memory, sliding-window observers. First, the general case of time-varying linear systems is considered and a new algorithm for the minimum-window observer (an observer with a memory limited to exactly the minimal set of data) is developed. However the high sensitivity of this observer to measurement noise severely limits its use [12]. Introducing the further constraints that the system and the observation matrix are time invariant leads to more useful and very efficient results. It is shown that with these constraints it is possible to construct simple and efficient filters for state estimation from noise-contaminated measurements using the results for the minimum-window observer. Also, using the concept of the pseudo

inverse, the memory of the observer may be extended so that a sliding-window observer of arbitrary length $\ell \geq n$ is obtained. In this case the algorithm for state estimation reduces to

$$\underline{\hat{x}}_k = F\underline{\hat{x}}_{k-1} + \underline{g}\, z_k \tag{1.5}$$

where F and g are constant. The performances of these filters when processing noisy measurements is illustrated with an example.

Finally, in Appendix A, the solution of a set of nonlinear equations using the results of Chapter IV is attempted. An iteration scheme is presented in which the value of the total difference quotient for the nonlinear part of the set of equations is iterated sequentially to its true solution. Computational results are presented for examples where this iteration process converges while other commonly used methods, such as Newton-Raphson, Gradient, and Linear Interpolation fail. This iteration scheme is then proposed for the solution of sets of nonlinear differential equation for networks containing nonlinear, memoryless, dissipative elements.

II. THE NORMALIZED LEAST-SQUARE-ERROR SOLUTION

In this chapter the solution of a set of overdetermined simultaneous linear equations is considered first, using the concepts of the pseudo inverse as developed byPenrose [4,5]. The usual least-square-error solution is presented and interpreted geometrically. It is then shown that an alternate solution, which has been designated here as the normalized least-square-error solution, is also possible and leads to a different geometrical interpretation. These results are then applied to an estimation problem, and a recursive algorithm, based upon the normalized least-square-error solution, is developed. The resulting equations are similar in form to the Kalman [2] type of discrete filter as discussed by R. C. K. Lee [3], but differ substantially in their precise formulation and the nature of the results. A numerical-estimation example comparing the least-square-error filter with the normalized least-square-error filter is presented. The recursive equations are then applied to the problem of identifying the coefficients of the difference equation describing a dynamic system from a sequence of noisy measurements of the system's response. The results of a numerical example are presented and compared with the results obtained using the usual least-square-error filter. These results indicate that the error in coefficient identification may be less for the normalized

least-square-error solution, as defined, than for the usual least-square-error solution available in the literature.

An alternate approach to the solution of the identification problem is to use estimates for the past system response rather than the past observations themselves in the problem formulation. As shown in Example 2.3 this procedure may result in a better identification of the system coefficients. Finally, it is demonstrated that in the presence of measurement noise a bias error in the identification begins to build whenever the input function is constant. However, when the input function is causing a significant dynamic system response, the estimation error approaches a constant bias. The analysis of bias error is performed by approximating the discrete formulations in the continuous time domain so that a limiting process can be performed readily. These results are verified experimentally.

A. LEAST-SQUARE-ERROR AND NORMALIZED LEAST-SQUARE-ERROR SOLUTION

The most common method of solving a set of m simultaneous equations in n unknowns, where m > n, is the least-square-error solution. The problem consists of solving the algebraic relationships

$$\underline{b} = A\underline{x} \tag{2.1}$$

where A is the m x n matrix of coefficients

$\underline{b}$ is the m x 1 vector of constants

$\underline{x}$ is the n x 1 vector of unknowns.

A solution for $\underline{x}$ is the best approximate solution introduced by Penrose [5].

$$\hat{\underline{x}} = A^{+}\underline{b} \tag{2.2}$$

where A^{+} is the Penrose pseudo-inverse, and $\hat{\underline{x}}$ is the best approximate solution. If the matrix A has rank n, this solution is obtained as*

$$\hat{\underline{x}} = [A^{T}A]^{-1}A^{T}\underline{b} \tag{2.3}$$

where A^{T} denotes the transpose of A and $[A^{T}A]^{-1}$ the inverse of $[A^{T}A]$. In this case the solution $\hat{\underline{x}}$ minimizes the cost function J

$$J = \|A\hat{\underline{x}} - \underline{b}\|^{2} = \|\underline{e}\|^{2} = \sum_{i=1}^{m} e_{i}^{2} \tag{2.4}$$

where $\underline{e} = A\hat{\underline{x}} - b$ (2.5)

and e_i represents the components of the vector $\underline{e}$. Thus this solution satisfies all the equations of (2.1) as close as possible in the least-square-error sense which is shown as follows.

---

*If A has rank less than n (i.e., the rows of A are not independent) the pseudo inverse has a different form as discussed in Chapter III.

The minimum of Eq. (1.4) occurs when

$$\frac{\partial J}{\partial \hat{\underline{x}}} = 2[A\hat{\underline{x}} - \underline{b}]^T A = 0 \tag{2.6}$$

or $$A^T A\hat{\underline{x}} = A^T\underline{b} \tag{2.7}$$

Therefore

$$\hat{\underline{x}} = [A^T A]^{-1} A^T\underline{b} \tag{2.8}$$

A geometric interpretation of the least-square-error solution may be obtained by considering the two dimensional case where the vector $\underline{x}$ has two components x_1, x_2. Each equation of (2.1) represents a line in the x_1, x_2 plane. These lines generally do not intersect at a single point. Now consider the sequence of lines given by

$$A\underline{x} = \underline{b} - \underline{e} \tag{2.9a}$$

where $\underline{e}$ is a vector with arbitrary elements, e_i. These lines are parallel to the original lines but shifted in the orthogonal direction by the amount, s_i, where

$$s_i = (a_{1i}^2 + a_{2i}^2)^{-\frac{1}{2}} e_i \ , \quad i = 1,2,\ldots m \tag{2.9b}$$

and a_{ij} are elements of the matrix A. In the least-square-error solution the lines are shifted so that they all pass through the point $\hat{x}_1$, $\hat{x}_2$ so that the cost function, J

$$J = \sum_{i=1}^{m} e_i^2 \tag{2.9c}$$

is minimized.

A different result may be obtained by considering each equation of (2.1) as a permissible geometric locus for the desired solution point. In general this locus is a hyperplane in n dimensional space, where n is the number of elements in the unknown vector $\underline{x}$. If all the loci intersect in a single point this solution corresponds to that of (2.3). If they do not intersect in a point, the solution point may be defined as the point which minimizes the sum of the distances squared to each locus. This solution is defined here as the normalized least-square-error solution. Its interpretation is quite different from the least-square-error solution and in certain types of identification problems is seen to be more meaningful than the usual results available in the literature.

The normalized least-square-error solution is a weighted least-square-error solution with weighting factors chosen such that the solution point lies as close as possible to all geometric loci described by (2.1). This result can be obtained by selecting the solution $\hat{\underline{x}}^*$ such that the scalar cost function

$$J^* = \sum_{i=1}^{m} d_i^2 \tag{2.10a}$$

is minimized, where the d_i's are the distances from $\hat{\underline{x}}^*$ to the respective loci designated by the subscript i. Before deriving this solution it is desirable to prove the following.

The distance from a point

$P^*(x_1^*, x_2^* \ldots x_n^*)$ to the plane

$$0 = a_1x_1 + a_2x_2 + \ldots + a_nx_n - c \tag{2.10b}$$

is given by

$$|d| = |(a_1^2 + a_2^2 + \ldots + a_n^2)^{-\frac{1}{2}} (a_1x_1^* + a_2x_2^* + \ldots + a_nx_n^* - c)| \tag{2.11}$$

Proof of Eq. (2.11) is given for the three dimensional case. Extension to higher dimensions is obvious.

Proof

Let the point $P^*(x^*,y^*,z^*)$ and the plane $c = a_1x+a_2y+a_3z$ be given. The unit vector normal to the plane is found by considering a plane through the origin parallel to the given plane. Its equation is

$$0 = a_1x + a_2y + a_3z \tag{2.12a}$$

which is the dot product of two vectors, one the position vector $\underline{r}$ to any point in the plane

$$\underline{r} = x\underline{i} + y\underline{j} + z\underline{k} \tag{2.12b}$$

and the other a vector normal to the plane

$$\underline{n} = a_1\underline{i} + a_2\underline{j} + a_3\underline{k} \tag{2.12c}$$

where $\underline{i}$, $\underline{j}$, and $\underline{k}$ are unit vectors defining the three-dimensional cartesian space. The unit normal vector is then

$$\underline{\eta} = \alpha_1 \underline{i} + \alpha_2 \underline{j} + \alpha_3 \underline{k} \qquad (2.12d)$$

where

$$\alpha_i = (a_1^2 + a_2^2 + a_3^2)^{-\frac{1}{2}} a_i \quad , \quad i = 1,2,3$$

The distance $|d|$ is then given by (See Fig. 2.1, where N is the point in the given plane closest to the given point P* and $\underline{NP^*}$ denotes the vector from N to P*)

$$|d| = |(\underline{r} - \underline{r}^*) \cdot \underline{\eta})| \qquad (2.12f)$$

where $\underline{r}$ is the position vector of a point in the given plane and $\underline{r}^*$ is the vector from the origin to the point P*.

Since

$$\underline{r} \cdot \underline{\eta} = \alpha_1 x + \alpha_2 y + \alpha_3 z = (a_1^2 + a_2^2 + a_3^2)^{-\frac{1}{2}} c \qquad (2.12g)$$

and

$$\underline{r}^* \cdot \underline{\eta} = (a_1^2 + a_2^2 + a_3^2)^{-\frac{1}{2}} (a_1 x^* + a_2 y^* + a_3 z^*) \qquad (2.12h)$$

it follows that

$$|d| = |(a_1^2 + a_2^2 + a_3^2)^{-\frac{1}{2}} (a_1 x^* + a_2 y^* + a_3 z^* - c)| \qquad (2.12i)$$

Extension to higher dimensions follows the same arguments as above and leads to the general result of (2.11).

The solution to the equations (2.1) which minimizes the cost function (2.9) may be obtained by considering a slight modification of (2.3). Consider the vector $\underline{d}$ with elements d_i given by

$$\underline{d} = WA\underline{x} - W\underline{b} \qquad (2.13a)$$

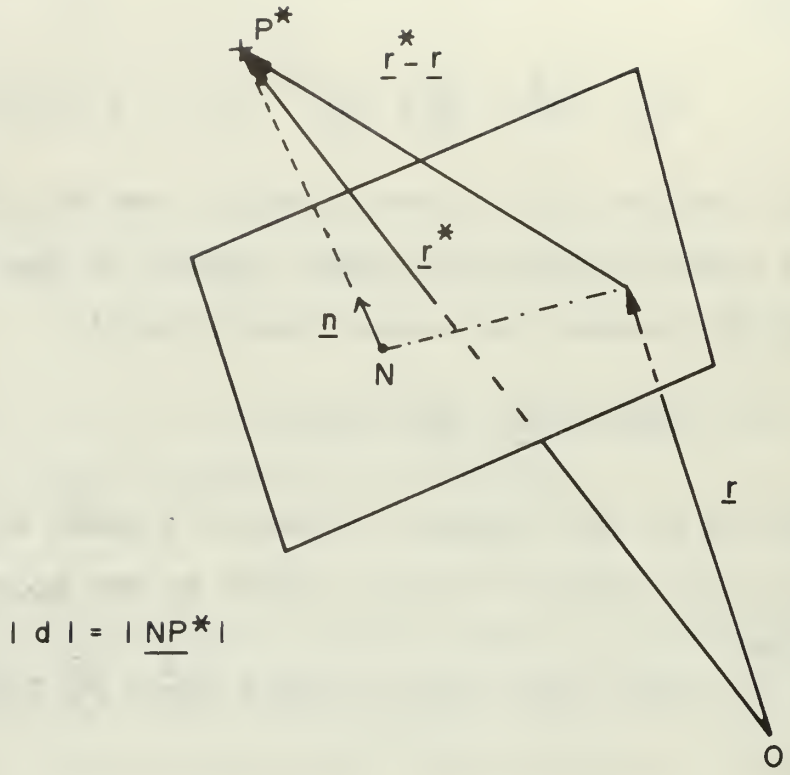

FIG. 2.1 DIAGRAM FOR PROOF OF Eq. (2.11)

where W is a diagonal mxm matrix of weighting or normalizing factors w_{ii} given by

$$w_{ii} = \left(\sum_{j=1}^{n} a_{ij} \right)^{-\frac{1}{2}} = \left(\underline{a}_i^T \underline{a}_i \right)^{-\frac{1}{2}}, \quad i=1,2,\ldots,m \qquad (2.14)$$

Eq. (2.13a) may be rewritten as

$$\underline{d} = A^*\underline{x} - \underline{b}^* \qquad (2.13b)$$

where $A^* = WA$ (2.15)

and $\underline{b}^* = W\underline{b}$ (2.16)

The solution of (2.13b) in the least-square-error sense is the desired solution, namely

$$\hat{\underline{x}}^* = [A^{*T}A^*]^{-1} A^{*T} \underline{b}^* \qquad (2.17)$$

It should be emphasized that this method solves a different set of equations (2.13a), derived from the original set of equations (2.1) by normalizing each equation, using the standard least-square-error-solution. Thus the normalized least-square-error solution is a weighted least-square-error solution of (2.1). This is quite different from other possible solutions of (2.1) minimizing alternate cost functions [18,21], i.e.

$$J_1 = \sum_{i=1}^{m} e_i^{2p}$$

where p is an integer or

$$J_2 = \sum_{i=1}^{m} |e_i|$$

The following example demonstrates that the solutions (2.3) and (2.17) may differ appreciably.

Example 2.1: Solve the set of 4 equations in 2 unknowns:

$$\text{Given} \quad \underline{b} = A\underline{x} \quad , \quad \underline{b} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \quad , \quad A = \begin{bmatrix} 1 & 1 \\ -1 & 1 \\ 10 & 10 \\ -10 & 10 \end{bmatrix}$$

According to (2.3)

$$\hat{\underline{x}} = \begin{bmatrix} .0 \\ .099 \end{bmatrix}$$

According to (2.8)

$$\hat{\underline{x}}^* = \begin{bmatrix} .0 \\ .05 \end{bmatrix}$$

These results as well as the lines defined by the above equations are shown in Fig. 2.2.

This new approach to the solution may yield more meaningful results where the indiscriminate use of the least-square-error procedure leads to unexpected or undesirable results. Consider for example the problem of estimating the parameters of the semiconductor diode model [14] from measured electrical data. If the diode is forward biased the model essentially reduces to a resistor R, representing the combined

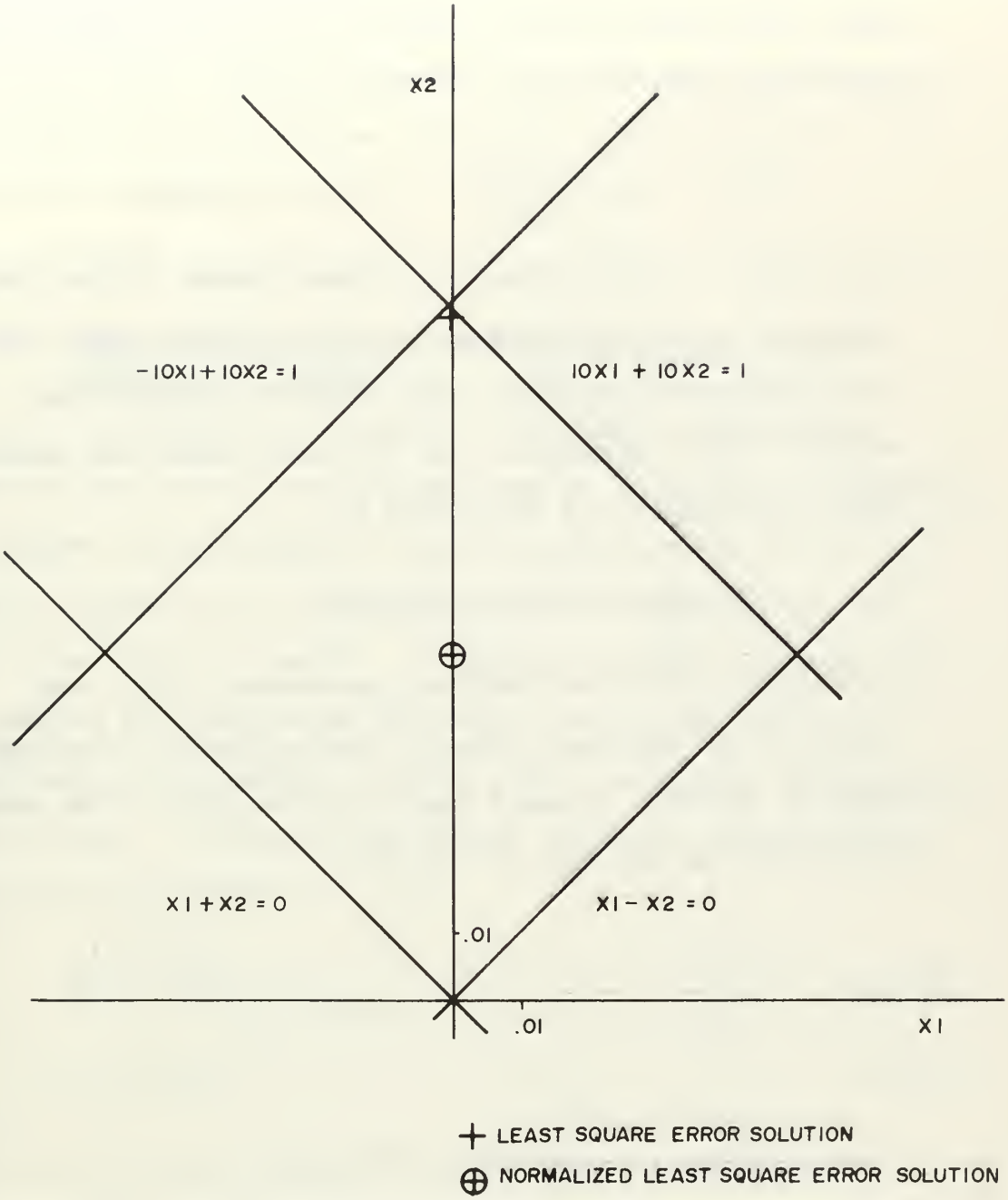
X2
-10X1+10X2=1
10X1+10X2=1
X1+X2=0
X1-X2=0
.01
.01
X1
+ LEAST SQUARE ERROR SOLUTION
⊕ NORMALIZED LEAST SQUARE ERROR SOLUTION

FIG. 2.2 EX. 2.1

body, lead, and contact resistance of the actual diode, in series with an ideal diode given by

$$V_D = k \ln\left(\frac{I-I_s}{I_s}\right) \simeq k\ln(I) - k\ln(I_s) \ .$$

V_D is the voltage across the ideal diode, I the diode current, I_s the reverse saturation current and k the characteristic constant (for constant temperature) of the diode. If V represents the voltage across the actual diode, the model is described by

$$V = IR + k\ln(I) - k\ln(I_s) \ .$$

An estimate for the constant parameters R, k, and $-k\ln(I_s)$ is obtained from ln measurements of voltage and current for the forward range of the diode. The measurements may be written in the form

$$\begin{bmatrix} V_1 \\ V_2 \\ \cdot \\ \cdot \\ \cdot \\ V_m \end{bmatrix} = \begin{bmatrix} 1 & I_1 & \ln(I_1) \\ 1 & I_2 & \ln(I_2) \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ 1 & I_m & \ln(I_m) \end{bmatrix} \begin{bmatrix} -k\ln(I_s) \\ R \\ k \end{bmatrix} \ .$$

The least-square-error solution results in a model which is very poor for small currents and excellent for very large currents, because the weights attached to the measurement vectors $[1 \ I_i \ \ln(I_i)]$ may differ by orders of magnitude. Therefore the normalized least-square-error solution, which weighs

each measurement vector equally, is preferred, resulting in a model which describes the actual diode adequately for the entire forward range.

B. RECURSIVE RELATIONSHIPS

1. Development of the Recursive Relations for an Estimation Problem

In estimation* problems the least-square-error solution is widely used in the form of a recursive relation for sequential estimation where in addition to a set of solved equations one new equation is considered and its data taken into account**. A typical example of such an estimation problem is to determine the set of initial conditions for a dynamic system from a sequence of discrete observations. Consider the system equation given in discrete form as follows:

$$\underline{y}_k = \Phi(T)\ \underline{y}_{k-1} \tag{2.18}$$

$$z_k = M\ \underline{y}_k \tag{2.19}$$

where $\underline{y}_k$ and $\underline{y}_{k-1}$ denote the state vectors at discrete times kT and (k-1)T respectively. $\Phi(T)$ is the known transition matrix of the system for the discrete time interval T. M is the known observer matrix of dimensions 1 x n and z_k is the scalar observation at time kT. If the state

---

*The term estimation is used to designate problems involving the solution of (2.1) where the elements of A are known exactly.

**See Ref. 3, page 51.

vector at k = 0 is denoted by the vector $\underline{x}$ the observations can be listed in the following form:

$$\begin{bmatrix} z_o \\ z_1 \\ \vdots \\ z_k \end{bmatrix} = \begin{bmatrix} M \\ M\Phi \\ \vdots \\ M\Phi^{k-1} \end{bmatrix} \underline{x} \tag{2.20}$$

Estimation of the vector $\underline{x}$ is equivalent to solving (2.1). Thus in general form (2.20) may be written as

$$\underline{z}_k = A_k \underline{x} \tag{2.21}$$

If A_k is of maximum rank, Lee [3] defines (M,Φ) as an observable pair and (2.21) has a least-square-error solution according to (2.3):

$$\hat{\underline{x}}_k = P_k A_k^T \underline{z}_k \tag{2.22}$$

where $\underline{z}_k$ is the vector of k observations

A_k the k x n coefficient matrix, $A_k^T = [M^T \mid \Phi^T M^T \mid \ldots \mid \Phi^{k-1 T} M^T]$

P_k the inverse of the matrix $A_k^T A_k$ and

$\hat{\underline{x}}_k$ the least-square-error solution for $\underline{x}$ based upon the k observations

Now consider the (k+1)st equation

$$z_{k+1} = \underline{a}^T \underline{x} \tag{2.33}$$

where z_{k+1} is the new observation or data in the vector $\underline{z}_{k+1}$ and $\underline{a}^T$ a row of new coefficients. $\underline{a}^T$ is equal to $M\Phi^{k+1}$ for

the estimation example above. It is now possible to define the matrix A_{k+1} and the observation vector $\underline{z}_{k+1}$.

$$A_{k+1} = \begin{bmatrix} A_k \\ - - - \\ \underline{a}^T \end{bmatrix}, \quad \underline{z}_{k+1} = \begin{bmatrix} \underline{z}_k \\ - - - \\ z_{k+1} \end{bmatrix} \tag{2.24}$$

The least-square-error solution then takes the form

$$\hat{\underline{x}}_{k+1} = P_{k+1} A_{k+1} \underline{z}_{k+1} \tag{2.25}$$

where P_{k+1} is the inverse of the matrix $[A_{k+1}^T A_{k+1}]$. This result can be written in the following recursive form [3].

$$\left\{ \begin{array}{l} \hat{\underline{x}}_{k+1} = \hat{\underline{x}}_k + \dfrac{P_k \underline{a}}{1+\underline{a}^T P_k \underline{a}} (z_{k+1} - \underline{a}^T \underline{x}_k) \\ \\ P_{k+1} = P_k - \dfrac{P_k \underline{a}\underline{a}^T P_k}{1+\underline{a}^T P_k \underline{a}} \end{array} \right\} \tag{2.26}$$

It should be noted that $(1+\underline{a}^T P_k \underline{a})$ is a scalar and is treated accordingly. The derivation of (2.26) normally available in the literature is rather involved and a short derivation which has been developed here is presented.

$$P_k^{-1} = A_k^T A_k \tag{2.27}$$

$$P_{k+1}^{-1} = A_{k+1}^T A_{k+1} = A_k^T A_k + \underline{a}\,\underline{a}^T = P_k^{-1} + \underline{a}\,\underline{a}^T$$

$$P_{k+1}^{-1} = [I + \underline{a}\,\underline{a}^T P_k] P_k^{-1} \tag{2.28a}$$

After premultiplying by P_{k+1} and postmultiplying by P_k (2.28a) becomes

$$P_k = P_{k+1} + P_{k+1}\,\underline{a}\underline{a}^T\,P_k \qquad \text{(a.28b)}$$

Combining (2.28b) with (2.28a) yields

$$P_{k+1} = P_k - P_k\,[I + \underline{a}\underline{a}^T P_k]^{-1}\,\underline{a}\underline{a}^T P_k \qquad (2.29)$$

The expression $[I+\underline{a}\,\underline{a}^T P_k]^{-1}\underline{a}$ may be simplified.

Let $\quad \underline{\xi} = [I + \underline{a}\underline{a}^T P_k]^{-1}\underline{a}$

then $\quad \underline{\xi}(1+\underline{a}^T P_k \underline{a}) = [I + \underline{a}\underline{a}^T P_k]^{-1}\underline{a}\,(1 + \underline{a}^T P_k \underline{a})$

$$= [I + \underline{a}\underline{a}^T P_k]^{-1}\,[\underline{a} + \underline{a}\underline{a}^T P_k \underline{a}]$$

$$= [I + \underline{a}\underline{a}^T P_k]^{-1}\,[I + \underline{a}\underline{a}^T P_k]\underline{a}$$

$$= \underline{a}$$

Thus $\underline{\xi} = \underline{a}/(1 + \underline{a}^T P_k \underline{a})$ and (2.9) reduces to

$$P_{k+1} = P_k - \frac{P_k \underline{a}\underline{a}^T P_k}{1+\underline{a}^T P_k \underline{a}} \qquad (2.30)$$

Then $\quad \hat{\underline{x}}_{k+1} = P_{k+1}\,A_{k+1}^T\,\underline{z}_{k+1}$

$$= \left[P_k - \frac{P_k \underline{a}\underline{a}^T P_k}{1+\underline{a}^T P_k \underline{a}}\right]\left[A_k^T \underline{z}_k + z_{k+1}\underline{a}\right]$$

$$= P_k A_k^T \underline{z}_k - \frac{P_k \underline{a}}{1+\underline{a}^T P_k \underline{a}}\,\underline{a}^T P_k A_k^T \underline{z}_k + z_{k+1}\left(P_k \underline{a} - \frac{P_k \underline{a}\underline{a}^T P_k \underline{a}}{1+\underline{a}^T P_k \underline{a}}\right)$$

$$= \hat{\underline{x}}_k - \frac{P_k \underline{a}}{1+\underline{a}^T P_k \underline{a}} \underline{a}^T \hat{\underline{x}}_k + z_{k+1} \frac{P_k \underline{a}}{1+\underline{a}^T P_k \underline{a}}$$

$$\hat{\underline{x}}_{k+1} = \hat{\underline{x}}_k + \frac{P_k \underline{a}}{1+\underline{a}^T P_k \underline{a}} (z_{k+1} - \underline{a}^T \hat{\underline{x}}_k) \tag{2.31}$$

(2.30) and (2.31) are the desired equations of the recursive form (2.26).

By comparing (2.3) and (2.17) the recursive relationship for the normalized least-square-error solution follows

$$\left\{ \begin{array}{l} \hat{\underline{x}}^*_{k+1} = \hat{\underline{x}}^*_k + \dfrac{P^*_k \underline{a}^*}{1+\underline{a}^{*T} P^*_k \underline{a}^*} (z^*_{k+1} - \underline{a}^{*T} \hat{\underline{x}}^*_k) \\ \\ P^*_{k+1} = P^*_k - \dfrac{P^*_k \underline{a}^* \underline{a}^{*T} P^*_k}{1+\underline{a}^{*T} P^*_k \underline{a}^*} \end{array} \right\} \tag{2.32a}$$

Now because of the normalization

$$\left. \begin{array}{l} z^*_{k+1} = (\underline{a}^T \underline{a})^{-\frac{1}{2}} z_{k+1} \\ \\ \underline{a}^* = (\underline{a}^T \underline{a})^{-\frac{1}{2}} \underline{a} \end{array} \right\} \tag{2.32b}$$

Substituting (2.32b) into (2.32a) yields

$$\left\{ \begin{array}{l} \hat{\underline{x}}^*_{k+1} = \hat{\underline{x}}^*_k + \dfrac{P^*_k \underline{a}}{\underline{a}^T \underline{a} + \underline{a}^T P^*_k \underline{a}} (a_{k+1} - \underline{a}^T \hat{\underline{x}}^*_k) \\ \\ P^*_{k+1} = P^*_k - \dfrac{P^*_k \underline{a} \underline{a}^T P^*_k}{\underline{a}^T \underline{a} + \underline{a}^T P^*_k \underline{a}} \end{array} \right\} \tag{2.32c}$$

From (2.32c) it follows that explicit normalization of each equation is unnecessary. Instead a slight modification of

(2.26), as given in (2.32), results in the desired algorithm. It should be noted that (2.32) is not valid for the meaningless observation with the coefficient vector $\underline{a} = \underline{0}$, which must be excluded from the recursive procedure. For this case

$$\left.\begin{array}{l} \hat{\underline{x}}^*_{k+1} = \hat{\underline{x}}^*_k \\ \\ P^*_{k+1} = P^*_k \end{array}\right\} \text{for } \underline{a} = \underline{0} \qquad (2.32d)$$

In the following example sequential estimates for a vector of constant elements are computed from noise contaminated measurements using both the least-square-error and the normalized least-square-error procedure. The measurement noise is derived from a noise population with zero mean and a distribution of finite extent (i.e. uniform distribution), rather than from a normal distribution, in order to conform with practical problems where the largest possible measurement error is bounded.

Example 2.2: Estimate the unknown vector $\underline{x}$ of dimensions 2 x 1 from the scalar measurements given as

$$z_k = M_k \underline{x} + \nu_k \qquad (2.33a)$$

where

$$M_k = [1, \ (\frac{k-1}{k})^2] \qquad (2.33b)$$

M_k is a time varying observation matrix, z_k the scalar observation and ν_k the measurement noise. At time instant k

the k equations, according to (2.33a), may be written in matrix form as

$$\underline{z}_k = A_k \underline{x} + \underline{\nu}_k \tag{2.33c}$$

where

$$A_k = \begin{bmatrix} 1 & 0 \\ 1 & 1/4 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ 1 & (\frac{k-1}{k})^2 \end{bmatrix} \tag{2.33d}$$

$$\underline{z}_k = [z_1 z_2 \ \cdots \ z_k]^T \tag{2.33e}$$

$$\underline{\nu}_k = [\nu_1 \nu_2 \ \cdots \ \nu_k]^T \tag{2.33f}$$

Since $\underline{x}$ must be estimated from noisy measurements, $\underline{z}_k$, it is necessary to solve the equation

$$\underline{z}_k = A_k \underline{x} \tag{2.33g}$$

The solution of (2.33g) for the estimate of $\underline{x}$ is given by

$$\hat{\underline{x}}_k = P_k A_k^T \underline{z}_k \tag{2.33h}$$

where $$P_k = (A_k^T A_k)^{-1} \tag{2.33i}$$

The estimation error is then defined as

$$\tilde{\underline{x}}_k = \hat{\underline{x}}_k - \underline{x}_k - P_k A_k^T \underline{z}_k - P_k A_k^T (\underline{z}_k - \underline{\nu}_k)$$

$$= P_k A_k^T \underline{\nu}_k \tag{2.33k}$$

Results for a specific case, where $\underline{x} = \begin{bmatrix} 0.1 \\ 0.9 \end{bmatrix}$ and the measurement noise ν_k is a sample from a uniform noise population with maximum deviation ±0.1 and zero mean are shown in Figure 2.3 and Figure 2.4. The sum ε of the absolute estimation errors

$$\varepsilon = |\tilde{x}_k^1| + |\tilde{x}_k^2| \tag{2.33ℓ}$$

where $\tilde{x}_k^1$ and $\tilde{x}_k^2$ are the two elements of $\underline{\tilde{x}}_k$ is shown for both estimators in Figure 2.5. Note that the estimation error does not approach zero.

The experimental result of Example 2.2 may be verified by considering a similar estimation problem where the sequence of observation matrices is given as

$$A_k = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ 1 & 1 \end{bmatrix} \tag{2.33m}$$

The matrix P_k in (2.26) for this problem takes the form

$$P_k = \begin{bmatrix} 1 & -1 \\ -1 & 1+\frac{1}{k} \end{bmatrix} \quad \text{for } k \geq 2$$

In the limit as k goes to infinity

$$\lim_{k\to\infty} P_k = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{2.33n}$$

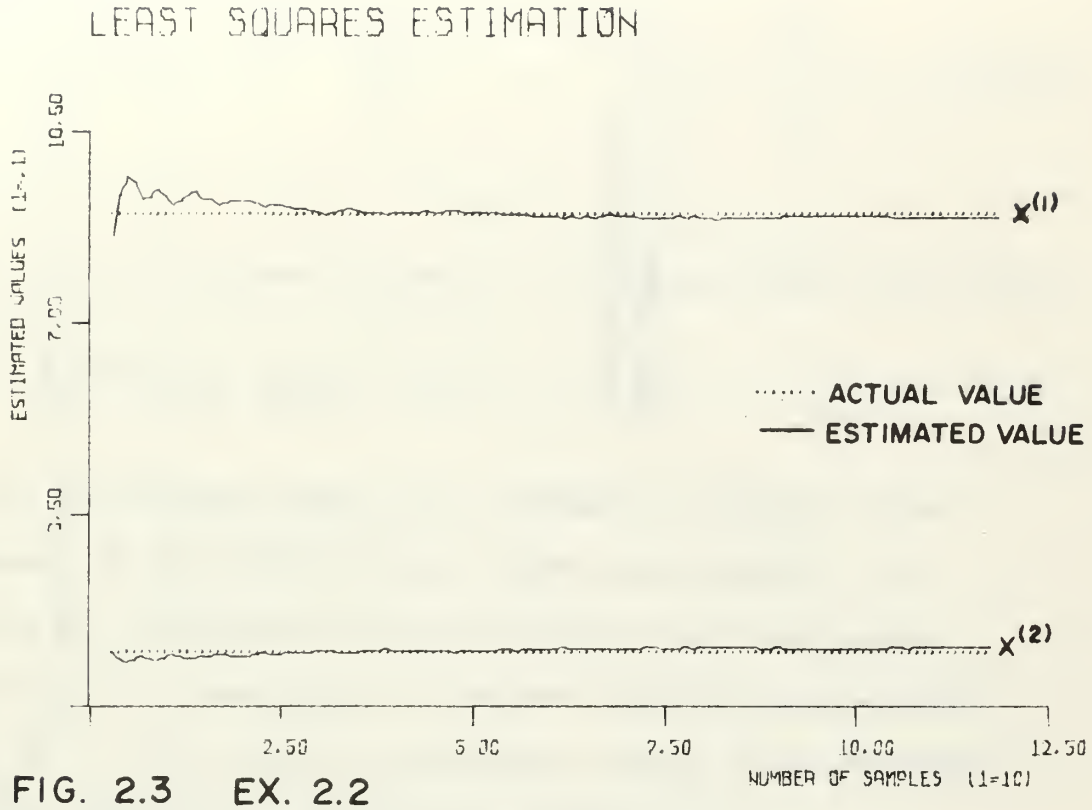

FIG. 2.3 EX. 2.2

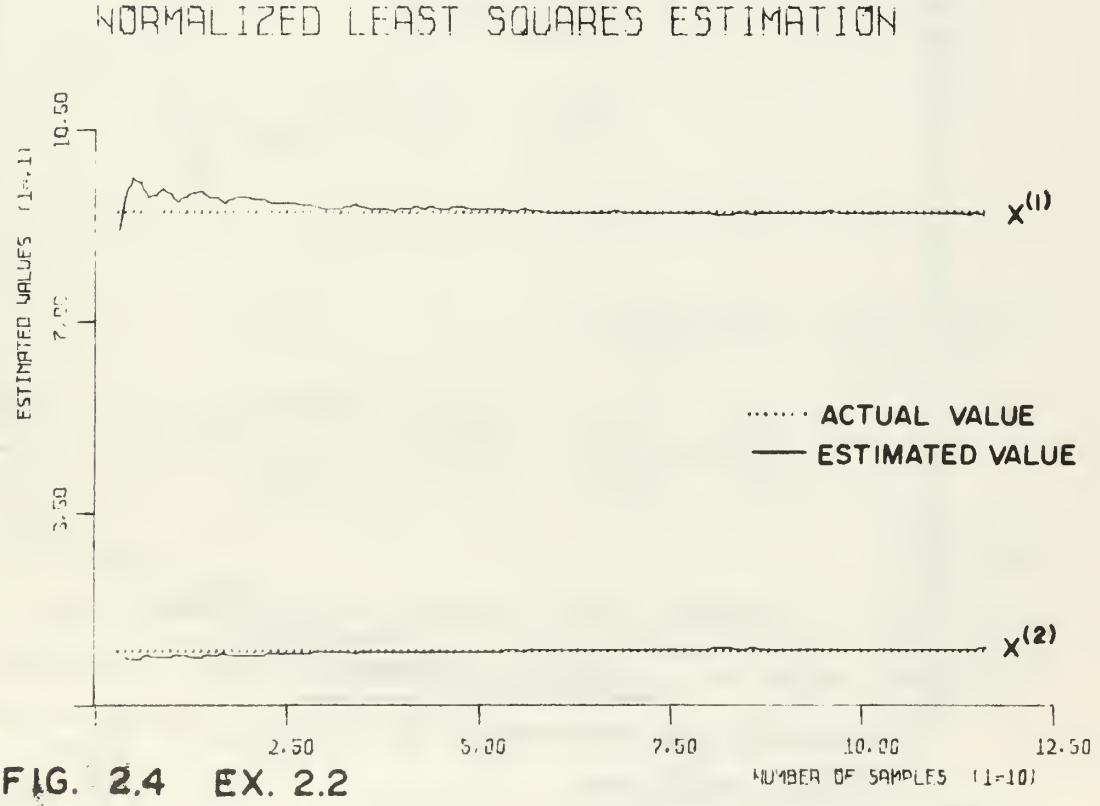

FIG. 2.4 EX. 2.2

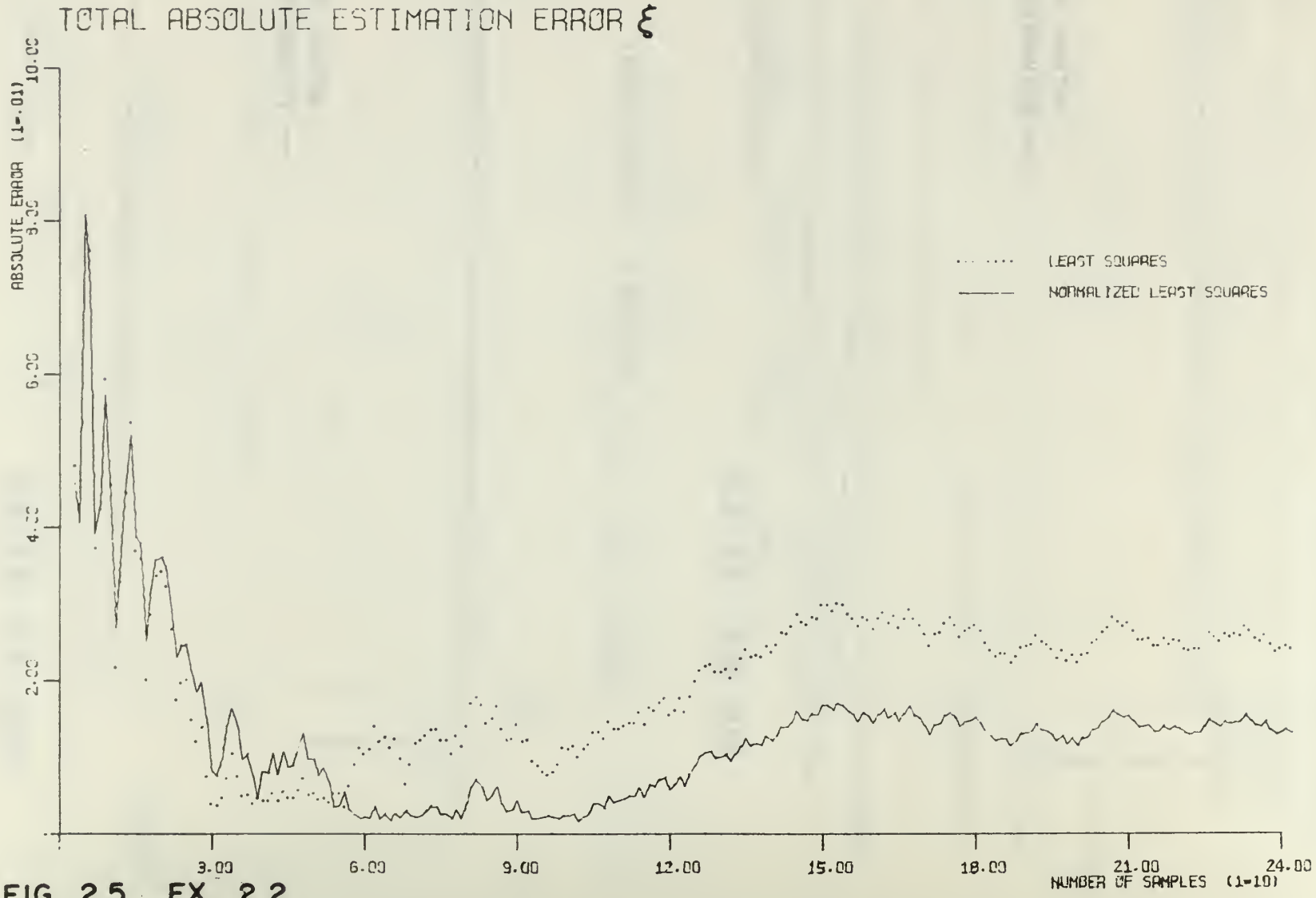
TOTAL ABSOLUTE ESTIMATION ERROR ξ
ABSOLUTE ERROR (1=.01)
10.00
6.00
2.00
LEAST SQUARES
NORMALIZED LEAST SQUARES
3.00
6.00
9.00
12.00
15.00
18.00
21.00
24.00
NUMBER OF SAMPLES (1=10)

FIG. 2.5 EX. 2.2

and $$\lim_{k\to\infty} \tilde{\underline{x}}_k = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 0 & 1 & & 1 \end{bmatrix} \begin{bmatrix} \nu_1 \\ \nu_2 \\ \vdots \\ \nu_k \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \nu_1 \qquad (2.33o)$$

Since ν_1 is a random variable, it follows from (2.33o) that

$$\lim_{k\to\infty} \{\text{Prob } (\hat{\underline{x}}-\underline{x}) = 0\} = 0$$

and the estimation error will approach a constant bias dependent on the value of the first noise sample. In Example 2.2 the result is quite similar and the estimation error will approach a constant bias with probability equal to 1. This bias is dependent mainly upon the first few noise samples as shown in the following.

$$A^T A = \begin{bmatrix} k & \sum_{2}^{k} (1 - \frac{1}{i})^2 \\ \sum_{2}^{k} (1-\frac{1}{i})^2 & \sum_{2}^{k} (1 - \frac{1}{i})^4 \end{bmatrix}$$

and $$P_k = \frac{1}{\sum_{2}^{k} (1-\frac{1}{i})^4 - \left[\frac{1}{k} \sum_{2}^{k} (1-\frac{1}{i})^2\right]^2} \begin{bmatrix} \frac{1}{k} \sum_{2}^{k} (1-\frac{1}{i})^4 & -\frac{1}{k} \sum_{2}^{k} (1-\frac{1}{i})^2 \\ -\frac{1}{k} \sum_{2}^{k} (1-\frac{1}{i})^2 & 1 \end{bmatrix} \qquad (2.33p)$$

In the limit as k goes to infinity the elements of P_k do not go to zero but approach constant values. An approximation of $\lim_{k\to\infty} P_k$ may be obtained by converting the piecewise-constant functions in (2.33p) to continuous functions and the summations to integrations. Thus

$$\lim_{k\to\infty} \frac{1}{k} \sum_{2}^{k} \left(1-\frac{1}{i}\right)^4 \simeq \lim_{k\to\infty} \frac{1}{k} \int_{2-\frac{1}{2}}^{k-\frac{1}{2}} \left(1-\frac{1}{x}\right)^4 dx$$

$$= \lim_{k\to\infty} \left\{\frac{1}{k}\left[x - 4\ell nx - \frac{6}{x} + \frac{2}{x^2} - \frac{1}{3x^3}\right]_{1.5}^{k+.5}\right\}$$

$$= 1 - 4 \lim_{k\to\infty} \left[\frac{1}{k} \ell n(k+.5)\right]$$

$$= 1$$

Since

$$\lim_{k\to\infty} \frac{1}{k} \ell n(k) = \lim_{k\to\infty} \left\{k^{-1}\left[\left(1-\frac{1}{k}\right) + \frac{1}{2}\left(1-\frac{1}{k}\right)^2 + \ldots\right]\right\}$$

$$= 0$$

Also

$$\lim_{k\to\infty} \frac{1}{k} \sum_{2}^{k} \left(1-\frac{1}{x}\right)^2 dx \simeq \lim_{k\to\infty} \frac{1}{k} \int_{2-\frac{1}{2}}^{k+\frac{1}{2}} \left(1-\frac{1}{x}\right)^2 dx$$

$$= \lim_{k\to\infty} \left\{\frac{1}{k}\left[x-2\ell nx - \frac{1}{x}\right]_{1.5}^{k+.5}\right\}$$

$$= 1 - 2 \lim_{k\to\infty} \left[\frac{1}{k} \ell n\ (k+.5)\right]$$

$$= 1$$

and

$$\lim_{k\to\infty} \left\{\sum_{2}^{k} \left(1-\frac{1}{i}\right)^4 - \frac{1}{k}\left[\sum_{z}^{k} \left(1-\frac{1}{i}\right)^2\right]^2\right\}$$

$$\simeq \lim_{k\to\infty} \left\{\int_{2-\frac{1}{2}}^{k+\frac{1}{2}} \left(1-\frac{1}{x}\right)^4 dx - \frac{1}{k}\left[\int_{2-\frac{1}{2}}^{k+\frac{1}{2}} \left(1-\frac{1}{x}\right)^2 dx\right]^2\right\}$$

$$\simeq \lim_{k\to\infty} \left\{ \left[x - 4\ell nx - \frac{6}{x} + \frac{2}{x^2} - \frac{1}{3x^3}\right]_{1.5}^{k+.5} - \frac{1}{k}\left[\left(x - 2\ell nx - \frac{1}{x}\right)\Big|_{1.5}^{k+.5}\right]^2 \right\}$$

$$\simeq -\left(1.5 - 4\ell n(1.5) - \frac{6}{1.5} + \frac{2}{1.5^2} - \frac{1}{3x1.5^3}\right.$$

$$+\lim_{k\to\infty} \left\{ (k+.5) - 4\ell n(k+.5) - \frac{(k+.5)^2}{k} + 4\,\frac{(k+.5)}{k}\ell n(k+.5) - \frac{4}{k}[\ell n(k+.5)]^2 \right\}$$

$$\simeq 3.33 + \lim_{k\to\infty} \frac{4}{k}\,[\ell nk]^2$$

$$\simeq 3.33$$

Since

$$\lim_{k\to\infty} \frac{4}{k}\,[\ell n(k)]^2 = 4 \lim_{k\to\infty} [k^{-\frac{1}{2}}\ell n(k)]^2$$

$$= 4 \lim_{k\to\infty} \{k^{-\frac{1}{2}}[(1-\tfrac{1}{k}) + \tfrac{1}{2}(1-\tfrac{1}{k})^2 \ldots]\}^2$$

$$= 0$$

Thus

$$\lim_{k\to\infty} P_k \simeq \frac{1}{3.33}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \qquad (2.33q)$$

and also

$$\lim_{k\to\infty} \underline{x}_k = \lim_{k\to\infty} P_k A_k^T \underline{\nu}_k$$

$$\simeq .3 \begin{bmatrix} 1 \\ \\ -1 \end{bmatrix} \sum_{i=1}^{\infty} \frac{1}{i}\,(2-\tfrac{1}{i})\nu_i \qquad (2.33r)$$

$$\simeq .3 \begin{bmatrix} 1 \\ \\ -1 \end{bmatrix} \{\nu_1 + \tfrac{3}{4}\nu_2 + \tfrac{5}{9}\,\nu_3 + \tfrac{7}{16}\,\nu_4 + \tfrac{9}{25}\,\nu_5 \ldots\}$$

Eq. (2.33)r is a weighted sum which places emphasis on the first few noise samples and whose weighting factors approach zero as k becomes larger and larger. Thus for large k the estimation error depends mainly upon the first terms of the summation in (2.33)r and approaches a constant bias.

As a consequence of Example 2.2 the assertion in Lee [Ref. 3, page 53] that $\lim_{k\to\infty} P_k = [0]$ is contradicted and the estimate obtained is not consistent.

2. Application to Identification Problems

The foregoing results may also be used in identification* problems. Consider a discrete-time system characterized by the discrete equation

$$x_k = a_1 x_{k-1} + a_2 x_{k-2} + \ldots + a_n x_{k-n} + b_1 u_{k-1} + b_2 u_{k-2} + \ldots + b_m u_{k-m} \tag{2.34}$$

where in general $n \geq m$, and x_{k-n} and u_{k-n} represent the system response and driving function at time $t = (k-n)T$. It is desired to determine the coefficients a_i and b_j (where $i=1,2,\ldots n$ and $j=1,2,\ldots m$) from sequential measurements of the output. If the measurements z_k are noiseless then

$$z_k = x_k = \left[x_{k-1} \cdots x_{k-n} \,\middle|\, u_{k-1} \cdots u_{k-m}\right] \left[\begin{array}{c} a_n \\ \hline b_m \end{array}\right]$$

after $k = n+m$ measurements the following data bank will result if the system starts with zero initial conditions.

---

*The expression identification is used to designate problems involving the solution of (2.1) when some or all of the elements in A are random variables.

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_{n+m} \end{bmatrix} = \left[\begin{array}{ccc|ccc} x_0 & 0 & 0 & u_0 & 0 & 0 \\ x_1 & x_0 & 0 & u_1 & u_0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n+m-1} & x_{n+m-2} & x_m & u_{n+m-1} & u_{n+m-2} \cdots & u_n \end{array}\right] \begin{bmatrix} a_1 \\ \vdots \\ a_n \\ b_1 \\ \vdots \\ b_m \end{bmatrix} \tag{2.35}$$

or

$$\underline{z}_{n+m} = [X_{n+m} \mid U_{n+m}] \begin{bmatrix} \underline{a}_n \\ \hline \underline{b}_m \end{bmatrix} \tag{2.36}$$

The identification problem is then to solve (2.35) for the vector $\begin{bmatrix} \underline{a}_n \\ \hline \underline{b}_m \end{bmatrix}$. An exact solution is obtained only if the matrix $[X_{n+m} \mid U_{n+m}]$ is nonsingular. Sequential estimation is then possible using (2.26) or (2.32) when the number of observations $K \geq m+n$ and the matrix $[X_{n+m} \mid U_{n+m}]$ has maximum rank equal to m+n.

In the presence of measurement noise the observations of the output become

$$z_{k+1} = x_{k+1} + \nu_{k+1}$$

where ν_{k+1} is the measurement noise. Eq. (2.35) may then be approximated by

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_k \end{bmatrix} = \left[\begin{array}{ccc|c} z_0 & 0 & \cdots & 0 \\ z_1 & z_0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ z_{k-1} & z_{k-2} & & z_{k-n} \end{array}\right] U_k \begin{bmatrix} a_1 \\ \vdots \\ a_n \\ \hline -b_1 \\ \vdots \\ b_n \end{bmatrix} \tag{2.37a}$$

or

$$\underline{z}_k = [z_k \mid U_k] \begin{bmatrix} -\underline{a}_n \\ \underline{b}_m \end{bmatrix} \tag{2.37b}$$

This solution is proposed by R.C.K. Lee [3].

An alternate approach is to use an estimate for the element x_i in (2.35), denoted by $\hat{x}_i$ in the matrix X_k. Thus

$$\hat{x}_i = [\hat{x}_{i-1} \cdots \hat{x}_{i-n} \mid u_{i-1} \cdots u_{i-m}] \begin{bmatrix} -\underline{\hat{a}}_n \\ \underline{\hat{b}}_m \end{bmatrix} \tag{2.38}$$

This approach, as shown in Example 2.3, has a distant advantage over the solution of (2.37b). This example also demonstrates the difference between the least-square-error and the normalized least-square-error solutions as given by (2.26) and (2.32).

Example 2.3: Identify the coefficients a_1 and b_1 in the difference equation

$$x_{k+1} = a_1 x_k + b_1 u_k \tag{2.39a}$$

from noise corrupted measurements z_{k+1}, where

$$z_{k+1} = x_{k+1} + \nu_{k+1} \tag{2.39b}$$

and ν_i is a sample of the measurement noise with the following statistical properties

$$\left.\begin{aligned} &E\{\nu_i\} = 0 \\ &E\{\nu_i \cdot \nu_j\} = \delta_{i,j}.\sigma^2 \quad , \quad \delta_{i,j} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \end{aligned}\right\} \tag{2.40}$$

where E{ } denotes the expected value, and σ^2 is the variance of the measurement noise. Specifically consider the discrete equation

$$x_{k+1} = .9x_k + .1\,u_k \tag{2.41}$$

For a given input sequence $\{u_1 \ldots u_k\}$ and zero initial conditions, the output sequence $\{x_0, x_1 \ldots x_k\}$ is generated using (2.41). The output sequence is then corrupted with noise, taken from a uniform distribution with maximum deviation ±0.1 with zero mean to obtain the sequence of noisy measurements $\{z_o, z_1 \ldots z_k\}$ which then are processed according to Eqs. (2.26) and (2.32). For both approaches, one using the z_i's and the other using the $\hat{x}_i$'s of Eq. (2.38) as elements in the coefficient matrix X, two computations are made - one where the input is a unit step and one where the input is a sampled cosine wave of unit amplitude.

Typical results using the same measurement data for both estimators (the least-square-error and the normalized least-square-error) are shown in Fig. 2.6 through 2.17. Figs. 2.6 and 2.7 present the estimates for a_1 and b_1 for the least-square-error and the normalized least-square-error solutions for a step input. Fig. 2.8 presents the magnitude of the total identification error $\varepsilon_k = |\hat{a}_1(k)-a_1| + |\hat{b}_1(k)-b_1|$ for both cases. Figs. 2.9, 2.10, 2.11 present the identifications using the estimated past values for X_k as given by (2.38). Figs. 2.12 through 2.17 present the corresponding data when the input is a cosine function. The

results indicate that the identification error is generally smaller when the normalized least-square-error method is used and that the identification error depends upon whether the input function causes a significant system response. In the following it will be demonstrated that for the step input the identification error approaches a very large value, compared with the parameters to be identified, independent of the variance of the measurement noise while for the cosine input the identification approaches a constant bias dependent on the variance of measurement noise. Consider the least-square-error solution of (2.37b) which takes the form

$$\begin{bmatrix} \hat{a}_1 \\ \vdots \\ \hat{a}_n \\ \hline \hat{b}_1 \\ \vdots \\ \hat{b}_m \end{bmatrix} = \{[Z_k \mid U_k]^T [Z_k \mid U_k]\}^{-1} [Z_k \mid U_k]^T \underline{z}_k \tag{2.42}$$

where $k \geq m+n$. For Example 2.3 this may be written as

$$\begin{bmatrix} \hat{a}_1 \\ \\ \hat{b}_1 \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{k-1} z_i^2 & \sum_{i=0}^{k-1} u_i z_i \\ \\ \sum_{i=0}^{k-1} u_i z_i & \sum_{i=0}^{k-1} u_i^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=0}^{k-1} z_i z_{i+1} \\ \\ \sum_{i=0}^{k-1} u_i z_{i+1} \end{bmatrix} \tag{2.43a}$$

or

$$\begin{bmatrix} \hat{a}_1 \\ \hat{b}_1 \end{bmatrix} = \begin{bmatrix} \frac{1}{k}\sum_{i=0}^{k-1} z_i^2 & \frac{1}{k}\sum_{i=0}^{k-1} u_i z_i \\ \frac{1}{k}\sum_{i=0}^{k-1} u_i z_i & \frac{1}{k}\sum_{i=0}^{k-1} u_i^2 \end{bmatrix}^{-1} \begin{bmatrix} \frac{1}{k}\sum_{i=0}^{k-1} z_i z_{i+1} \\ \frac{1}{k}\sum_{i=0}^{k-1} u_i z_{i+1} \end{bmatrix} \tag{2.43b}$$

In the limit as k goes to infinity the summations in (2.43b), provided that u_k and x_k remain bounded, take the following form

$$\lim_{k\to\infty} \frac{1}{k}\sum_{i=0}^{k-1} z_i^2 = \lim_{k\to\infty} \frac{1}{k}\left[\sum_{i=0}^{k-1} x_i^2 + 2\sum_{i=0}^{k-1} x_i\nu_i + \sum_{i=0}^{k-1} \nu_i^2\right]$$

$$= \lim_{k\to\infty} \frac{1}{k}\sum_{i=0}^{k-1} x_i^2 + \sigma^2$$

$$\lim_{k\to\infty} \frac{1}{k}\sum_{i=0}^{k-1} u_i z_i = \lim_{k\to\infty} \frac{1}{k}\left[\sum_{i=0}^{k-1} u_i x_i + \sum_{i=0}^{k-1} u_i \nu_i\right]$$

$$= \lim_{k\to\infty} \frac{1}{k}\sum_{i=0}^{k-1} u_i x_i$$

$$\lim_{k\to\infty} \frac{1}{k}\sum_{i=0}^{k-1} z_i z_{i+1} = \lim_{k\to\infty} \frac{1}{k}\left[\sum_{i=0}^{k-1} x_i x_{i+1} + \sum_{i=0}^{k-1} x_i \nu_{i+1} + \sum_{i=0}^{k-1} x_{i+1}\nu_i + \sum_{i=0}^{k-1} \nu_i \nu_{i+1}\right]$$

$$= \lim_{k\to\infty} \frac{1}{k}\sum_{i=0}^{k-1} x_i x_{i+1}$$

$$\lim_{k\to\infty} \frac{1}{k}\sum_{i=0}^{k-1} u_i z_{i+1} = \lim_{k\to\infty} \frac{1}{k}\left[\sum_{i=0}^{k-1} u_i x_i + \sum_{i=0}^{k-1} u_i \nu_{i+1}\right]$$

$$\lim_{k\to\infty} \frac{1}{k}\sum_{i=0}^{k-1} u_i x_i$$

Then for very large k the following relation is obtained from (2.43b)

$$\begin{bmatrix} \sum_{i=0}^{k-1} x_i^2 + k\sigma^2 & \sum_{i=0}^{u-1} u_i x_i \\ \sum_{i=0}^{k-1} u_i x_i & \sum_{i=0}^{k-1} u_i^2 \end{bmatrix} \begin{bmatrix} \hat{a}_1 \\ \hat{b}_1 \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{k-1} x_i x_{i+1} \\ \sum_{i=0}^{k-1} u_i x_{i+1} \end{bmatrix} \tag{2.44}$$

The corresponding expression for noiseless observations is

$$\begin{bmatrix} \sum_{i=0}^{k-1} x_i^2 & \sum_{i=0}^{k-1} u_i x_i \\ \sum_{i=0}^{k-1} x_i u_i & \sum_{i=0}^{k-1} u_i^2 \end{bmatrix} \begin{bmatrix} \hat{a}_1 \\ \hat{b}_1 \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{k-1} x_i x_{i+1} \\ \sum_{i=0}^{k-1} u_i x_{i+1} \end{bmatrix} \tag{2.45}$$

or

$$\begin{bmatrix} \sum_{i=0}^{k-1} x_i^2 + k\sigma^2 & \sum_{i=0}^{k-1} u_i x_i \\ \sum_{i=0}^{k-1} u_i x_i & \sum_{i=0}^{k-1} u_i^2 \end{bmatrix} \begin{bmatrix} a_1 \\ b_1 \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{k-1} x_i x_{i+1} \\ \sum_{i=0}^{k-1} u_i x_{i+1} \end{bmatrix} + \begin{bmatrix} k\sigma^2 a_1 \\ 0 \end{bmatrix} \tag{2.46}$$

The identification error for very large k is then obtained by combining (2.44) and (2.46)

$$\begin{bmatrix} \sum_{i=0}^{k-1} x_i^2 + k\sigma^2 & \sum_{i=0}^{k-1} u_i x_i \\ \sum_{i=0}^{k-1} u_i x_i & \sum_{i=0}^{k-1} u_i^2 \end{bmatrix} \begin{bmatrix} \hat{a}_1 - a_1 \\ \hat{b}_1 - b_1 \end{bmatrix} = \begin{bmatrix} k\sigma^2 a_1 \\ 0 \end{bmatrix}$$

or

$$\begin{bmatrix} \tilde{a}_1 \\ \\ \tilde{b}_1 \end{bmatrix} = \begin{bmatrix} \hat{a}_1 \\ \\ \hat{b}_1 \end{bmatrix} - \begin{bmatrix} a_1 \\ \\ b_1 \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{k-1} x_i^2 + k\sigma^2 & \sum_{i=0}^{k-1} u_i x_i \\ \\ \sum_{i=0}^{k-1} u_i x_i & \sum_{i=0}^{k-1} u_i^2 \end{bmatrix}^{-1} \begin{bmatrix} -k\sigma^2 a_1 \\ \\ 0 \end{bmatrix} \tag{2.47}$$

In the limit as k goes to infinity, Eq. (2.47) is easily evaluated by considering the corresponding continuous system with the transfer function

$$\frac{x(s)}{u(s)} = \frac{1}{s+1} \tag{2.48}$$

Integration of (2.48) yields (2.39a) when the forcing function u(t) is approximated as piecewise constant and the sampling time Δt satisfies the relation

$$e^{-\Delta t} = a_1$$

or

$$|\Delta t| = |\ell n\ a_1| \tag{2.49}$$

For the step input, when the system is initially at rest

$$\left.\begin{aligned} u(t) &= 1 \\ x(t) &= 1 - e^{-t} \end{aligned}\right\} \tag{2.50}$$

Then

$$\left.\begin{aligned}
\lim_{k\to\infty} \sum_{i=0}^{k-1} x_i^2 &= \lim_{t\to\infty} \int_0^t x^2(t)dt = \lim_{t\to\infty} t \\
\lim_{k\to\infty} \sum_{i=0}^{k-1} u_i x_i &= \lim_{t\to\infty} \int_0^t u(t)x(t)dt = \lim_{t\to\infty} t \\
\lim_{k\to\infty} \sum_{i=0}^{k-1} u_i^2 &= \lim_{t\to\infty} t \\
\lim_{k\to\infty} \sigma^2 \sum_{i=0}^{k-1} 1 &= \lim_{t\to\infty} t\sigma^2
\end{aligned}\right\} \qquad (2.51)$$

and

$$\begin{aligned}
\lim_{k\to\infty}\begin{bmatrix}\tilde{a}_1\\ \tilde{b}_1\end{bmatrix} &= \lim_{t\to\infty}\left\{\begin{bmatrix}t(1+\sigma^2) & t\\ t & t\end{bmatrix}^{-1}\begin{bmatrix}-t\sigma^2 a_1\\ 0\end{bmatrix}\right\} \\
&= \lim_{t\to\infty}\left\{\frac{1}{\sigma^2 t^2}\begin{bmatrix}t & -t\\ -t & (1+\sigma^2)t\end{bmatrix}\begin{bmatrix}-t\sigma^2 a_1\\ 0\end{bmatrix}\right\} \\
&= \begin{bmatrix}-a_1\\ +a_1\end{bmatrix} \qquad (2.52)
\end{aligned}$$

also

$$\lim_{k\to\infty}\begin{bmatrix}\hat{a}_1\\ \hat{b}_1\end{bmatrix} = \begin{bmatrix}a_1\\ b_1\end{bmatrix} + \lim_{k\to\infty}\begin{bmatrix}\tilde{a}_1\\ \tilde{b}_1\end{bmatrix} = \begin{bmatrix}0\\ b_1+a_1\end{bmatrix} \qquad (2.53)$$

This demonstrates that identification of system parameters from a step input is only feasible as long as the system response is not close to the final value, as shown in Fig. 2.6.

Similar considerations yield the estimation error for the unit cosine forcing function. Using (2.48) and

$$U(s) = \frac{s}{s^2+1} \tag{2.54}$$

it follows

$$x(s) = \frac{s}{(s^2+1)(s+1)} = \frac{-\frac{1}{2}}{s+1} + \frac{\frac{1}{2}s+\frac{1}{2}}{s^2+1} \tag{2.55}$$

$$\left.\begin{aligned} \text{and} \quad u(t) &= \cos(t) \\ x(t) &= (\cos t + \sin t - e^{-t}) \end{aligned}\right\} \tag{2.56}$$

also

$$\left.\begin{aligned} \lim_{k\to\infty} \sum_{i=0}^{k-1} x_i^2 &= \lim_{t\to\infty} \int_0^t x^2(t)dt = \lim_{t\to\infty} \frac{1}{4}t \\ \lim_{k\to\infty} \sum_{i=0}^{k-1} u_i x_i &= \lim_{t\to\infty} \int_0^t u(t)x(t)dt = \lim_{t\to\infty} \frac{1}{4}t \\ \lim_{k\to\infty} \sum_{i=0}^{k-1} u_i^2 &= \lim_{t\to\infty} \int_0^t u^2(t)dt = \lim_{t\to\infty} \frac{1}{2}t \end{aligned}\right\} \tag{2.57}$$

Then from (2.47)

$$\lim_{k\to\infty} \begin{bmatrix} \tilde{a}_1 \\ \tilde{b}_1 \end{bmatrix} = \lim_{t\to\infty} \left\{ \begin{bmatrix} (\frac{1}{4}+\sigma^2)t & \frac{1}{4}t \\ \frac{1}{4}t & \frac{1}{2}t \end{bmatrix}^{-1} \begin{bmatrix} -t\sigma^2 a_1 \\ 0 \end{bmatrix} \right\}$$

$$= \frac{-\sigma^2 a_1}{\frac{1}{16}+\frac{1}{2}\sigma^2} \begin{bmatrix} \frac{1}{2} & \frac{1}{4} \\ \frac{1}{4} & (\frac{1}{4}+\sigma^2) \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$= \frac{a_1}{1 + \frac{1}{8\sigma^2}} \begin{bmatrix} -1 \\ \frac{1}{2} \end{bmatrix} \tag{2.58}$$

The identification error as shown in Eq. (2.58) approaches a constant bias which is reasonably small for a large signal to noise ratio. For the specific example (Ex. 2.3) this bias as obtained from (2.47) for very large k is

$$\begin{bmatrix} \tilde{a}_1 \\ \tilde{b}_1 \end{bmatrix} \simeq \begin{bmatrix} -.0212 \\ .0106 \end{bmatrix}$$

for the least-square-error solution and

$$\begin{bmatrix} \tilde{a}_1 \\ \tilde{b}_1 \end{bmatrix} \simeq \begin{bmatrix} -.0163 \\ .0054 \end{bmatrix}$$

for the normalized least-square-error solution. The total bias error reduction for the normalized least-square-error solution is approximately 32% compared with the least-square-error solution. This agrees with the experimental results obtained as shown in Fig. 2.14.

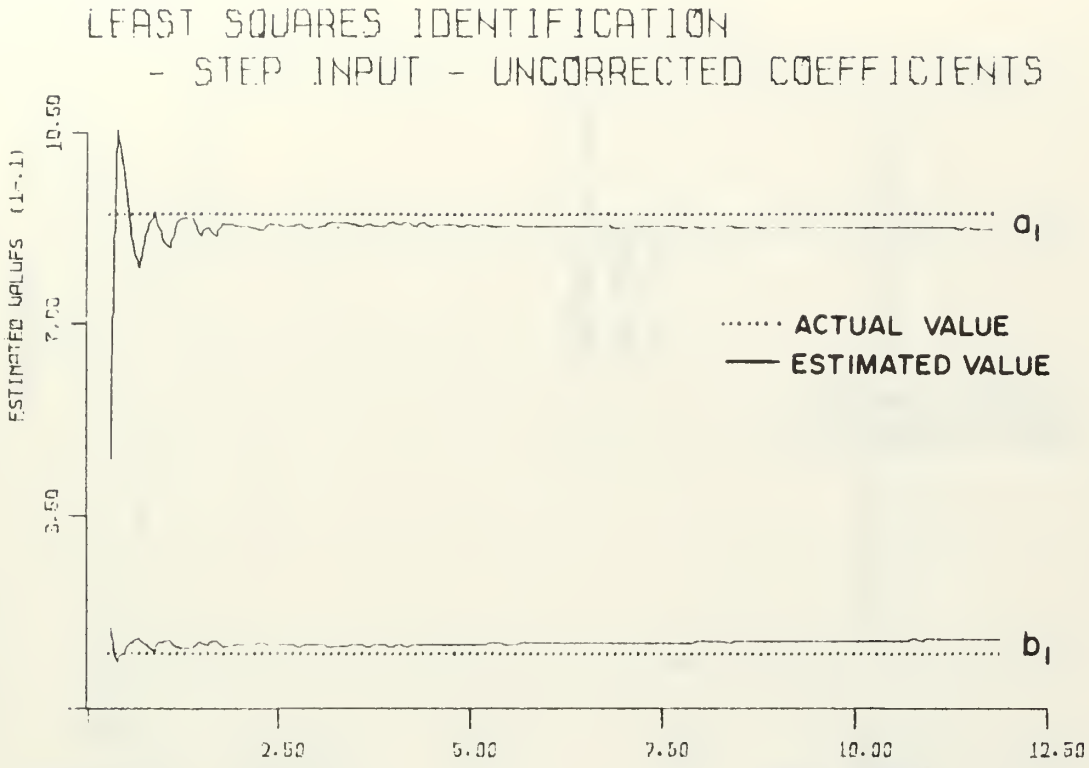

FIG 2.6 EX. 2.3

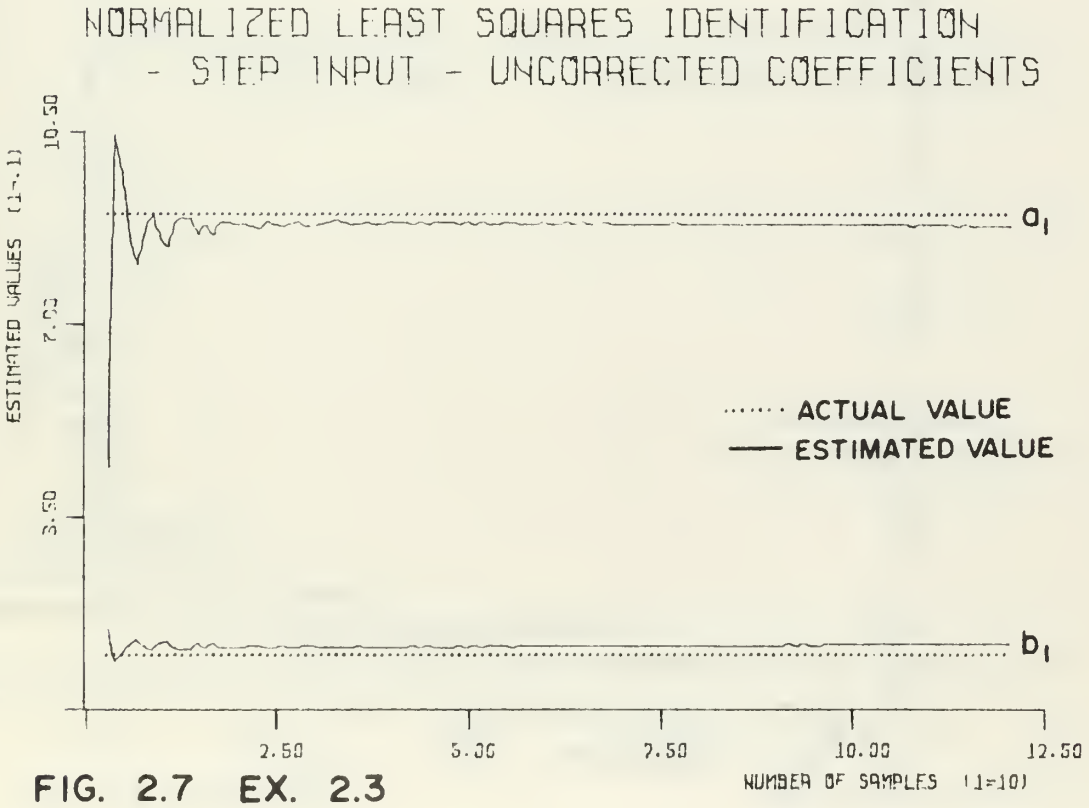

FIG. 2.7 EX. 2.3

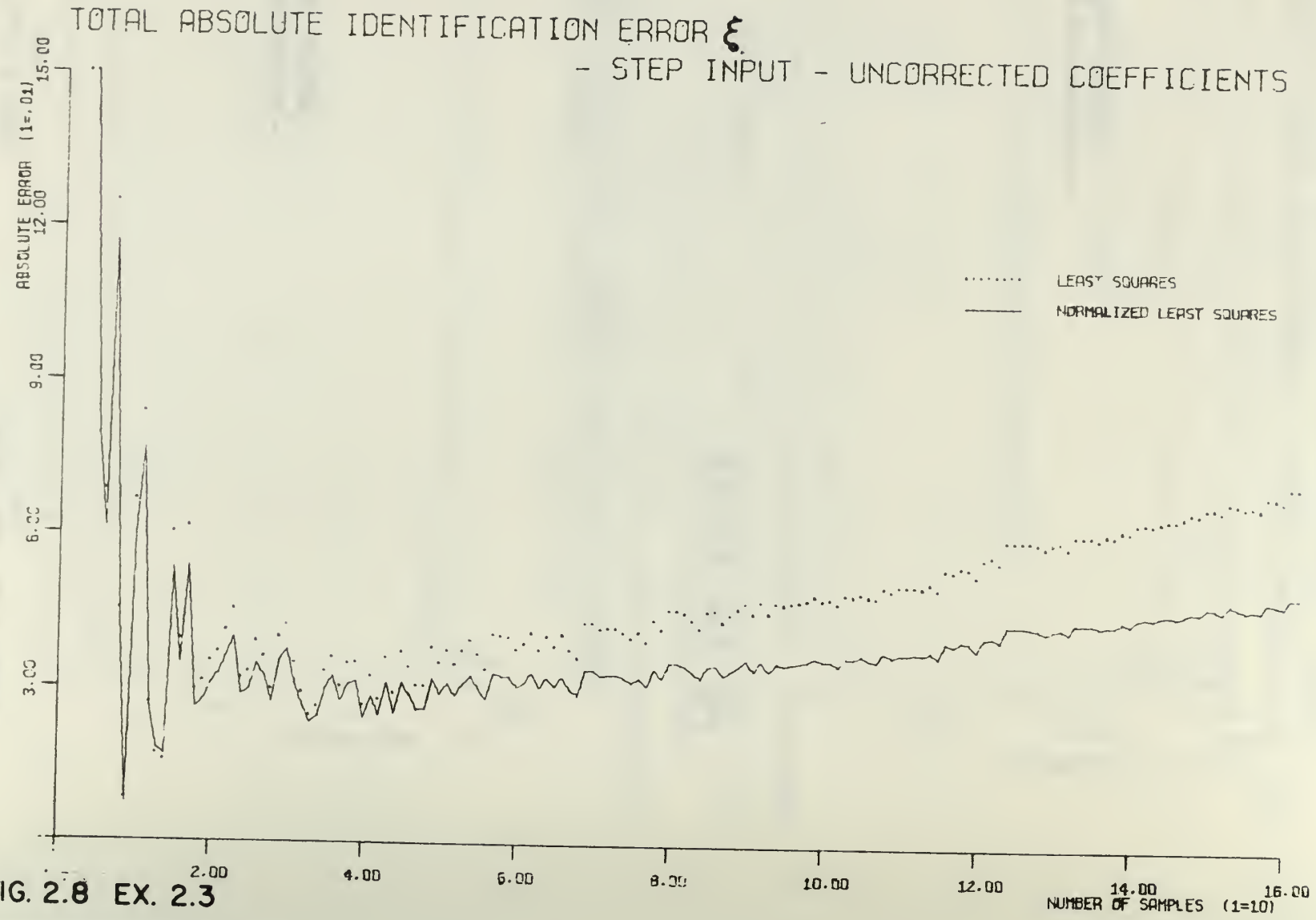
TOTAL ABSOLUTE IDENTIFICATION ERROR ξ
- STEP INPUT - UNCORRECTED COEFFICIENTS
ABSOLUTE ERROR (1=.01)
15.00
12.00
9.00
6.00
3.00
LEAST SQUARES
NORMALIZED LEAST SQUARES
2.00
4.00
6.00
8.00
10.00
12.00
14.00
16.00
NUMBER OF SAMPLES (1=10)

FIG. 2.8 EX. 2.3

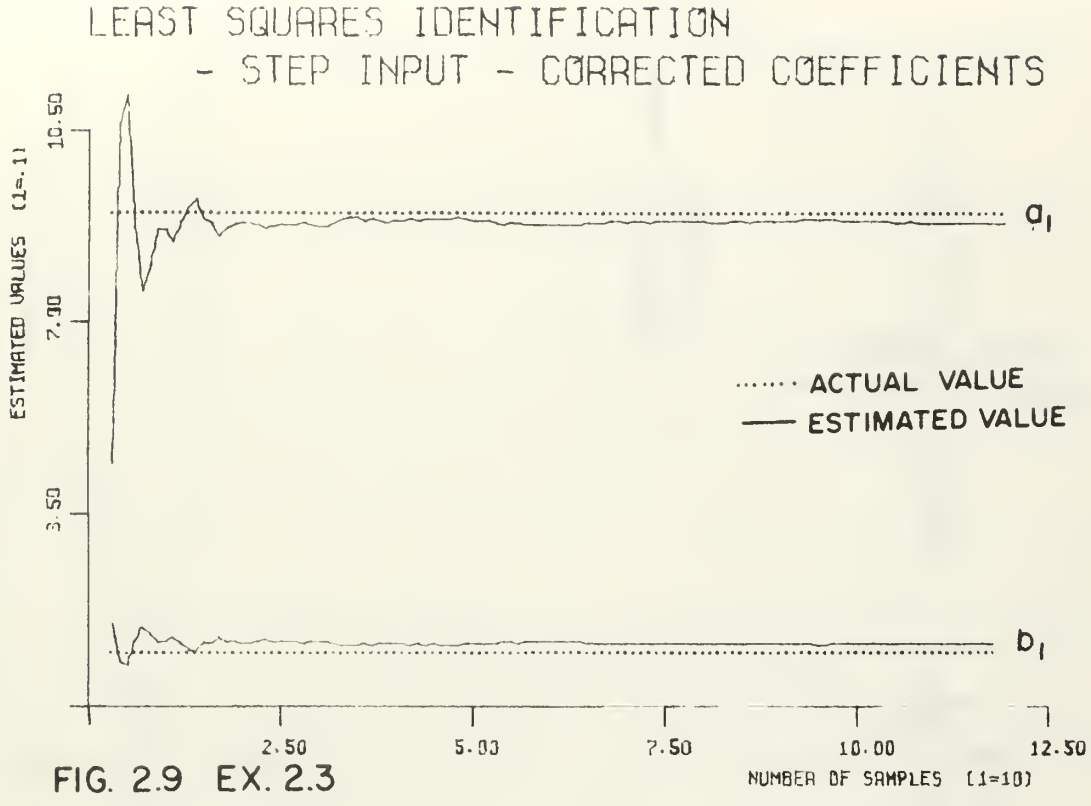

FIG. 2.9 EX. 2.3

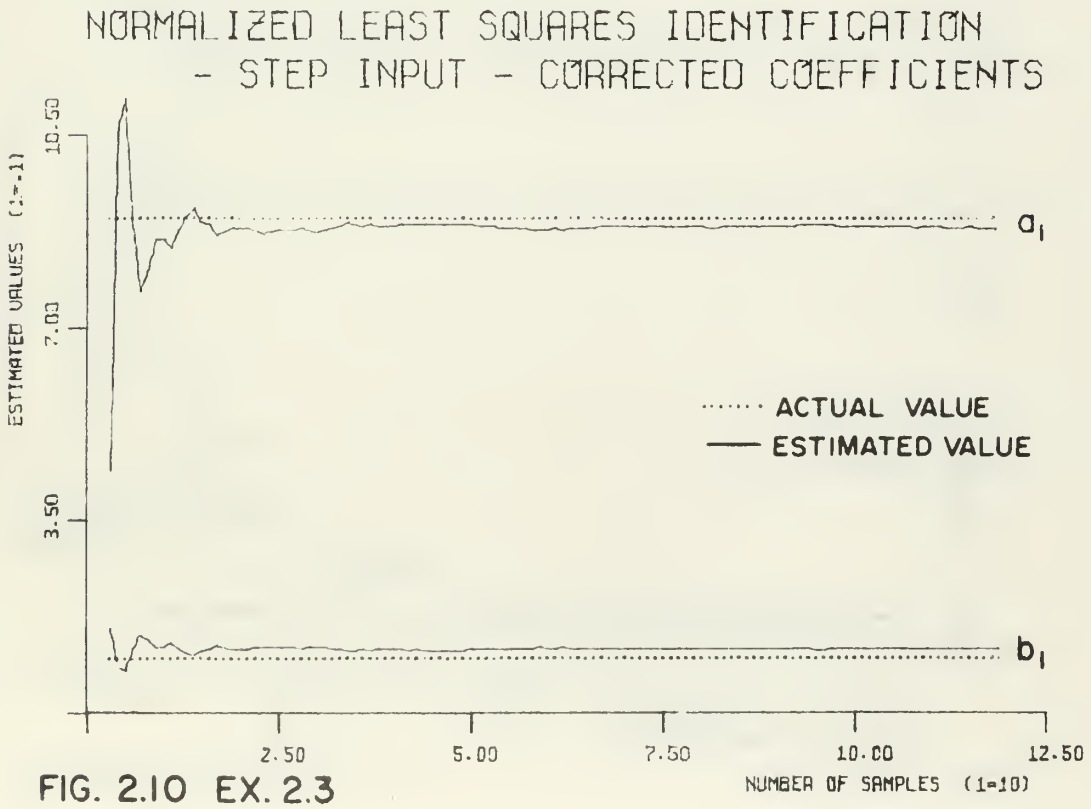

FIG. 2.10 EX. 2.3

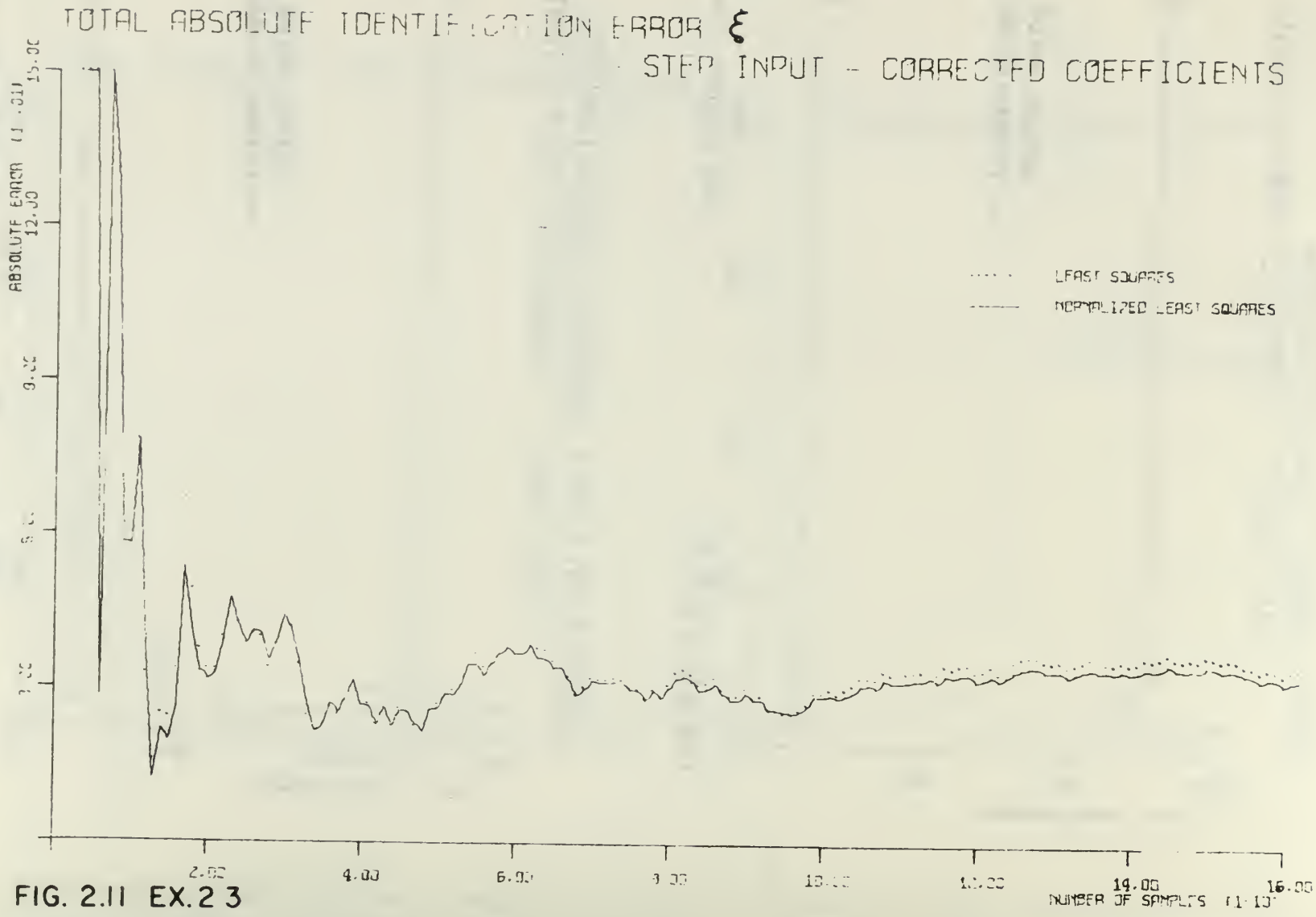
TOTAL ABSOLUTE IDENTIFICATION ERROR ξ
STEP INPUT - CORRECTED COEFFICIENTS
ABSOLUTE ERROR
12.00
LEAST SQUARES
NORMALIZED LEAST SQUARES
2.00
4.00
6.00
16.00
NUMBER OF SAMPLES

FIG. 2.11 EX. 2 3

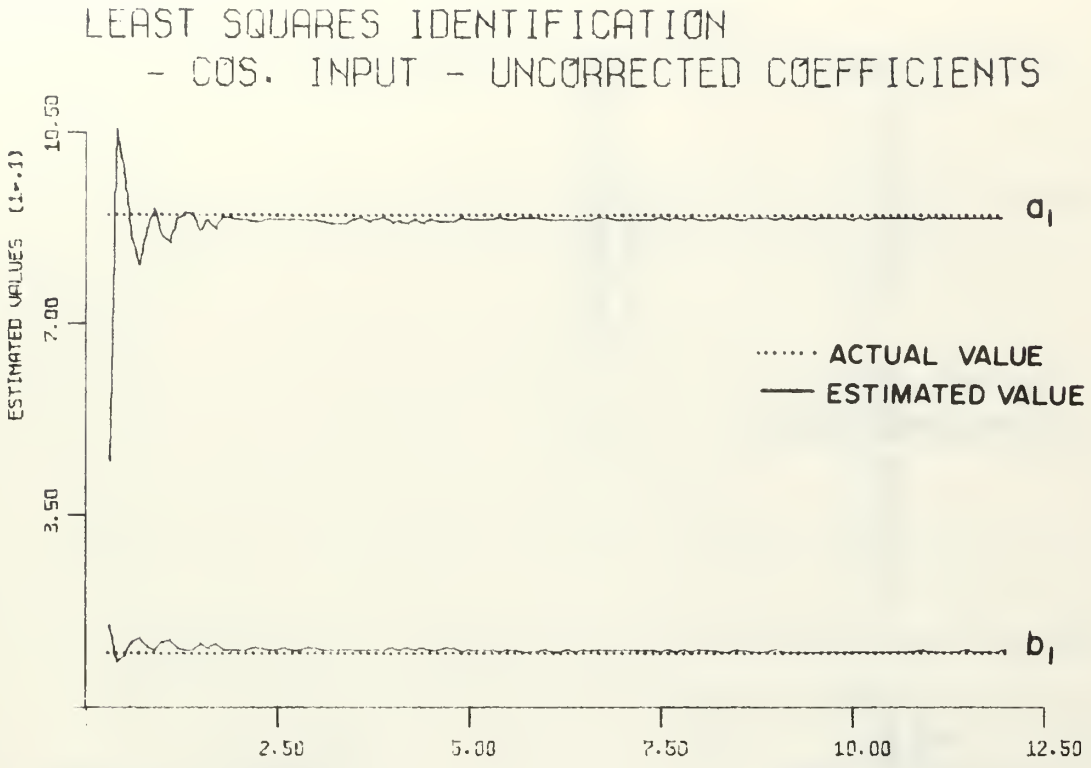

FIG. 2.12 EX. 2.3

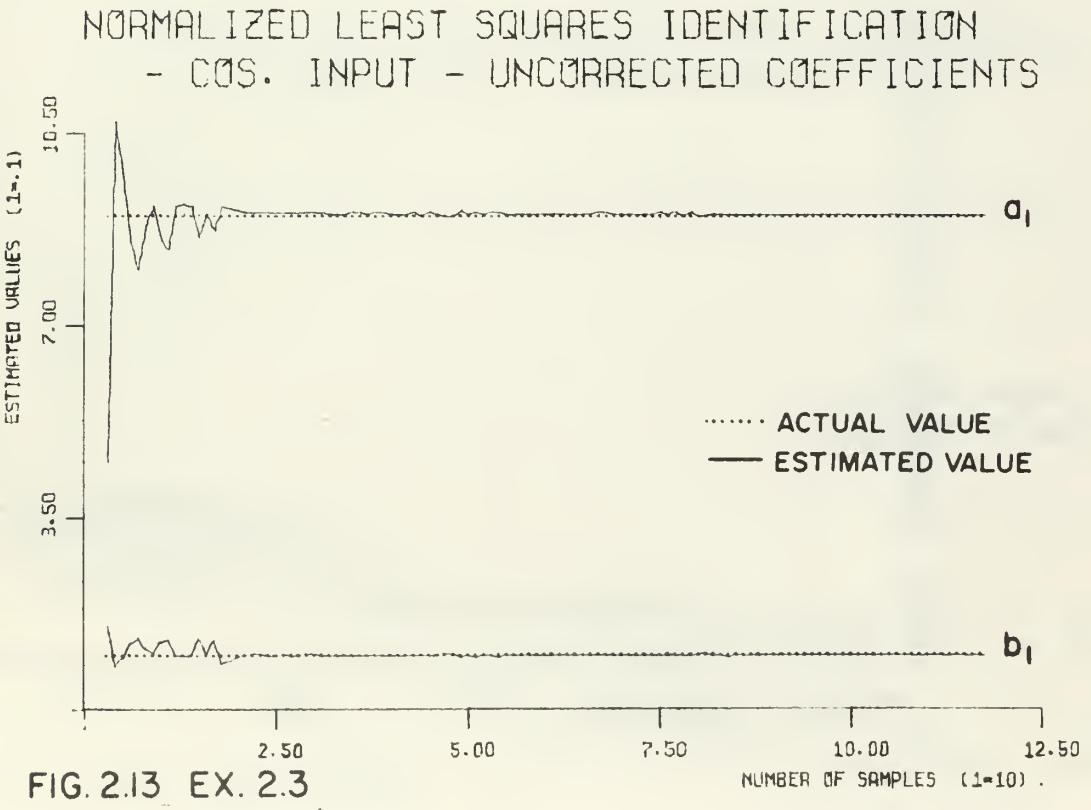

FIG. 2.13 EX. 2.3

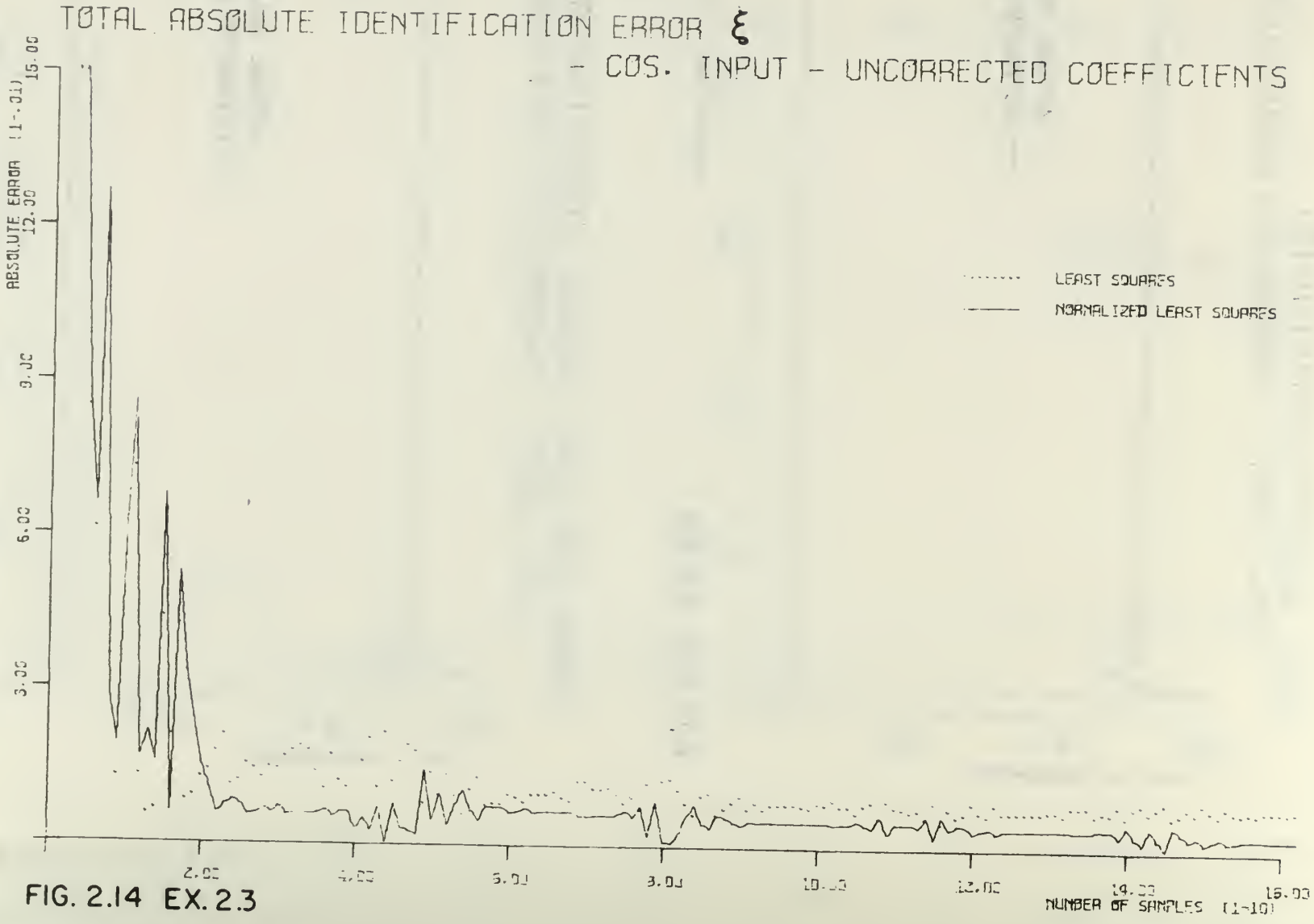
TOTAL ABSOLUTE IDENTIFICATION ERROR ξ
- COS. INPUT - UNCORRECTED COEFFICIENTS
ABSOLUTE ERROR (1-.01)
15.00
12.00
9.00
6.00
3.00
LEAST SQUARES
NORMALIZED LEAST SQUARES
2.00
4.00
6.00
8.00
10.00
12.00
14.00
16.00
NUMBER OF SAMPLES (1-10)

FIG. 2.14 EX. 2.3

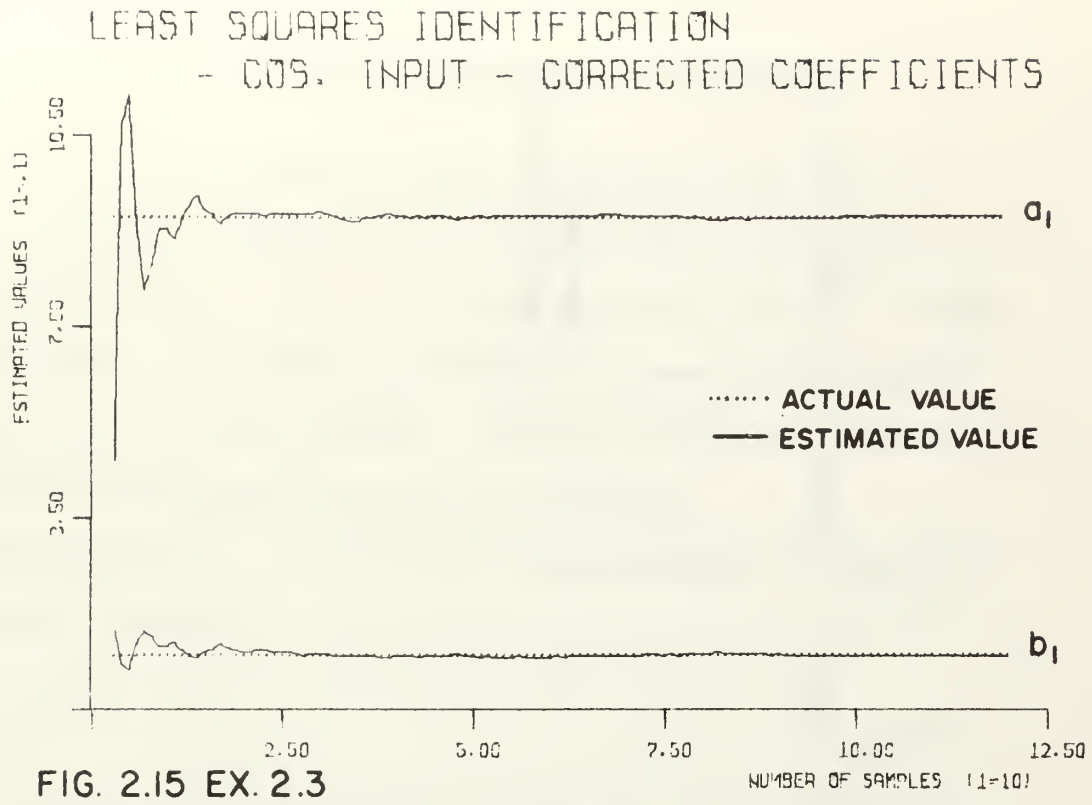

FIG. 2.15 EX. 2.3

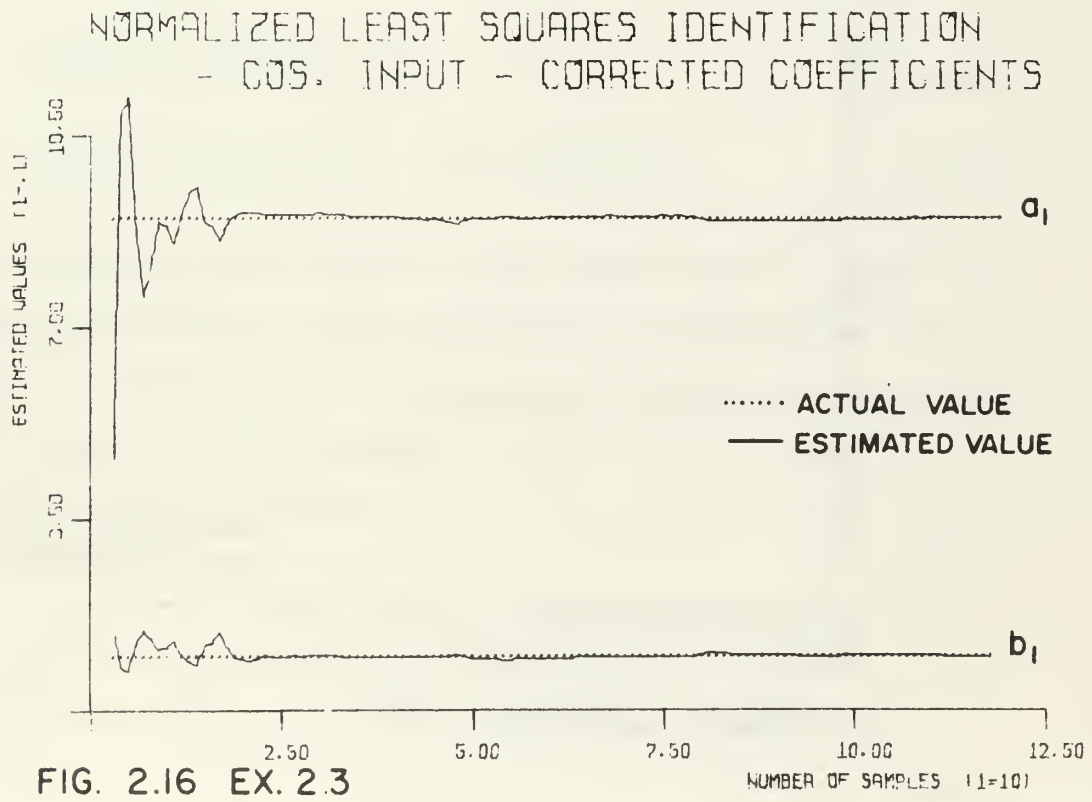

FIG. 2.16 EX. 2.3

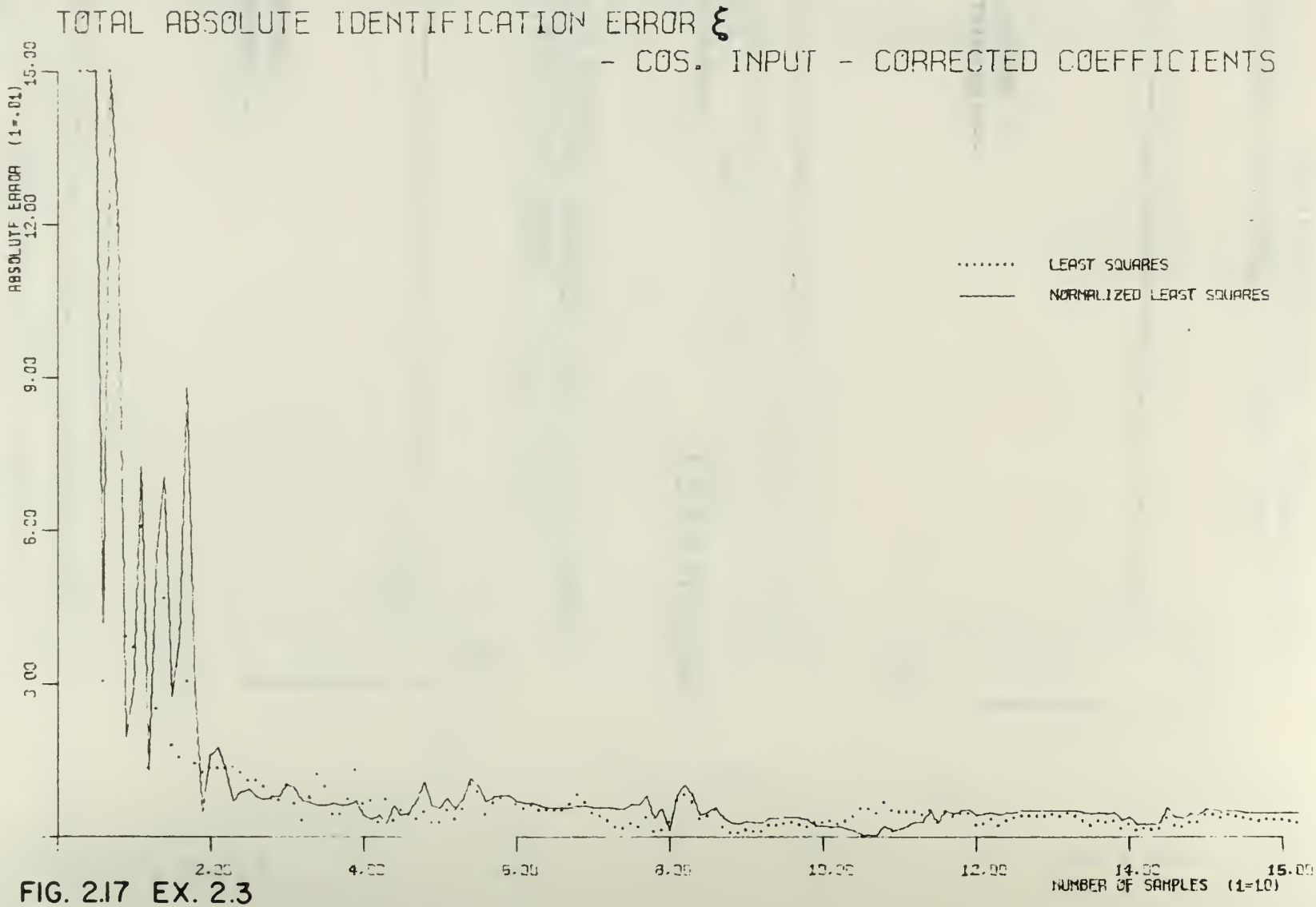
TOTAL ABSOLUTE IDENTIFICATION ERROR ξ
- COS. INPUT - CORRECTED COEFFICIENTS
ABSOLUTE ERROR (1=.01)
15.00
12.00
9.00
6.00
3.00
LEAST SQUARES
NORMALIZED LEAST SQUARES
2.00
4.00
6.00
8.00
10.00
12.00
14.00
15.00
NUMBER OF SAMPLES (1=10)

FIG. 2.17 EX. 2.3

III. THE BEST APPROXIMATE SOLUTION

In estimation theory, quite frequently one has to solve a set of inconsistent or insufficient specified linear equations. Since in these cases an exact or unique solution does not exist, an optimal or best approximate solution has to be accepted. Penrose [4,5] has defined this best approximate solution as follows.

Definition 1: X_o is the best approximate solution of the linear equation

$$f(X) = G, \tag{3.1}$$

where X and G are rectangular matrices, if for all $X \neq X_o$ either

$$\|f(X) - G\| > \|f(X_o) - G\| \tag{3.2a}$$

or $$\|f(X) - G\| = \|f(X_o) - G\| \quad \text{and} \quad \|X\| \geq \|X_o\| \ , \tag{3.2b}$$

where $\|X\|$ denotes the norm of X defined as $\|X\| = \text{trace } X^T X$.

In the discussion which follows, X is restricted to be a vector of dimensions n x 1 with real elements denoted by $\underline{x}$. Then Eq. (3.1) may be written as

$$A\underline{x} = \underline{b} \tag{3.3}$$

where A is an m x n matrix and $\underline{b}$ is an m x 1 vector.

The best approximate solution for $\underline{x}$, according to definition 1, is the least-square-error solution if A has

maximum rank, and the minimum-norm least-square-error solution if A has rank less than maximum. This solution is obtained by using a generalized matrix inverse developed by Penrose [4], and denoted in the work which follows as the pseudo inverse. The solution to (3.3) is thus written as

$$\underline{\hat{x}} = A^{+}\underline{b} \tag{3.4}$$

where $\underline{\hat{x}}$ is the best approximate solution and A^{+} the pseudo inverse.

In the following the definition and some properties of the pseudo inverse, as given by Penrose, are stated. Then some alternate expressions for the pseudo inverse are discussed. Finally a new recursive formulation for the sequential solution of Eq. (3.3) is presented. This formulation has the advantage over previously published results (Wells [6]) in that the dimensions of the matrices in the algorithm remain constant irrespective of the size and rank of A. A flow diagram for the computation of the algorithm is presented and illustrated with a numerical example. Finally, the recursive algorithm is adapted to the problem of estimating the states of time-varying, linear systems from noise-contaminated measurements.

A. THE PSEUDO INVERSE

1. Definition and Properties

Penrose [4] defines the following

Definition 2: Four matrix equations are defined

$$AYA = A \tag{3.5a}$$

$$YAY = Y \tag{3.5b}$$

$$[AY]^* = AY \tag{3.5c}$$

$$[YA]^* = YA \tag{3.5d}$$

where * denotes the conjugate transpose. These equations have a unique solution for Y. This solution is called the pseudo inverse and is denoted by $Y = A^+$.

The essential feature of this definition is that any expression for the inverse of matrix A is established as the unique pseudo inverse if and only if it satisfies Eq. (3.5). As a consequence of definition 2 the pseudo inverse has the following properties

$$A^{++} = A \tag{3.6a}$$

$$A^{*+} = A^{+*} \tag{3.6b}$$

$$A^+ = A^{-1} \text{ if A is nonsingular} \tag{3.6c}$$

$$(\lambda A)^+ = \lambda^{-1} A^+ \tag{3.6d}$$

$$(A^*A)^+ = A^+A^* \tag{3.6e}$$

A, A^*A, A^+, A^+A have rank equal to the trace of A^+A

In addition, for completeness, define [1]

$$0^+ = 0^T \tag{3.6f}$$

2. Alternate Expressions for the Pseudo Inverse

It is desirable to be able to express this inverse by a mathematical formula and the following results are

essentially available in the literature as discussed by Deutsch [1], Koenig [6], et. al.

a. Overdetermined case, $m \geq n$, $r=n$

As shown in Chapter I, the solution (3.4) is obtained using

$$A^{+} = [A^{T}A]^{-1}A^{T} \tag{3.7}$$

which corresponds to the minimum mean-square-error solution of (3.3).

b. Underdetermined case, $m \leq n$, $r=m$

The solution (2.4) is obtained using

$$A^{+} = A^{T}[AA^{T}]^{-1} \tag{3.8}$$

which corresponds to the minimum-norm solution.

Equation (3.8) satisfies definition 2 and is thus the desired pseudo inverse. The fact that the solution is the minimum-norm solution can be demonstrated geometrically for the three-dimensional case as follows:

Given two equations in three unknowns

$$\begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \tag{3.9a}$$

Find the minimum-norm solution for the unknown vector $[x\ y\ z]^{T}$.

Eqs. (3.9a) represent two planes

$$\underline{r} \cdot \underline{a} = c_1 \tag{3.9b}$$

$$\underline{r} \cdot \underline{b} = c_2 \tag{3.9c}$$

where $\underline{r}$ is the position vector from the origin to the point (x,y,z)

$\underline{a}$ a vector with components a_1, a_2, a_3

and $\underline{b}$ a vector with components b_1, b_2, b_3

Normal vectors to the planes are given by $\underline{a}$ and $\underline{b}$

Then any point on the line of intersection of the two planes satisfies (3.9b) and (3.9c) and the desired solution is the point on the line of intersection closest to the origin. Let the vector from the origin O to this point N be designated by $\underline{ON}$ (see Fig. 3.1). $\underline{ON}$ is a linear combination of the vectors $\underline{a}$ and $\underline{b}$

$$\underline{ON} = \gamma_1\underline{a} + \gamma_2\underline{b} \quad ; \quad \underline{ON} = \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} \tag{3.9d}$$

where γ_1 and γ_2 are scalars, which are determined from the condition that $\underline{ON}$ has to satisfy (3.9b) and (3.9c) as the position vector $\underline{r}$.

Thus

$$(\gamma_1\underline{a} + \gamma_2\underline{b}) \cdot \underline{a} = c_1$$

$$(\gamma_1\underline{a} + \gamma_2\underline{b}) \cdot \underline{b} = c_2$$

and

$$\begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} = \begin{bmatrix} \underline{a}\cdot\underline{a} & \underline{a}\cdot\underline{b} \\ \underline{a}\cdot\underline{b} & \underline{b}\cdot\underline{b} \end{bmatrix}^{-1} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \tag{3.9e}$$

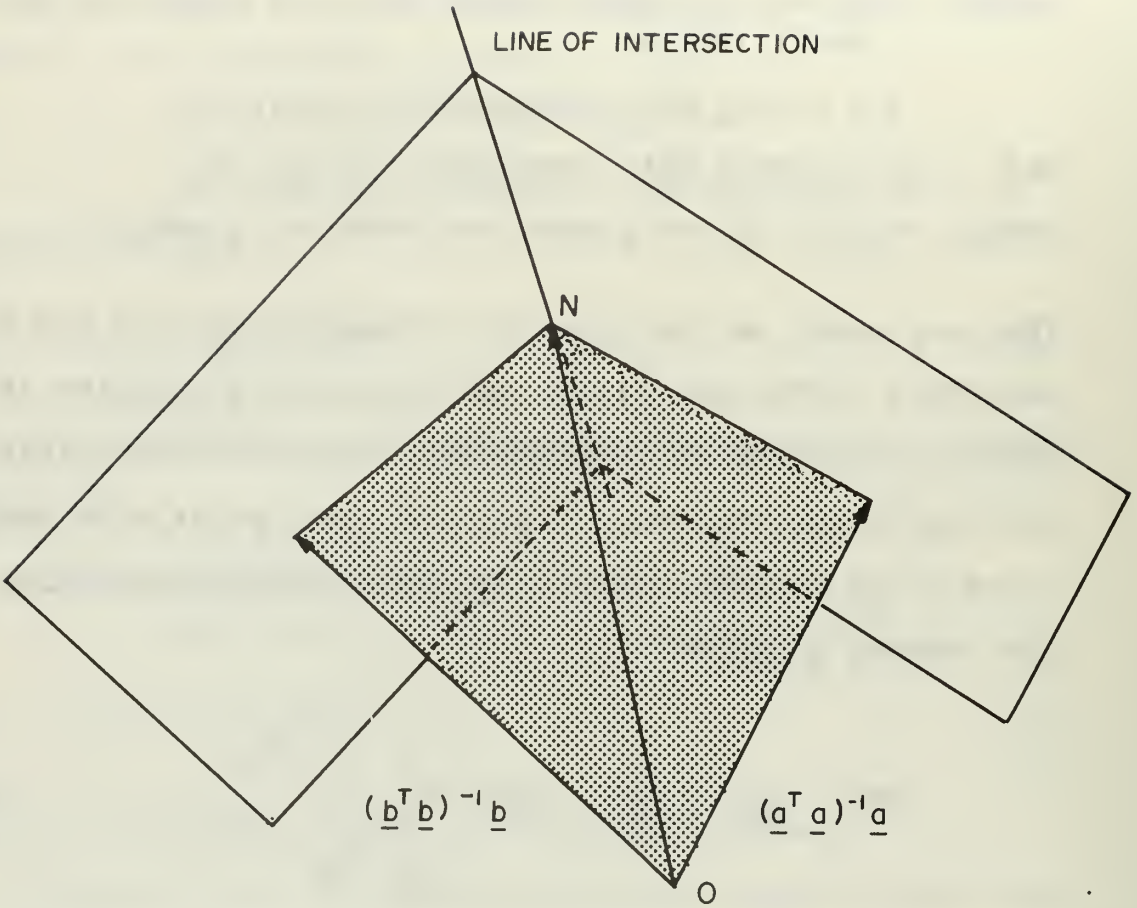

FIG. 3.1 MINIMUM-NORM SOLUTION $\underline{ON}$.

Equation (3.9d) may be written in component form using (3.9e) as

$$\begin{bmatrix} x_N \\ y_N \\ z_N \end{bmatrix} = \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{bmatrix} \begin{bmatrix} \sum_{i=1}^{3} a_i^2 & \sum_{i=1}^{3} a_i b_i \\ \sum_{i=1}^{3} a_i b_i & \sum_{i=1}^{3} b_i^2 \end{bmatrix}^{-1} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \tag{3.9g}$$

or

$$\begin{bmatrix} x_N \\ y_N \\ z_N \end{bmatrix} = \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{bmatrix} \left\{ \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{bmatrix} \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{bmatrix} \right\}^{-1} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \tag{3.10a}$$

If the matrix A is defined as in (3.9a)

$$A = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{bmatrix} \tag{3.10b}$$

the result (3.10a) may be written in the form of (3.4)

$$\begin{bmatrix} x_N \\ y_N \\ z_N \end{bmatrix} = A^+ \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \tag{3.10b}$$

where $A^+ = A^T[AA^T]^{-1}$.

This shows that in this case the pseudo inverse in (3.4) results in the minimum-norm solution.

c. Underdetermined case, $m \geq n$, $r < n$ or $m \leq n$, $r < m$

The solution (3.4) is obtained using either

$$A^+ = A^T N \left[N^T A A^T N\right]^{-1} N^T \tag{3.14a}$$

or $$A^{+} = M^{T}[MA^{T}AM^{T}]^{-1}MA^{T} \qquad (3.14b)$$

which corresponds to the minimum norm solution of minimum square error. Matrices N and M are defined as factors of A [1]

$$A = N \cdot M \qquad (3.11)$$

where matrix N is of dimension m x r, and matrix M is of dimension r x n. The rows of N are chosen such that they constitute a set of base vectors for the column space spanned by A. Matrix M is then the transformation of N to A. Its dimensions are necessarily r x n. For example, the columns of N might be chosen as all the independent columns in A. The pseudo inverse is then given as

$$A^{+} = M^{T}[MM^{T}]^{-1}\ [N^{T}N]^{-1}N^{T} \qquad (3.12)$$

because (3.12) satisfies all four equations in definition 1 as indicated below:

(1) $$\begin{aligned} A^{+}A &= M^{T}[MM^{T}]^{-1}[N^{T}N]^{-1}N^{T}NM \\ &= M^{T}[MM^{T}]^{-1}M \\ &= [A^{+}A]^{T} \end{aligned}$$

(2) $$\begin{aligned} AA^{+} &= NM\ M^{T}[MM^{T}]^{-1}[N^{T}N]^{-1}N^{T} \\ &= N[N^{T}N]^{-1}N^{T} \\ &= [AA^{+}]^{T} \end{aligned}$$

(3) $$\begin{aligned} AA^{+}A &= NM\ M^{T}[MM^{T}]^{-1}[N^{T}N]^{-1}N^{T}NM \\ &= NM \\ &= A \end{aligned}$$

$$(4)\quad A^+AA^+ = M^T[MM^T]^{-1}[N^TN]^{-1}N^TNMM^T[MM^T]^{-1}[N^TN]^{-1}N^T$$
$$= M^T[MM^T]^{-1}[N^TN]^{-1}N^T$$
$$= A^+$$

Expression (3.12) which involves two matrix inversions may be simplified further to expressions involving only one matrix inversion. Since both matrices M and N have rank r, (3.11) can be solved for either one

$$M = [N^TN]^{-1}N^TA \tag{3.13a}$$

$$N = AM^T[MM^T]^{-1} \tag{3.13b}$$

Substitution of (3.13a) into (3.12) yields

$$A^+ = A^TN[N^TAA^TN]^{-1}N^T \tag{3.14a}$$

and substitution of (3.13b) into (3.12) results in

$$A^+ = M^T[MA^TAM^T]^{-1}MA^T \tag{3.14b}$$

All expressions, (3.12), (3.14a), and (3.14b), are valid general expressions for the pseudo inverse. However, if the matrices involved in actual computation are to be as small in dimensions as possible, (3.14a) and (3.14b) should be used as follows:

(1) $m \leq n$, $r \geq m$

$$A^+ = A^TN[N^TAA^TN]^{-1}N^T$$

(2) $m \geq n$, $r \leq n$

$$A^+ = M^T[MA^TAM^T]^{-1}MA^T$$

B. RECURSIVE ALGORITHM FOR THE SEQUENTIAL LEAST-SQUARE FIT

In Chapter I a recursive relationship for sequential estimation based on the equation

$$\underline{b} = A\underline{x} \tag{3.15}$$

and its least-square-error solution

$$\hat{\underline{x}} = [A^T A]^{-1} A^T \underline{b} \tag{3.16}$$

is given. However this recursive form is only valid if the matrix $[A^T A]$ is nonsingular such that its inverse $P = [A^T A]^{-1}$ exists.

Consider now the case where no assurance as to the existence of the inverse can be given. Using the pseudo inverse it is possible to write formally

$$\begin{aligned} \hat{\underline{x}} &= A^+ \underline{b} \\ &= A^+ [AA^+]^T \underline{b} \\ &= A^+ A^{T+} A^T \underline{b} \\ &= [A^T A]^+ A^T \underline{b} \end{aligned} \tag{3.17a}$$

or $$\hat{\underline{x}} = P\, A^T \underline{b}$$

where P is the pseudo inverse of $[A^T A]$. The dimensions of the matrix P are always n x n independent of m or r. This is significant because in a sequential procedure both m and r increase as more data are incorporated. Therefore, the use of an algorithm updating the matrix A^+, as proposed by T.N.E. Greyville [22], may not be practical for sequential estimation because the dimension, m, of A^+ grows at each

step. Alternate methods [7,8] using an updating procedure for A^+ until the matrix A has dimension n x n are considered below:

(1) Direct updating of A^+.

At each step, as additional measurements are incorporated, the size of A^+ grows one column for each step. C. H. Wells [7,8] presents an algorithm for the updating of A^+. However his procedure has the disadvantage that when initiating an estimation problem the rank of A must increase at each step until maximum rank is reached. This is not the case if the first n equations in (3.15) are dependent.

(2) Updating A^+ using (3.8) as long as $m \leq n$.

A recursive algorithm or direct computation based on (3.2) is possible only if the square matrix AA^T, of growing dimensions, remains non-singular. Thus the rank of A has to increase at each step which may not be true. Furthermore, a recursive form could be used only as a starting procedure up until the matrix AA^T has dimensions n x n.

Consequently, it is desirable to find a recursive formulation similar to (2.26), where all matrices involved have constant dimensions regardless of rank or size of A. This result is accomplished here with a recursive form for the matrix P, where P is the pseudo inverse of A^TA.

In order to derive this recursive algorithm, consider the set of equations

$$\begin{bmatrix} \underline{z}_k \end{bmatrix} = \begin{bmatrix} A_k \end{bmatrix} \begin{bmatrix} \underline{x} \end{bmatrix} \tag{3.18a}$$

where the subscript k denotes the number of equations. Assume the solution of the form (3.17a)

$$\begin{aligned} \hat{\underline{x}}_k &= [A_k^T A_k]^+ A_k^T \underline{z}_k \\ &= P_k A_k^T \underline{z}_k \end{aligned} \tag{3.19a}$$

where P_k is the pseudo inverse of $A_k^T A_k$ and $\hat{\underline{x}}_k$ the best approximate estimate for $\underline{x}$ based upon the last k equations. Then at the next step (3.18a) takes the form

$$\begin{bmatrix} \underline{z}_k \\ \hline z_{k+1} \end{bmatrix} = A_{k+1}\underline{x} = \begin{bmatrix} A_k \\ \hline \underline{a}^T \end{bmatrix} \underline{x} \tag{3.18b}$$

where z_{k+1} is a new scalar measurement and $\underline{a}^T$ is a row vector of coefficients relating the observation z_{k+1} to $\underline{x}$.

The solution to (3.18b) is then

$$\hat{\underline{x}}_{k+1} = P_{k+1} A_{k+1}^T \underline{z}_{k+1} . \tag{3.19b}$$

In order to find an alternate expression for (3.19b) let

$$\underline{x} = \hat{\underline{x}}_k + \underline{\Delta x} . \tag{3.20}$$

Substitute (3.20) into (3.18b) and premultiply with A_{k+1}^T to obtain

$$A_{k+1}^T \begin{bmatrix} \underline{z}_k \\ \hline z_{k+1} \end{bmatrix} = A_{k+1}^T A_{k+1} (\hat{\underline{x}}_k + \Delta\underline{x}),$$

or

$$A_k^T \underline{z}_k + z_{k+1}\underline{a} = [A_k^T A_k + \underline{a}\,\underline{a}^T](\hat{\underline{x}}_k + \Delta\underline{x}).$$

But

$$A_k^T A_k = P_k^+ .$$

Thus

$$A_k^T \underline{z}_k - P_k^+ \hat{\underline{x}}_k + (z_{k+1} - \underline{a}^T \hat{\underline{x}}_i)\underline{a} = [P_k^+ + \underline{a}\,\underline{a}^T]\Delta\underline{x} \qquad (3.21a)$$

The term $A_k^T \underline{z}_k - P_k^+ \hat{\underline{x}}_k$ can be shown to equal the null vector as follows. According to the defining equations of the pseudo inverse, (3.5),

$$\begin{aligned} A_k^T &= A_k^T A_k^{T+} A_k^T \\ &= [A_k^+ A_k]^T A_k^T \\ &= A_k^+ A_k A_k^T \\ &= A_k^+ [A_k A_k^+] A_k A_k^T \\ &= A_k^+ A_k^{+T} A_k^T A_k A_k^T \\ &= [A_k^T A_k]^+ [A_k^T A_k] A_k^T = P_k P_k^+ A_k^T \end{aligned} \qquad (3.22)$$

Also since $P_k = P_k^T$ and $P_k^+ = [P_k^+]^T$

$$\begin{aligned} P_k P_k^+ &= [P_k P_k^+]^T \\ &= P_k^{+T} P_k^T \\ &= P_k^+ P_k \end{aligned} \qquad (3.23)$$

Then using (3.22) and (3.23),

$$A_k^T \underline{z}_k - P_k^+ \hat{\underline{x}}_k = A_k^T \underline{z}_k - P_k^+ P_k A_k^T \underline{z}_k$$

$$= [I - P_k^+ P_k] A_k^T z_k$$

$$= [I - P_k P_k^+] A_k^T z_k$$

$$= [I - P_k P_k^+] P_k P_k^+ A_k^T z_k = \underline{0} \ . \qquad (3.24)$$

Then Eq. (3.21a) reduces to

$$(z_{k+1} - \underline{a}^T \hat{\underline{x}}_k)\ \underline{a} = [P_k^+ + \underline{a}\,\underline{a}^T]\ \Delta\underline{x}. \qquad (3.21b)$$

The solution of (3.21b) for $\Delta\underline{x}$ is obtained as follows.

(1) If P_k has rank $r < n$ and $[I - P_k^+ P_k]\ \underline{a} \neq \underline{0}$ the solution

$$\Delta\underline{x}^{(1)} = \frac{[I - P_k^+ P_k]\ \underline{a}}{\underline{a}^T [I - P_k^+ P_k] \underline{a}} (z_{k+1} - \underline{a}^T \underline{x}_k) \qquad (3.25)$$

satisfies (2.21b) by inspection. $\Delta\underline{x}^{(1)}$ is not defined if either P_k is of rank n, which implies that $[I - P_k^+ P_k] = 0$, or if $[I - P_k^+ P_k]\ \underline{a} = 0$.

(2) If P_k has rank $r = n$, or if $[I - P_k^+ P_k]\underline{a} = \underline{0}$ the solution is given by

$$\Delta\underline{x}^{(2)} = \frac{P_k \underline{a}}{1 + \underline{a}^T P_k \underline{a}} (z_{k+1} - \underline{a}^T \hat{\underline{x}}_k) \qquad (3.26)$$

Substitution of (3.26) into (3.21b) yields

$$[P_k^+ + \underline{a}\,\underline{a}^T]\ \Delta\hat{\underline{x}}^{(2)} = \frac{z_{k+1} - \underline{a}^T \hat{\underline{x}}_k}{1 + \underline{a}^T P_k \underline{a}} \{\underline{a} - (\underline{a}^T P_k \underline{a})\underline{a} - [I - P_k^+ P_k]\underline{a}\} \qquad (3.27)$$

Since either $[I-P_k^+P_k] = 0$ when P_k has rank $r = n$, or $[I-P_k^+P_k]\underline{a} = 0$ (3.27) reduces to (3.21b). Using the above two possible solutions for $\Delta\underline{x}$, which according to the conditions (1) and (2) are mutually exclusive, recursive forms for the solution of (3.18a) are established.

(1) Recursive relation if P_k has rank $r = n$, or if $[I-P_k^+P_k]\underline{a} = \underline{0}$

From (3.20) and (3.26),

$$\hat{\underline{x}}_{k+1} = \hat{\underline{x}}_k + \frac{P_k\underline{a}}{1+\underline{a}^T P_k\underline{a}} (z_{k+1} - \underline{a}^T\hat{\underline{x}}_k) \qquad (3.28a)$$

Combining (3.28a) and (3.19b) determines the updating procedure for the matrix P_{k+1}:

$$\hat{\underline{x}}_{k+1} = P_k A_k^T \underline{z}_k - \frac{P_k\underline{a}\underline{a}^T}{1+\underline{a}^T P_k\underline{a}} P_k A_k^T \underline{z}_k + z_{k+1} \frac{P_k\underline{a}}{1+\underline{a}^T P_k\underline{a}}$$

$$= [P_k - \frac{P_k\underline{a}\underline{a}^T P_k}{1+\underline{a}^T P_k\underline{a}}] A_k^T\underline{z}_k + z_{k+1} \frac{P_k\underline{a}}{1+\underline{a}^T P_k\underline{a}} .$$

Since

$$[P_k - \frac{P_k\underline{a}\underline{a}^T P_k}{1+\underline{a}^T P_k\underline{a}}] \underline{a} = \frac{P_k\underline{a}}{1+\underline{a}^T P_k\underline{a}}$$

$$\hat{\underline{x}}_{k+1} = [P_k - \frac{P_k\underline{a}\underline{a}^T P_k}{1+\underline{a}^T P_k\underline{a}}] [A^T\underline{z}_k + z_{k+1}\underline{a}] .$$

Also

$$\hat{\underline{x}}_{k+1} = P_{k+1} [A_k^T\underline{z}_k + z_{k+1}\underline{a}] .$$

Thus,

$$P_{k+1} = P_k - \frac{P_k \underline{a}\,\underline{a}^T P_k}{1+\underline{a}^T P_k \underline{a}} \quad . \tag{3.28b}$$

The new matrix satisfies the defining equations for the pseudo inverse (3.5). Thus (3.28a) and(3.28b) constitute the required recursive form

$$\left.\begin{aligned} \hat{\underline{x}}_{k+1} &= \hat{\underline{x}}_k + \frac{P_k \underline{a}}{1+\underline{a}^T P_k \underline{a}} (z_{k+1} - \underline{a}^T \hat{\underline{x}}_k) \\ P_{k+1} &= P_k - \frac{P_k \underline{a}\,\underline{a}^T P_k}{1+\underline{a}^T P_k \underline{a}} \end{aligned}\right\} \tag{3.28}$$

(2) Recursive relation if P_k has rank $r<n$ and $[I-P_k^+ P_k]\,\underline{a} \neq \underline{0}$. This condition excludes the solution (3.26), which does not satisfy (3.21b). Then from (3.20) and (3.25)

$$\hat{\underline{x}}_{k+1} = \hat{\underline{x}}_k + \frac{[I-P_k^+ P_k]\underline{a}}{\underline{a}^T [I-P_k^+ P_k]\underline{a}} (z_{k+1} - \underline{a}^T \hat{\underline{x}}_k) \tag{3.29a}$$

For notational convenience define

$$\underline{g}_{k+1} = \frac{[I-P_k^+ P_k]\underline{a}}{\underline{a}^T [I-P_k^+ P_k]\underline{a}} \tag{3.30}$$

and note that $\underline{g}_{k+1}$ has the following properties

$$\left.\begin{aligned} \underline{a}^T \underline{g}_{k+1} &= \underline{g}_{k+1}^T \underline{a} = 1 \\ P_k \underline{g}_{k+1} &= P_k^+ \underline{g}_{k+1} = \underline{0} \\ \underline{g}_{k+1}^T P_k &= \underline{g}_{k+1}^T P_k^+ = \underline{0}^T \end{aligned}\right\} \tag{3.31}$$

The desired updating procedure for P_{k+1} is found by combining (3.20) and (3.25)

$$\hat{\underline{x}}_{k+1} = \hat{\underline{x}}_k + \underline{g}_{k+1}\,(z_{k+1} - \underline{a}^T\hat{\underline{x}}_k)$$

$$= P_k A_k^T \underline{z}_k - \underline{g}_{k+1}\underline{a}^T P_k A_k^T \underline{z}_k + z_{k+1}\underline{g}_{k+1}$$

$$= [P_k - \underline{g}_{k+1}\underline{a}^T P_k] A_k^T \underline{z}_k + z_{k+1}\underline{g}_{k+1}$$

Also,

$$\hat{\underline{x}}_{k+1} = [P_{k+1}] A^T \underline{z}_k + z_{k+1} P_{k+1}\underline{a}$$

Then P_{k+1} must satisfy the following conditions

(a) $P_{k+1} = P_{k+1}^T$ (3.32a)

since $A_{k+1}^T A_{k+1}$ is a symetric matrix.

(b) $P_{k+1}A_k^T = [P_k - \underline{g}_{k+1}\underline{a}^T P_k]A_k^T$ (3.32b)

(c) $P_{k+1}\,\underline{a} = \underline{g}_{k+1}$ (3.32c)

A possible solution satisfying the above conditions is

$$P_{k+1} = P_k - \underline{g}_{k+1}\underline{a}^T P_k - P_k\underline{a}\underline{g}_{k+1}^T + (1+\underline{a}^T P_k\underline{a})\underline{g}_{k+1}\underline{g}_{k+1}^T$$

(3.29b)

Assuming symmetry of P_k, Eq. (3.29b) satisfies (3.32a) by inspection. Using (3.22) and (3.31), Eq. (3.29b) can be shown to satisfy (3.32b):

$$P_{k+1}A_k^T = [P_k - \underline{g}_{k+1}\underline{a}^T P_k]A_k^T - P_k\underline{a}\,\underline{g}_{k+1}^T A_k^T + (1+\underline{a}^T P_k\underline{a})\underline{g}_{k+1}\underline{g}_{k+1}^T A_k^T$$

Since

$$\underline{g}_{k+1}^T A_k^T = \underline{g}_{k+1}^T P_k P_k^+ + A_k^T$$

and

$$\underline{g}_{k+1}^T P_k = \underline{0}^T,$$

then

$$P_{k+1} A_k^T = [I - \underline{g}_{k+1} \underline{a}^T P_k] A_k^T$$

The last condition (3.32c) is also satisfied by (3.29b) since

$$P_{k+1}\underline{a} = P_k\underline{a} - \underline{g}_{k+1}\underline{a}^T P_k\underline{a} - P_k\underline{a}\,\underline{g}_{k+1}^T\underline{a} + (1+\underline{a}^T P_k\underline{a})\underline{g}_{k+1}\underline{g}_{k+1}^T\underline{a}$$

$$= P_k\underline{a} - (\underline{a}^T P_k\underline{a})\underline{g}_{k+1} - P_k\underline{a} + (1+\underline{a}^T P_k\underline{a})\underline{g}_{k+1}$$

Hence

$$P_{k+1}\underline{a} = \underline{g}_{k+1}$$

In order to prove that (3.29b) is indeed the correct and unique expression for the pseudo inverse $P_{k+1}=[A_{k+1}^T A_{k+1}]^+$, P_{k+1} has to satisfy the defining equations for the pseudo inverse, (3.5). Proof that the equations in definition 2 are satisfied follows:

(1) Proof Using (3.31) and (3.23),

$$P_{k+1}P_{k+1}^+ = [P_k - \underline{g}_{k+1}\underline{a}^T P_k - P_k\underline{a}\,\underline{g}_{k+1}^T + (1+\underline{a}^T P_k\underline{a})\underline{g}_{k+1}\underline{g}_{k+1}^T][P_k^+ + \underline{a}\,\underline{a}^T]$$

$$= P_k P_k^+ - \underline{g}_{k+1}\underline{a}^T P_k P_k^+ + P_k\underline{a}\,\underline{a}^T - \underline{g}_{k+1}(\underline{a}^T P_k\underline{a})\underline{a}^T$$

$$- P_k\underline{a}\,\underline{g}_{k+1}^T\underline{a}\,\underline{a}^T + (1+\underline{a}^T P_k\underline{a})\underline{g}_{k+1}\underline{g}_{k+1}^T\underline{a}\,\underline{a}^T$$

$$= P_k P_k^+ - \underline{g}_{k+1}\underline{a}^T P_k P_k^+ + \underline{g}_{k+1}\underline{a}^T \qquad (3.32a)$$

$$P_{k+1}P_{k+1}^{+} = P_kP_k^{+} + \frac{[I-P_kP_k^{+}]\underline{a}\,\underline{a}^T[I-P_kP_k^{+}]}{\underline{a}^T[I-P_kP_k^T]\underline{a}} \qquad (3.32b)$$

Thus $P_{k+1}P_{k+1}^{+}$ is symmetric and

$$P_{k+1}P_{k+1}^{+} = [P_{k+1}P_{k+1}^{+}]^T .$$

(2) $$P_{k+1}^{+}P_{k+1} = [P_{k+1}^{+}P_{k+1}]^T$$

This follows since P_{k+1}, P_{k+1}^{+}, $P_{k+1}P_{k+1}^{+}$ are symmetric.

(3) Using (3.32b)

$$P_{k+1}^{+}P_{k+1}P_{k+1}^{+} = P_{k+1}^{+}[P_{k+1}P_{k+1}^{+}]$$

$$= [P_k^{+}+\underline{a}\,\underline{a}^T]\left\{P_kP_k^{+} + \frac{[I-P_kP_k^{+}]\underline{a}\,\underline{a}^T[I-P_kP_k^{+}]}{\underline{a}^T[I-P_kP_k^{+}]\,\underline{a}}\right\}$$

$$= P_k^{+}P_kP_k^{+} + \underline{a}\,\underline{a}^TP_kP_k^{+} + \underline{a}\,\underline{a}^T[I-P_kP_k^{+}]$$

$$= P_k^{+} + \underline{a}\,\underline{a}^T$$

$$P_{k+1}^{+}P_{k+1}P_{k+1}^{+} = P_{k+1}^{+} .$$

(4) Using (3.32a)

$$P_{k+1}P_{k+1}^{+}P_{k+1} = [P_{k+1}P_{k+1}^{+}]P_{k+1}$$

$$= \left\{P_kP_k^{+} + \underline{g}_{k+1}\underline{a}^T[I-P_kP_k^{+}]\right\}[P_k-\underline{g}_{k+1}\underline{a}^TP_k-P_k\underline{a}\,\underline{g}_{k+1}^T$$

$$+ (1+\underline{a}^TP_k\underline{a})\underline{g}_{k+1}\underline{g}_{k+1}^T]$$

$$=P_kP_k^{+}P_k - \underline{g}_{k+1}\underline{a}^T\underline{g}_{k+1}\underline{a}^TP_k - P_kP_k^{+}P_k\underline{a}\,\underline{g}_{k+1}^T + (1+\underline{a}^TP_k\underline{a})\underline{g}_{k+1}\underline{a}^T\underline{g}_{k+1}\underline{g}_{k+1}^T$$

$$= P_k - \underline{g}_{k+1}\underline{a}^T P_k - P_k \underline{a}\, \underline{g}_{k+1}^T + (1+\underline{a}^T P_k \underline{a})\underline{g}_{k+1}\underline{g}_{k+1}^T$$

$$P_{k+1}P_{k+1}^+P_{k+1} = P_{k+1} \cdot$$

This concludes the proof and (3.29b) is indeed the desired updating procedure for the matrix P_k.

To complete the recursive algorithm for the solution of (3.18) a recursive form for the matrix $R_k = [I-P_kP_k^+]$ has to be established. From (3.32a)

$$P_{k+1}P_{K+1}^+ = P_kP_k^+ + \underline{g}_{k+1}\underline{a}^T\,[I-P_kP_k^+]$$

then

$$[I-P_{k+1}P_{k+1}^+] = [I-P_kP_k^+] - \underline{g}_{k+1}\underline{a}^T\,[I-P_kP_k^+]$$

and

$$R_{k+1} = R_k - \underline{g}_{k+1}\,\underline{a}^T R_k \qquad (3.33)$$

Note that the matrix R_k remains unchanged if the recursive form (3.28) is applicable because, in this case,

$$P_{k+1}P_{k+1}^+ = \left[P_k - \frac{P_k\underline{a}\,\underline{a}^TP_k}{1+\underline{a}^TP_k\underline{a}}\right]\left[P_k^+ + \underline{a}\,\underline{a}^T\right]$$

$$= P_kP_k^+ + P_k\underline{a}\,\underline{a}^T - \frac{P_k\underline{a}\,\underline{a}^TP_k}{1+\underline{a}^TP_k\underline{a}}\,P_k^+ - \frac{\underline{a}^TP_k\underline{a}}{1+\underline{a}^TP_k\underline{a}}\,P_k\underline{a}\,\underline{a}^T$$

$$P_{k+1}P_{k+1}^+ = P_kP_k^+ + \frac{P_k\underline{a}\,\underline{a}^T}{1+\underline{a}^TP_k\underline{a}}\,[I-P_kP_k^+]\,.$$

Since either

$$[I-P_kP_k^+] = 0 \qquad \text{or} \qquad \underline{a}^T[I-P_kP_k^+] = \underline{0}^T,$$

then

$$P_{k+1}P_{k+1}^{+} = P_kP_k^{+}$$

and

$$R_{k+1} = R_k \ .$$

Then the complete recursive algorithm, if P_k has rank < n and $[I-P_kP_k^{+}]\underline{a} \neq \underline{0}$, is given by (3.29a), (3.30), (3.29b), and (3.33), which are summarized as

$$\left\{\begin{array}{ll} \underline{g}_{k+1} = \dfrac{R_k\underline{a}}{\underline{a}^TR_k\underline{a}} & \text{(a)} \\ \hat{\underline{x}}_{k+1} = \hat{\underline{x}}_k + \underline{g}_{k+1}(z_{k+1} - \underline{a}^T\underline{x}_k) & \text{(b)} \\ P_{k+1} = P_k - \underline{g}_{k+1}\underline{a}^TP_k - P_k\underline{a}\underline{g}_{k+1}^T + (1+\underline{a}^TP_k\underline{a})\underline{g}_{k+1}\underline{g}_{k+1}^T & \text{(c)} \\ R_{k+1} = R_k - \underline{g}_{k+1}\underline{a}^TR_k & \text{(d)} \end{array}\right\} \quad (3.34)$$

where $R_k = [I-P_kP_k^{+}]$ is an idempotent matrix, the trace of which is equal to n-r, where r is the rank of A_k and n the maximum possible rank of A_k. A computation flow diagram for the recursive algorithm (3.28) and (3.34) is given in Fig. 3.2. Note that if the normalized least-square-error solution is desired, only Eq. (3.24c) has to be changed to

$$P_{k+1} = P_k - \underline{g}_{k+1}\underline{a}^TP_k - P_k\underline{a}\,\underline{g}_{k+1}^T + (\underline{a}^T\underline{a} + \underline{a}^TP_k\underline{a})\underline{g}_{k+1}\underline{g}_{k+1}^T \quad (3.34e)$$

The complete algorithm, Eqs. (3.28) and (3.34), is illustrated in the following simple example.

LEAST SQUARES FIT RECURSIVE ALGORITHM

COMPUTATION DIAGRAM

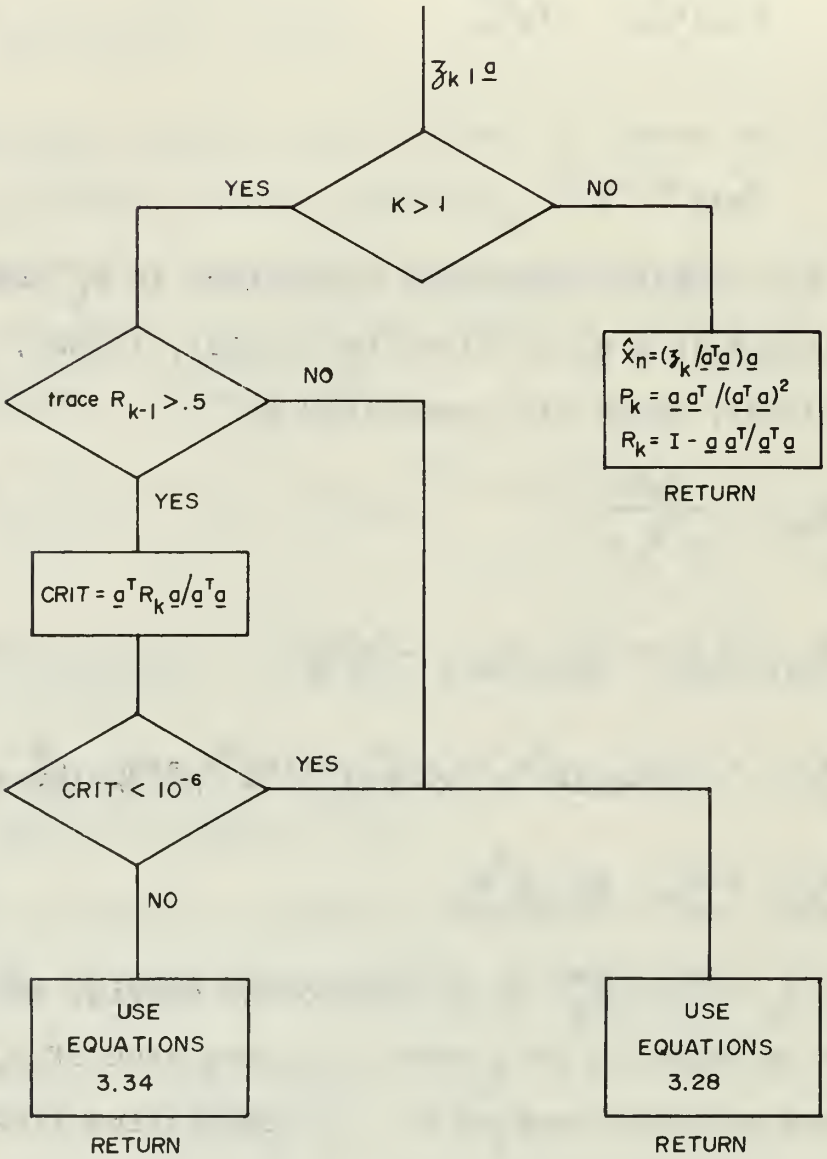

FIG. 3.2

Example 3.1 Given the four equations below, calculate the best approximate estimate for $\underline{x}$ sequentially.

$$1 = [1 \quad 0]\underline{x}$$

$$2 = [3 \quad 0]\underline{x}$$

$$3 = [1 \quad 1]\underline{x}$$

$$4 = [2 \quad 1]\underline{x}$$

Following the computation diagram in Fig. 3.2 the results below are obtained

a) $k = 1 \qquad z_1 = 1 \qquad a^T = [1 \quad 0]$

$$\hat{\underline{x}}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} ; \quad P_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} ; \quad R_1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

trace $R_1 = 1$

b) $k = 2 \qquad z_2 = 2 \qquad \underline{a}^T = [3 \quad 0]$

trace $R_1 = 1$, CRIT = 0, then use (3.28)

$$\hat{\underline{x}}_2 = \hat{\underline{x}}_1 + \frac{P_1\underline{a}}{1+\underline{a}^T P \underline{a}} = \begin{bmatrix} 1.3 \\ 0 \end{bmatrix}$$

$$P_2 = P_1 - \frac{P_1\underline{a}\underline{a}^T P_1}{1+\underline{a}^T P_1 \underline{a}} = \begin{bmatrix} .1 & 0 \\ 0 & 0 \end{bmatrix}$$

$$R_2 = R_1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

trace $R_2 = 1$.

c) $k = 3 \qquad z_3 = 3 \qquad \underline{a}^T = [1 \quad 1]$

trace $R_1 = 1$, CRIT = .5, then use (3.34)

$$\underline{g}_3 = \frac{R_2\underline{a}}{\underline{a}^T R_2 \underline{a}_2} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\hat{\underline{x}}_3 = \hat{\underline{x}}_2 + \underline{g}_3 (z_3 - \underline{a}^T x_2) = \begin{bmatrix} 1.3 \\ 1.7 \end{bmatrix}$$

$$P_3 = P_2 - \underline{g}_3\underline{a}^T P_2 - P_2\underline{a}\underline{g}_3^T + (1+\underline{a}^T P_2 \underline{a})\underline{g}_3\underline{g}_3^T = \begin{bmatrix} .1 & -.1 \\ -.1 & 1.1 \end{bmatrix}$$

$$R_3 = R_2 - \underline{g}_3\underline{a}^T R_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

trace of $R_3 = 0$, thus all following equations are processed according to Eq. (3.28).

d) $k = 4 \qquad z_4 = 4 \qquad \underline{a}^T = [2 \quad 1]$

$$\hat{\underline{x}}_4 = \hat{\underline{x}}_3 + \frac{P_k\underline{a}}{1+\underline{a}^T P_k \underline{a}} (z_4 - \underline{a}^T \underline{x}_3) = \begin{bmatrix} 1.29 \\ 1.67 \end{bmatrix}$$

$$P_4 = P_3 - \frac{P_3\underline{a}\,\underline{a}^T P_3}{1+\underline{a}^T P_3 \underline{a}} = \begin{bmatrix} .0952 & -.104 \\ -.104 & .714 \end{bmatrix}$$

The vector $\hat{\underline{x}}_4$ is then the best approximate solution to the given set of equations.

The recursive algorithm presented here is more general than the one used in Chapter II, because it includes a starting procedure. Regardless of the rank of A the estimate

obtained is the best approximate solution according to definition 1.

C. ESTIMATING THE STATES OF A LINEAR DYNAMIC SYSTEM

An alternate interpretation for the above recursive algorithm is that of determining the constant state vector $\underline{x}$ for the following system from a series of noise-contaminated scalar measurements:

System: $\underline{x}_{k+1} = I\, x_k$

Measurement: $z_{k+1} = M_{k+1}\underline{x}_{k+1} + \nu_{k+1}$

where M_{k+1} is the time-varying observation matrix and ν_{k+1} is the measurement noise.

In order to be able to estimate the state vector for a dynamic system, where the transition matrix $\Phi_{k+1,k}$ is in general time-varying and not equal to the identity matrix, it is desirable to develop an algorithm similar to (3.28) and (3.34) for the following systems and the scalar measurements z

System: $\underline{x}_{k+1} = \Phi_{k+1,k}\underline{x}_k$ (3.35a)

Measurement: $z_{k+1} = M_{k+1}\underline{x}_{k+1} + \nu_{k+1}$ (3.35b)

For the case when the system of equations for the estimation of $\underline{x}_k$ is determined or overdetermined (P_k has rank n) Eqs. (3.28) are easily adapted to include the transition matrix [3]. Let the equation

$$\underline{z}_k = A_k \underline{x}_k + \nu_k \tag{3.36a}$$

be valid at time instant k, then the solution is

$$\hat{\underline{x}}_k = [A_k^T A_k]^{-1} A_k^T \underline{z}_k \tag{3.36b}$$

and

$$P_k = [A_k^T A_k]^{-1} \tag{3.36c}$$

Using the property of the transition matrix that

$$[\Phi_{k,k-1}]^{-1} = \Phi_{k-1,k} \ , \tag{3.37a}$$

Eqs. (3.36) at time instant k+1 take the following form

$$\underline{z}_k = A_k \Phi_{k,k-1} \underline{x}_{k+1} \tag{3.37b}$$

$$\begin{aligned} \hat{\underline{x}}_{k+1} &= [\Phi_{k,k+1}^T A_k^T A_k \Phi_{k,k+1}]^{-1} \Phi_{k,k+1}^T A_k^T \underline{x}_k \\ &= \Phi_{k+1,k} \hat{\underline{x}}_k \end{aligned} \tag{3.37c}$$

$$\begin{aligned} Q_{k+1} &= [\Phi_{k,k+1}^T A_k^T A_k \Phi_{k,k+1}]^{-1} \\ &= \Phi_{k+1,k} P_k \Phi_{k+1,k}^T \end{aligned} \tag{3.37d}$$

Thus whenever the matrix P_k has rank n, the valid recursive algorithm for the dynamic case, including the new observation z_{k+1}, is

$$\left.\begin{aligned} Q_{k+1} &= \Phi_{k+1,k} P_k \Phi^T_{k+1,k} \\ P_{k+1} &= Q_{k+1} - \frac{Q_{k+1} M^T_{k+1} M_{k+1} Q_{k+1}}{1+M_{k+1} Q_{k+1} M^T_{k+1}} \\ \hat{\underline{x}}_{k+1} &= \Phi_{k+1,k} \hat{\underline{x}}_k + \frac{Q_{k+1} M^T_{k+1}}{1+M_{k+1} Q_{k+1} M^T_{k+1}} \end{aligned}\right\} \qquad (3.38)$$

Consider now the case when $[A_k^T A_k]$ is singular so that the pseudo inverse

$$P_k = [A_k^T A_k]^+ \qquad (3.39)$$

should be used. At time instant k+1

$$Q_{k+1} = [\Phi^T_{k,k+1} P_k^+ \Phi_{k,k+1}]^+ \qquad (3.40)$$

A comparison of (3.39) and (3.40) can easily be made using (3.14a). Let

$$A_k = N \cdot M$$

then

$$A_k^+ = M^T [M A_k^T A_k M^T]^{-1} M A_k^T$$

Also define

$$B_k = A_k \Phi_{k,k+1}$$

then

$$B^+ = \Phi^T_{k,k+1} M^T [M \Phi_{k,k+1} A_k^T A_k \Phi^T_{k,k+1} M^T]^{-1} M \Phi_{k,k+1} A_k^T$$

Thus (3.39) and (3.40) may be written as

$$P_k = M^T [M P_k^+ M^T]^{-1} M$$

$$Q_{k+1} = \Phi_{k,k+1}^T M^T [M \Phi_{k,k+1} P_k^+ \Phi_{k,k+1}^T M^T]^{-1} M \Phi_{k,k+1}$$

This reveals that there is no simple relationship between the matrices $[A_k^T A_k]^+$ and $[\Phi_{k,k+1}^T A_k^T A_k \Phi_{k,k+1}]^+$. Thus (3.34) cannot easily be adapted to yield the best approximate estimate for the state vector at k+1. However an acceptable alternative for a starting procedure is to estimate $\underline{x}_1$ sequentially using (3.28) and (3.34) until P_k reaches maximum rank; then (3.38) may be used. The intermediate estimates when P_k has rank $r \leq n$ are given by

$$\hat{\underline{x}}_k = \Phi_{k,1} \hat{\underline{x}}_1 .$$

IV. FINITE ITERATION METHODS

In previous chapters methods for the solution of $\underline{A}\underline{x}=\underline{b}$ and for the pseudo inversion in recursive form have been presented for the solution of the sequential-estimation problem. In this chapter, finite iteration methods for the solution of a set of linear equations and for matrix pseudo inversion are presented. These methods are based upon an infinite sequential error-correcting scheme, proposed by J. Nagumo and A. Noda [10], combined with the Gram - Schmidt process [9]. The derived methods require only a finite number (equal to the rank of A) of iterations. This approach has also been considered by L. D. Pyle [23] and some of the results presented in this chapter are similar to his. For the proper use of Pyle's algorithm it is necessary to rearrange the given set of equations whenever the constant b_1 in the first equation, $\underline{a}_1^T\underline{x} = b_1$, is equal to zero. Since this may not always be convenient in practice, an alternate algorithm is presented in which the computation starts unconditionally. Section A presents the basic iteration procedure with geometric interpretation. In Section B(1) this method is combined with the Gram - Schmidt process resulting in a procedure for the solution of a consistent set of equations. These results are extended in Section B(2) to solve a set of inconsistent equations and the solution is shown to be identical to the best approximate solution according to Penrose. An alternate

method accomplishing the same result is then derived. In Section C, the foregoing methods are extended to solve for the matrix inverse, when it exists, and for the pseudo inverse. In Appendix A these results are applied to the iterative solution of a set of non-linear equations.

A. INFINITE ITERATION PROCEDURE

Consider the problem of solving the following consistent set of equations. The term consistent is used to denote the fact that the system of equations is assumed to have either an exact solution or a unique locus for the solution. Alternatively $\underline{b}$ is contained within the vector space spanned by the column vectors of A.

$$A\underline{x} = \underline{b} \tag{4.1}$$

Let $\underline{a}_i^T$ represent the i'th row of A and b_i the i'th element in b. Then (3.1) may be written as

$$\begin{aligned} \underline{a}_1^T\underline{x} &= b_1 \\ \underline{a}_2^T\underline{x} &= b_2 \\ &\vdots \\ \underline{a}_m^T\underline{x} &= b_m \end{aligned} \tag{4.2}$$

The iteration scheme proposed by J. Nagumo and A. Noda [10] solves each equation in (4.2) successively for $\underline{x}$ by adding to each approximation for $\underline{x}$ a correction of appropriate size in the direction normal to the hyper-plane in x-space represented by $\underline{a}_i^T\underline{x} = b_i$. After solving the m'th equation

the process starts over again with the first equation. Let the i'th approximation for $\underline{x}$ be denoted by $\underline{x}_i$ and the initial estimate for $\underline{x}$ by $\underline{x}_o = 0$. Then the method may be presented as

$$\left\{\begin{array}{l} \underline{x}_{1+jm} = \underline{x}_{0+jm} + (b_1 - \underline{a}_1^T \underline{x}_{0+jm}) \dfrac{\underline{a}_1}{\underline{a}_1^T \underline{a}_1} \\ \\ \underline{x}_{2+jm} = \underline{x}_{1+jm} + (b_2 - \underline{a}_2^T \underline{x}_{1+jm}) \dfrac{\underline{a}_2}{\underline{a}_2^T \underline{a}_2} \\ \\ \quad \vdots \\ \\ \underline{x}_{m+jm} = \underline{x}_{m-1+jm} + (b_m - \underline{a}_m^T \underline{x}_{m-1+jm}) \dfrac{\underline{a}_m}{\underline{a}_m^T \underline{a}_m} \end{array}\right\} \tag{4.3}$$

Eq. (4.3) may be derived as follows. Consider the i'th equation in (4.2) and assume that the solution has the form

$$\underline{x}_i = \underline{x}_{i-1} + \Delta \underline{x}_i \tag{4.4}$$

where $\underline{x}_{i-1}$ is the solution for $\underline{x}$ obtained from solving the (i-1)st equation. Combining the i'th equation of (4.2) with (4.4) yields

$$(b_i - g_i^T \underline{x}_{i-1}) = \underline{a}_i^T \Delta \underline{x}_i \tag{4.5}$$

which may be solved for $\Delta \underline{x}_i$ using the best approximate solution according to Penrose, namely

$$\Delta \underline{x}_i = (b_i - \underline{a}_i^T \underline{x}_{i-1}) \frac{\underline{a}_i}{\underline{a}_i^T \underline{a}_i} \quad . \tag{4.6}$$

Then $\Delta \underline{x}_i$ is the minimum norm solution of (4.5), and

$$\underline{x}_i = \underline{x}_{i-1} + (b_i - \underline{a}_i^T \underline{x}_{i-1}) \frac{\underline{a}_i}{\underline{a}_i^T \underline{a}_i} \qquad (4.7)$$

This describes the equations in (4.3) before the iteration process starts over again with the first equation in (4.2). The convergence rate of this iteration scheme, although quite rapid at first, decreases asymptotically towards zero as the solution is approached. The limit is the exact solution if the system is determined, that is, if the rank of A is equal to the number of unknown elements in $\underline{x}$. However, if the system is undetermined, the minimum-norm solution, as discussed in Chapter III, is approached since each correction $\underline{\Delta x}_i$ is in the direction normal to the plane described by the i'th equation in (4.2)

B. FINITE ITERATION PROCEDURE

1. Sets of Consistent Equations

The foregoing iteration scheme requires an infinite number of steps to converge to the solution. If the process is truncated, only an approximation is obtained. As will be demonstrated this difficulty may be remedied by constraining the corrections to be orthogonal to each other.

Again consider the set of consistent equations (4.2) where each vector $\underline{a}_i$ is normal to the hyper-plane described by the i'th equation in (4.2). These normal vectors are generally not orthogonal to each other. However, using the Gram-Schmidt process [9], an orthogonal basis for the vector

space spanned by the vectors $\underline{a}_i$ (i = 1,...,m) may be obtained. According to this procedure a sequence of vectors $\underline{\alpha}_i$ is constructed from the set of vectors $\underline{a}_i$ in the following manner:

$$\begin{aligned}
\underline{\alpha}_1 &= \underline{a}_1 \\
\underline{\alpha}_2 &= \underline{a}_2 - \frac{\underline{\alpha}_1^T \underline{a}_2}{\underline{\alpha}_1^T \underline{\alpha}_1} \underline{\alpha}_1 \\
&\vdots \\
\underline{\alpha}_k &= \underline{a}_k - \sum_{i=1}^{k-1} \frac{\underline{\alpha}_i^T \underline{a}_k}{\underline{\alpha}_i^T \underline{\alpha}_i} \underline{\alpha}_i
\end{aligned} \tag{4.8}$$

then the set of vectors $\underline{\alpha}_i$ consists of mutually orthogonal vectors only. Note that $\frac{\underline{\alpha}_i^T \underline{a}_k}{\underline{\alpha}_i^T \underline{\alpha}_i} \underline{\alpha}_i$ is the component of $\underline{a}_k$ in the direction of $\underline{\alpha}_i$, which is subtracted from $\underline{a}_k$ leaving the normal component to $\underline{\alpha}_i$. In recursive formulation this orthogonalization process may be presented as follows.

Let $\{C_o, C_1, \ldots, C_k\}$ be a sequence of n x n matrices with $C_o = I$, then the set of mutually orthogonal base vectors $\underline{\alpha}_k$ are obtained from

$$\underline{\alpha}_k = C_{k-1} \underline{a}_i \quad \text{if } \underline{\alpha}_k = \underline{0}, \text{ recalculate } \underline{\alpha}_k \text{ using } \underline{a}_{i+1}$$

$$C_k = C_{k-1} - \frac{\underline{\alpha}_k \underline{\alpha}_k^T}{\underline{\alpha}_k^T \underline{\alpha}_k} \qquad \begin{matrix} i = 1,2,\ldots,m \\ k = 1,2,\ldots,r \end{matrix} \tag{4.9}$$

If the process yields a zero base vector, the corresponding vector $\underline{a}_i$ is a linear combination of the previously defined base vectors, and the i'th equation is a linear combination of the previous equations. Therefore the i'th equation and

the corresponding zero base vector may be disregarded, so that, finally, there are only r independent equations and r orthogonal base vectors, where r is the rank of A in (4.1). The exact solution for $\underline{x}$ or the minimum-norm solution for $\underline{x}$, if the system of equations is underdetermined, is obtained as a linear combination of these r base vectors. Using the form of (4.7) a correction $\underline{\Delta x}_k$ is made successively for each of the r mutually orthogonal base vectors

$$\underline{x}_k = \underline{x}_{k-1} + (b_k - \underline{a}_k^T \underline{x}_{k-1}) \frac{\underline{\alpha}_k}{\underline{\alpha}_k^T \underline{\alpha}_k} \tag{4.10}$$

It should be noted that the process (4.10) terminates after the r corrections are made. Thus the infinite iteration process of (4.3) essentially reduces to an r-step process. These results are summarized in (4.11)

$$\left\{ \begin{array}{ll} \underline{\alpha}_k = C_{k-1}\underline{a}_i \quad \text{if } \underline{\alpha}_k = \underline{0}, \text{ recalculate } \underline{\alpha}_k \text{ using } \underline{a}_{ii} & \\ C_k = C_{k-1} - \dfrac{\underline{\alpha}_k \underline{\alpha}_k^T}{\underline{\alpha}_k^T \underline{\alpha}_k} & \\ \underline{x}_k = \underline{x}_{k-1} + \dfrac{\underline{\alpha}_k}{\underline{\alpha}_k^T \underline{\alpha}_k} (b_k - \underline{a}_i^T \underline{x}_{k-1}) & \begin{array}{l} i = 1,2,\ldots,m \\ k = 1,2,\ldots,r \end{array} \end{array} \right\} \tag{4.11}$$

Note however that the first equation which starts the process has to have a nonzero element, b_1, in the vector $\underline{b}$, because if $b_1 = 0$, $\underline{x}_1 = \underline{0}$ which is not correct generally. If $b_1 = 0$ the process may be started by either choosing another suitable equation of (4.2) as the first equation

or by initializing $\underline{x}_o = \underline{a}_j$, where $\underline{a}_j \times \underline{a}_i \neq 0$. That is, $\underline{a}_j$ is not parallel to $\underline{a}_1$. Eqs. (4.11) may be rewritten in a compact form as

$$\left\{ \begin{array}{l} \underline{x}_k = \underline{x}_{k-1} + \dfrac{P_{k-1}\underline{a}_j}{\underline{a}_j^T P_{k-1}\underline{a}_j}(b_j - \underline{a}_j \underline{x}_{k-1}) \\ \\ P_k = P_{k-1} - \dfrac{P_{k-1}\underline{a}_j \underline{a}_j^T P_{k-1}}{\underline{a}_j^T P_{k-1}\underline{a}_j} \qquad k = 1,2,\ldots,r \end{array} \right\} \quad (4.12)$$

where $P_o = I$ and the index j denotes the succeeding equation in (4.2) for which $\underline{a}_j^T P_k \underline{a}_j \neq 0$. It is interesting to compare the form (4.12) with the form of the recursive least-square-error solution (2.26). The following example illustrates the use of (4.11).

Example 4.1: Use Eqs. (4.11) to solve the following set of equations for $\underline{x}$.

$$\begin{bmatrix} 1 & 0 & 2 \\ 1 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\underline{x}_o = 0 \qquad C_o = I$$

Step 1: $\underline{\alpha}_1 = C_o \underline{a}_1 = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}$

$$\underline{x}_1 = \underline{x}_0 + (b_1 - \underline{a}_1^T \underline{x}_o) \frac{\underline{\alpha}_i}{\underline{\alpha}_i^T \underline{\alpha}_i} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} - \frac{1}{5} \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} .2 \\ 0 \\ .4 \end{bmatrix}$$

$$C_1 = C_o - \frac{\underline{\alpha}_i \underline{\alpha}_i^T}{\underline{\alpha}_i^T \underline{\alpha}_i} = \begin{bmatrix} .8 & 0 & -.4 \\ 0 & 1 & 0 \\ -.4 & 0 & .2 \end{bmatrix}$$

Step 2: $\underline{\alpha}_2 = C_1\underline{a}_2 = \begin{bmatrix} .8 \\ 1 \\ -.4 \end{bmatrix}$

$$\underline{x}_2 = \underline{x}_1 + (b_2 - \underline{a}_2^T\underline{x}_1)\frac{\underline{\alpha}_2}{\underline{\alpha}_2^T\underline{\alpha}_2} = \begin{bmatrix} .2 \\ 0 \\ .4 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} .8 \\ 1 \\ -.4 \end{bmatrix}\begin{bmatrix} 1 \\ -1 \\ 4 \end{bmatrix}$$

$$C_2 = C_1 - \frac{\underline{\alpha}_2\underline{\alpha}_2^T}{\underline{\alpha}_2^T\underline{\alpha}_2} = \frac{1}{9}\begin{bmatrix} 4 & -4 & -2 \\ -4 & 4 & 2 \\ -2 & 2 & 1 \end{bmatrix}$$

Step 3: $\underline{\alpha}_3 = C_2\underline{a}_3 = \begin{bmatrix} -10/9 \\ +10/9 \\ +\ 5/9 \end{bmatrix}$

$$\underline{x}_3 = \underline{x}_2 + (b_3 - \underline{a}_3^T\underline{x}_2)\frac{\underline{\alpha}_3}{\underline{\alpha}_3^T\underline{\alpha}_3} = \frac{1}{9}\begin{bmatrix} 1 \\ -1 \\ 4 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} -.8 \\ .8 \\ .4 \end{bmatrix} = \begin{bmatrix} .2 \\ -.2 \\ .4 \end{bmatrix}$$

$$C_3 = C_2 - \frac{\underline{\alpha}_3\underline{\alpha}_3^T}{\underline{\alpha}_3^T\underline{\alpha}_3} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$\underline{x}_3$ is the unique solution to the given set of equations. The iteration sequence as well as the planes described by the set of equations are shown in Fig. 4.1.

2. Sets of Inconsistent Equations

If the set of equations (4.1) is inconsistent, as is usually the case in estimation problems, the vector $\underline{b}$ is not completely contained in the vector space spanned by the column vectors of A. A solution may be obtained by solving the equation

$$A\underline{x} = \underline{b}_A \tag{4.13}$$

ILLUSTRATION FOR EXAMPLE 4.1

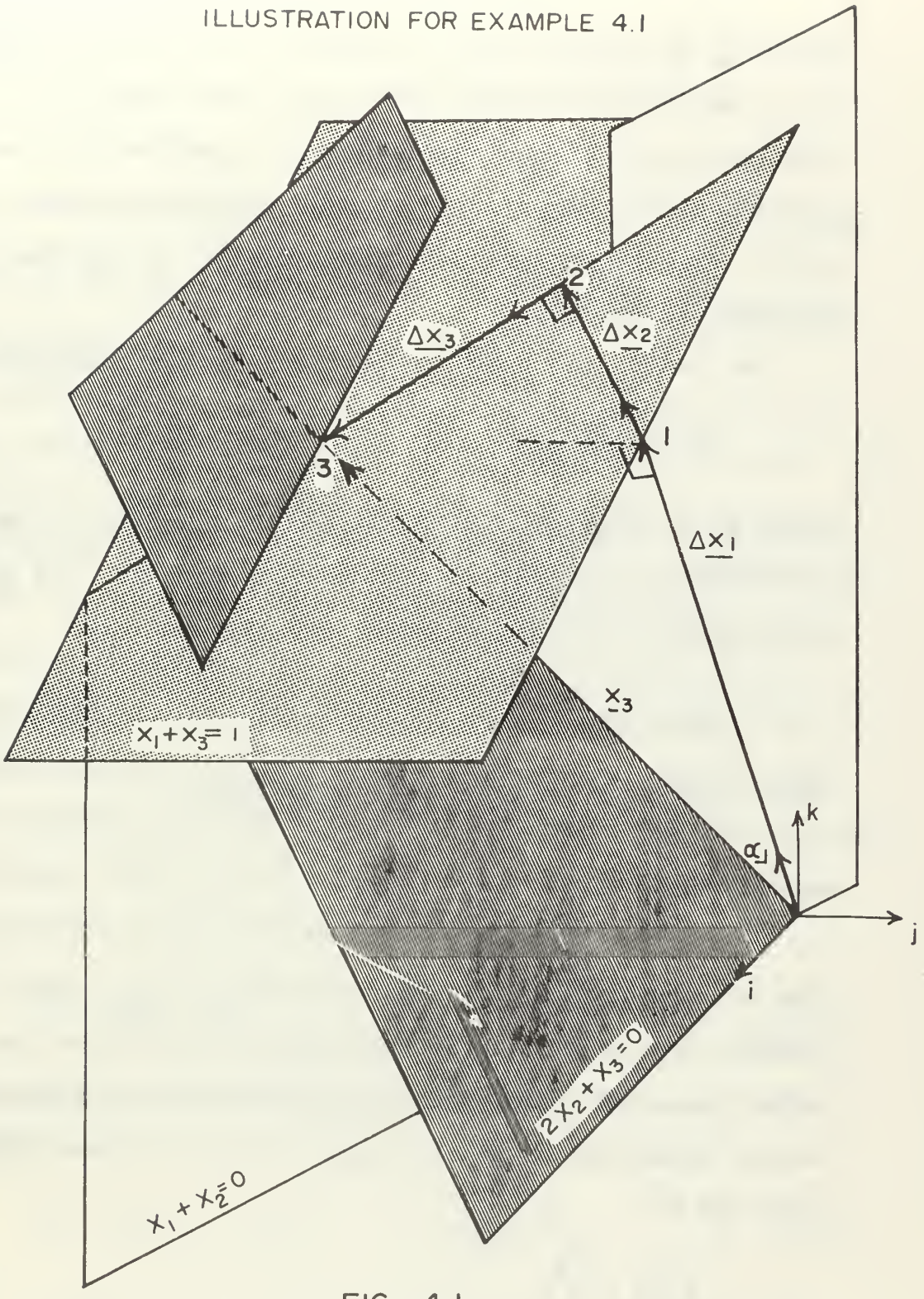

FIG. 4.1

where $\underline{b}_A$ is the part of $\underline{b}$ contained in the column space of A, or the projection of $\underline{b}$ into the column space of A. The remainder $\underline{b}_N = \underline{b} - \underline{b}_A$ represents that portion of $\underline{b}$ which is ignored in the solution, and is orthogonal to the space of A. This represents an optimal choice for $\underline{b}_A$ which may be shown as follows:

Let the vector $\underline{b}$ be decomposed into two components

$$\underline{b} = \underline{b}'_A + \underline{b}_E \tag{4.14}$$

where $\underline{b}'_A$ is contained in the space of A and $\underline{b}_E$ is the part of $\underline{b}$ which is ignored in the solution process. If $\underline{x}_k$ is the solution of the consistent set of equations

$$\underline{b}'_A = A\underline{x}_k$$

then $$\|\underline{b} - A\underline{x}_k\| = \|\underline{b}'_A - A\underline{x}_k + \underline{b}_E\| = \|\underline{b}_E\|$$

where $$\|\underline{b}_E\| = \sum_{i=1}^{m} (b_i - \underline{a}_i^T \underline{x}_k)^2$$

The minimum of $\|\underline{b}_E\|$ is obtained when $\underline{b}_E = \underline{b}_N$. Then the solution for $\underline{x}$ of the inconsistent set (4.13) is, by definition, the least-square-error solution. The standard way of achieving this projection of $\underline{b}$ is to premultiply (4.2) by A^T.

$$A^T A\underline{x} = A^T \underline{b}$$

$$= A^T \underline{b}_A + A^T \underline{b}_N$$

but since

$$A^T \underline{b}_N = \underline{0}$$

$$A^T A\underline{x} = A^T \underline{b}_A \tag{4.15}$$

Since (4.15) is a set of consistent equations, it may be solved using (4.11).

Another finite-step process for the solution of (4.1) which starts unconditionally but involves a little more computation may be derived following arguments similar to those which led to (4.11). Again consider the equations in (4.2). In order to obtain the minimum-norm solution (if the set is underdetermined) the desired solution may be represented, as in the previous section, as a linear combination of the row vectors of matrix A, or, equivalently, as a weighted sum of different base vectors representing the same space. The set of Eqs. (4.1), if it is assumed to be inconsistent, may be written

$$\underline{b} = A\underline{x} + \underline{b}_N \tag{4.16}$$

where again $\underline{b}_N$ is orthogonal to the space of A.
Now let

$$\underline{b} = \lambda_1 A\underline{\alpha}_1 + \lambda_2 A\underline{\alpha}_2 + \ldots + \lambda_r A\underline{a}_r + \underline{b}_N \tag{4.17}$$

where the $A\underline{\alpha}_k$ vectors are constructed to be mutually perpendicular and $\lambda_k A\underline{\alpha}_k$ is the part of $\underline{b}$ parallel to the vector $A\underline{\alpha}_k$. Thus the coefficients λ_k are determined from the condition

$$(A\underline{\alpha}_k)^T (\underline{b} - \underline{\alpha}_k A\underline{x}_k) = 0 \tag{4.18}$$

as

$$k = \frac{\underline{\alpha}_k^T A^T \underline{b}}{\underline{\alpha}_k^T A^T A \underline{\alpha}_k} \tag{4.19}$$

Substituting (4.19) into (4.17) yields

$$\underline{b} = \frac{\underline{\alpha}_1^T A^T \underline{b}}{\underline{\alpha}_1^T A^T A \underline{\alpha}_1} A\underline{\alpha}_1 + \ldots + \frac{\underline{\alpha}_r^T A^T \underline{b}}{\underline{\alpha}_r^T A^T A \underline{\alpha}_r} A\underline{\alpha}_r + \underline{b}_N$$

$$= A \left[\sum_{k=1}^{r} \frac{\underline{\alpha}_k^T A^T \underline{b}}{\underline{\alpha}_k^T A^T A \underline{\alpha}_k} \underline{\alpha}_k \right] + \underline{b}_N \tag{4.20}$$

Comparison with (4.16) yields the desired solution, namely,

$$\hat{\underline{x}} = \sum_{k=1}^{r} \frac{\underline{\alpha}_k^T A^T \underline{b}}{\underline{\alpha}_k A^T A \underline{\alpha}_k} \underline{\alpha}_k \ . \tag{4.21}$$

The mutually perpendicular vectors $A\underline{\alpha}_k$ are constructed using the Gram-Schmidt process. Using (4.9) with $\underline{\alpha}_k$ replaced by $A\underline{\alpha}_k$ and $\underline{a}_i$ replaced by $A\underline{a}_i$, yields

$$\left\{ \begin{array}{l} A\underline{\alpha}_k = C_{k-1} A\underline{a}_i \quad \text{if } A\underline{\alpha}_k = \underline{0}, \text{ recalculate } A\underline{\alpha}_k \text{ using } A\underline{a}_{i+1} \\ \\ C_k = C_{k-1} - \dfrac{A\underline{\alpha}_k \underline{\alpha}_k^T A^T}{\underline{\alpha}_k^T A^T A \underline{\alpha}_k} \qquad \begin{array}{l} i = 1,2,\ldots,m \\ k = 1,2,\ldots,r \end{array} \end{array} \right\} \tag{4.22}$$

Another equally acceptable set of mutually orthogonal base vectors for the column space of A may be obtained from

the vectors $A\underline{d}_i$, where the $\underline{d}_i$'s are the rows of the matrix A^TA, since the column space of A may be expressed using the set of all the vectors $A\underline{d}_i$, as well as the set of all vectors $A\underline{a}_i$. Using $A\underline{d}_i$ instead of $A\underline{a}_i$ in (4.22) requires less computation if the dimensions of A are such that $m \gg n$. Thus it is possible to write, starting with $C_o = 0$.

$$\left\{ \begin{array}{ll} A\underline{\beta}_k = C_{k-1}A\underline{d}_i & \text{if } A\underline{\beta}_k = \underline{0}, \text{ recalculate } A\underline{\beta}_k \text{ using } A\underline{d}_{i+1} \\ \\ C_k = C_{k-1} - \dfrac{A\underline{\beta}_k\underline{\beta}_k^TA^T}{\underline{\beta}_k^TA^TA\underline{\beta}_k} & \begin{array}{l} i = 1,2,\ldots,n \\ k = 1,2,\ldots,r \end{array} \end{array} \right\} \tag{4.23}$$

where $\underline{\beta}_k$ replaces $\underline{\alpha}_k$ in (4.17) and (4.21).
In order to obtain the solution (4.21), Eq. (4.23) is modified to yield an explicit form for the calculation of the $\underline{\beta}_k$'s, which when multiplied by A yield the orthogonal base vectors for the column space of A.

$$C_{k-1} = I - \frac{A\underline{\beta}_1\underline{\beta}_1^TA^T}{\underline{\beta}_1^TA^TA\underline{\beta}_1} - \ldots - \frac{A\underline{\beta}_{k-1}\underline{\beta}_{k-1}^TA^T}{\underline{\beta}_{k-1}^TA^TA\underline{\beta}_{k-1}}$$

$$= I - \sum_{i=1}^{k-1} \frac{A\underline{\beta}_i\underline{\beta}_i^TA^T}{\underline{\beta}_i^TA^TA\underline{\beta}_N}$$

then

$$A\underline{\beta}_k = \left[I - \sum_{i=1}^{k=1} \frac{A\underline{\beta}_i\underline{\beta}_i^TA^T}{\underline{\beta}_i^TA^TA\underline{\beta}_i}\right] A\underline{d}_i$$

$$= A\left[I - \sum_{i=1}^{k-1} \frac{\underline{\beta}_i\underline{\beta}_i^TA^TA}{\underline{\beta}_i^TA^TA\underline{\beta}_i}\right] \underline{d}_i$$

and

$$\underline{\beta}_k = \left[I - \sum_{i=1}^{k-1} \frac{\underline{\beta}_i \underline{\beta}_i^T A^T A}{\underline{\beta}^T A^T A \underline{\beta}_i} \right] \underline{d}_i$$

Combining (4.24), written in recursive form, with (4.21) yields the desired finite step process for the solution of (4.1). Let $C_o = I$ then

$$\left\{ \begin{array}{l} \underline{\beta}_k = C_{k-1}\underline{d}_i \quad \text{if } \underline{\beta}_k = 0, \text{ recalculate } \underline{\beta}_k \text{ using } \underline{d}_{i+1} \\ \\ C_k = C_{k-1} - \dfrac{\underline{\beta}_k \underline{\beta}_k^T A^T A}{\underline{\beta}_k^T A^T A \underline{\beta}_k} \\ \\ \hat{\underline{x}}_k = \hat{\underline{x}}_{k-1} + \dfrac{\underline{\beta}_k^T A^T \underline{b}}{\underline{\beta}_k^T A^T A \underline{\beta}_k} \underline{\beta}_k \end{array} \right\} \begin{array}{l} i = 1,2,\ldots,n \\ k = 1,2,\ldots,r \end{array} \tag{4.25}$$

The process is completed when $k = r$. $\hat{\underline{x}}_r$ is the desired solution.

C. MATRIX PSEUDO INVERSION

The foregoing computation methods may be extended to yield matrix inversion or pseudo inversion. However no computation time comparison with already existing methods has been made. Consider the solution of the matrix equation

$$AX = I \tag{4.26}$$

where the matrix A is square of dimensions n x n and non-singular. The inverse of A may be obtained as the solution of (4.26) from (4.12). Let $P_o = I$ and $X = A^T$ then

$$\left\{\begin{array}{l} X_k = X_{k-1} + \dfrac{P_{k-1}\underline{a}_k}{\underline{a}_k^T P_{k-1}\underline{a}_k}\,(\underline{i}_k^T - \underline{a}_k^T X_{k-1}) \\ \\ P_k = P_{k-1} + \dfrac{P_{k-1}\underline{a}_k\underline{a}_k^T P_{k-1}}{\underline{a}_k^T P_{k-1}\underline{a}_k} \qquad k = 1,2,\ldots,n \end{array}\right\} \tag{4.27}$$

where $\underline{i}_k^T$ is the i'th row of the identity matrix and the n'th approximation for X, X_n is the desired inverse. The initial condition $X_o = A^T$ is chosen to ensure that the starting conditions for (4.12) are satisfied, since $\underline{a}_2, \underline{a}_3, \ldots, \underline{a}_n$ are, by virtue of the nonsingularity of A, not parallel to $\underline{a}_1$.

The pseudo inverse may be obtained as follows. Let $C_o = I$ and $X_0 = 0$, then

$$\left\{\begin{array}{l} \underline{\delta}_k = C_{k-1}\underline{d}_i \quad \text{if } \underline{\delta}_k = \underline{0}\text{, recalculate } \underline{\delta}_k \text{ using } \underline{d}_{i+1} \\ \\ C_k = C_{k-1} - \dfrac{\underline{\delta}_k\underline{\delta}_k^T}{\underline{\delta}_k^T A^T A\underline{\delta}_k}\, A^T A \\ \\ X_k = X_{k-1} + \dfrac{\underline{\delta}_k\underline{\delta}_k^T}{\underline{\delta}_k^T A^T A\underline{\delta}_k}\, A^T \qquad \begin{array}{l} i = 1,2,\ldots,n \\ k = 1,2,\ldots,k \end{array} \end{array}\right\} \tag{4.28}$$

where $\underline{d}_i^T$ denotes the i'th row of $A^T A$, and X_r is the solu= tion for the pseudo inverse of A. Proof of (4.28) follows. From (4.28) it is evident that

$$\underline{\delta}_i^T A^T A\underline{\delta}_j = 0 \qquad \text{for } i \neq j \tag{4.29}$$

since the vectors $A\underline{\delta}_i$ (i = 1,2,...,r) are orthogonal. Also A^TA may be expressed as a matrix whose rows are linear combinations of the set of vectors $\underline{\delta}_i$. Thus

$$A^TA = [\underline{\delta}_1|\underline{\delta}_2|...|\underline{\delta}_r]\ M \tag{4.30}$$

The expression for C_r from (4.28) may be rewritten as

$$\begin{aligned} C_r &= \left[I - \sum_{i=1}^{r} \frac{\underline{\delta}_i \underline{\delta}_i^T A^T A}{\underline{\delta}_i^T A^T A \underline{\delta}_i} \right] \\ &= I - \left[\sum_{i=1}^{r} \frac{\underline{\delta}_i \underline{\delta}_i^T}{\underline{\delta}_i^T A^T A \underline{\delta}_i} \right] A^TA \end{aligned} \tag{4.31}$$

where the summation is identified as Y

$$Y = \sum_{i=1}^{r} \frac{\underline{\delta}_i \underline{\delta}_i^T}{\underline{\delta}_i^T A^T A \underline{\delta}_i} \tag{4.32}$$

so that

$$C_r = I - YA^TA \tag{4.33}$$

Using (4.29) it follows that

$$C_r\underline{\delta}_i = \underline{0} \tag{4.34}$$

and thus

$$C_rA^TA = 0 \tag{4.35}$$

and $$C_rY = 0 \tag{4.36a}$$

or $$Y = YA^TAY. \tag{4.36b}$$

Using (4.36b) to expand the product A^TAY yields

$$A^TAY = A^TAYA^TAY$$

or $$[A^TA - A^TAYA^TA]Y = 0. \tag{4.37a}$$

Since $Y \neq 0$ this yields

$$A^TA = A^TAY\ A^TA \tag{4.37b}$$

From (4.35)

$$A^TA - YA^TA\ A^TA = 0$$

and then also $[A^TA - YA^TA\ A^TA]Y = 0.$

This is combined with (4.37a) to yield

$$[A^TAY - YA^TA]\ A^TAY = 0. \tag{4.38a}$$

Since $A^TAY \neq 0$, then

$$A^TAY = YA^TA. \tag{4.38b}$$

Eqs. (4.36b), (4.37b) and (4.38b) by definition establish Y as the pseudo inverse of A^TA:

$$Y=[A^TA]^+ \tag{4.39}$$

Comparing (4.32) with the last equation in (4.28) when $k = r$ yields

$$X_r = YA^T = [A^TA]^+A^T = A^+ \tag{4.40}$$

Penrose [5] has also suggested a recursive method for computing the pseudo-inverse. Let

$$C_1 = I$$

then

$$C_{k+1} = I \cdot \frac{1}{k} \text{ trace } (C_k A^T A) - C_k A^T A \tag{4.41}$$

The product $C_{r+1} A^T A = 0$, where r is the rank of $A^T A$, then the pseudo inverse of $A^T A$ is

$$[A^T A]^+ = \frac{r}{\text{trace } C_r A^T A} C_r \tag{4.42}$$

The proof is given in Ref. 5. It should be noted that the Penrose method involves at least one matrix multiplication for each step, or approximately rn^3 operations, where n^3 operations are required to perform the multiplication of two n x n matrices, whereas the method of (4.28) requires approximately $5rn^2$ operations to obtain the same result, namely $[A^T A]^+$.

V. RECURSIVE ALGORITHM FOR THE SLIDING-WINDOW OBSERVER

Since the work of Luenberger [11], deterministic linear observation systems, called observers, have been recognized as practical alternatives to statistical optimum linear filters when efficient and fast real-time estimation of the system states is desired. Avoiding problems associated with the estimation of à priori statistics, the observer simply solves the estimation problem as a deterministic one and disregards statistical quantities. The simplest formulation for the observer is the sampled-data type which accepts the measurements or observations only at discrete points in time.

Consider the sampled-data system

$$\begin{aligned} \underline{x}_k &= \Phi_{k,k-1}\, \underline{x}_{k-1} \\ z_k &= M_k\, \underline{x}_k \end{aligned} \tag{5.1}$$

where $\underline{x}_k$ is the system state vector at time instant k, $\Phi_{k,k-1}$ is the general time-varying transition matrix from time (k-1)T to kT. M_k is the time-varying observation matrix of dimension 1 x n, and z_k is the scalar observation. As in other chapters, only the case of scalar observations is considered here in order to obtain results without time consuming matrix inversions. The observer for the system (5.1) is given as

$$\underline{\hat{x}}_k = \Phi_{k,k-1}\underline{\hat{x}}_{k-1} + \underline{g}_k(z_k - M_k\Phi_{k,k-1}\underline{\hat{x}}_{k-1}) \tag{5.2}$$

where $\underline{\hat{x}}_k$ is the estimate at time kT.

This observer equation may be rewritten as

$$\underline{\hat{x}}_k = F_k\underline{\hat{x}}_{k-1} + \underline{g}_k z_k \tag{5.3a}$$

where

$$F_k = [I - \underline{g}_k M_k]\Phi_{k,k-1} \tag{5.3b}$$

The matrix F_k is called the observer transition matrix. Bona [12] has shown that the eigenvalues of F_k, which are dependent on the choice of g_k, determine the performance of the observer in processing noise-contaminated observations. The time-varying gain, g_k, for the optimal filter is determined such that the trace of the error covariance matrix is minimized. For a specific time-invariant observable system [3], Bona [12] has demonstrated that constant gain observer with eigenvalues of approximately 0.5 approach the performance of the Kalman filter. As an observer design rule for time-invariant systems, he suggests the choice of the largest eigenvalue λ_L of F so that $(\lambda_L)^{\ell}$ is approximately zero, with the result that $(F)^{\ell}$ is approximately zero thereby limiting the memory of the observer to approximately the last ℓ observations. Because of the size limitations of the memory, this type of observer is also referred to as a sliding-window observer.

In the following work, stimulated by a list of unsolved observer problems [13], a recursive relationship for the sliding-window of minimal length for the general system of Eq. (5.1) is developed in Section A. In Section B these results are extended for the time-invariant system to yield the recursive form for a sliding-window of specified length. The approach to the solution is quite different from the one discussed by Bona, resulting in a new design rule for sliding-window observer of exact specified memory length. Finally, the results obtained are illustrated by an example.

A. THE MINIMUM-WINDOW OBSERVER

The minimum-window observer is the fastest linear observer possible in that it determines the states of a system from the necessary minimum number of observations. The eigenvalues of the observer transition matrix are all zero [12]. The desired recursive form is derived as follows. Consider system (5.1) and its observer equation (5.3). Suppose the system is of order n. At each instant of time k the state vector, $\underline{x}_k$, is determined from the last n observations. Thus

$$\begin{bmatrix} z_{k-n+1} \\ z_{k-n+2} \\ \vdots \\ z_{k-1} \\ z_k \end{bmatrix} = \begin{bmatrix} M_{k-n+1} & \Phi_{k-n+1,k} \\ M_{k-n+1} & \Phi_{k-n+2,k} \\ \vdots & \\ M_{k-1} & \Phi_{k-1,k} \\ M_k & \end{bmatrix} \underline{x}_k \qquad (5.4a)$$

or

$$\underline{z}_k = R(k)\ \underline{x}_k \tag{5.4b}$$

where the matrix R(k) relates the n last observations $\underline{z}_k$ to the state vector $\underline{x}_k$. The observability condition for this system is then that R(k) is nonsingular for all k. Because of this it is not necessary that each of the rows of R(k) be the product of an observable pair. The solution to (5.4b) is then trivial

$$\underline{x}_k = R(k)^{-1}\underline{z}_k \tag{5.5}$$

where $\underline{x}_k$ denotes the output of the observer. Let the matrix $R^{-1}(k) = C(k)$ be partitioned into column vectors

$$C(k) = \left[\underline{c}_1(k) \mid \underline{c}_2(k) \mid \ldots \mid \underline{c}_n(k)\right] . \tag{5.6}$$

Then (5.5) may be written as

$$\hat{\underline{x}}_k = \underline{c}_1(k)z_{k-n+1} + \underline{c}_2(k)z_{k-n+2} + \ldots + \underline{c}_{n-1}(k)z_{k-1} + \underline{c}_n(k)z_k \tag{5.7}$$

Expanding Eq. (5.3) yields

$$\begin{aligned}
\hat{\underline{x}}_k &= F_k\hat{\underline{x}}_{k-1} + \underline{g}_k z_k \\
&= F_k F_{k-1}\underline{x}_{k-2} + F_k\underline{g}_{k-1}z_{k-1} + \underline{g}_k z_k \\
&= F_k F_{k-1}\cdots F_{k-n+1}\underline{x}_{k-n+1} + F_k F_{k-1}\cdots F_{k-n+2}\underline{g}_{k-n+1}z_{k-n+1} \\
&\quad + F_k F_{k-1}\cdots F_{k-n+3}\underline{g}_{k-n+2}\cdot z_{k-n+2} \\
&\quad \vdots \\
&\quad + F_k\underline{g}_{k-1}z_{k-1} \\
&\quad + \underline{g}_k z_k
\end{aligned} \tag{5.8}$$

Comparing (5.7) and (5.8) on a term by term basis leads to the following conclusions

$$(1)\quad F_k F_{k-1} \cdots F_{k-n+1} = 0 \tag{5.9a}$$

$$(2)\quad \underline{g}_n = \underline{c}_n(k) \tag{5.9b}$$

$$(3)\quad \underline{c}_j(k) = F_k \underline{c}_{j+1}(k-1) \qquad j = 1,2,\ldots,n-1 \tag{5.9c}$$

so that

$$C_{(k)} = \left[F_k \underline{c}_2(k-1) \,\middle|\, \cdots \,\middle|\, F_k \underline{c}_n(k-1) \,\middle|\, C_n(k) \right] . \tag{5.9d}$$

The recursive relation for the gain vector $\underline{g}_k = \underline{c}_n(k)$ may be determined from (5.9a) using (5.9c) repeatedly. Thus

$$F_k [F_{k-1} \cdots F_{k-n+1} \underline{g}_{k-n}] = F_k \, \underline{c}_1(k-1) = 0 \tag{5.10}$$

since

$$F_k = [I - \underline{g}_k M_k] \, \Phi_{k-1,k}$$

(5.10) may be rewritten as

$$\Phi_{k-1,k} \underline{c}_1(k-1) = \underline{g}_k M_k \, \Phi_{k-1,k} \, \underline{c}_1(k-1) \tag{5.11}$$

The desired result is then obtained directly from (5.11)

$$\underline{g}_k = \underline{c}_n(k) = \Phi_{k-1,k} \, \underline{c}_1(k-1) \, [M_k \, \Phi_{k-1,k} \, \underline{c}_1(k-1)]^{-1} \tag{5.12a}$$

Note that for scalar observation Eq. (5.12) may be also written as

$$\underline{g}_k = \frac{\Phi_{k-1,k} \underline{c}_1(k-1)}{M_k \Phi_{k-1,k} \underline{c}_1(k-1)} \tag{5.12b}$$

For scalar observation the desired algorithm is therefore given by Eqs. (5.12b), (5.9b), (5.9c) and (5.3). This is summarized in (5.13):

$$\left\{\begin{array}{l} \underline{g}_k = \dfrac{\Phi_{k-1,k}\underline{c}_1(k-1)}{M_k\Phi_{k-1,k}\underline{c}_1(k-1)} \\ \\ F_k = [I - \underline{g}_k M_k]\ \Phi_{k-1,k} \\ \\ \underline{c}_j(k) = F_k\underline{c}_{j+1}(k-1),\ j = 1,2,\ldots,n-1;\ \underline{c}_n(k)=\underline{g}_k \\ \\ \hat{\underline{x}}_k = F_k\ \hat{\underline{x}}_{k-1} + \underline{g}_k\ z_k \end{array}\right\} \tag{5.13}$$

This completes the derivation of the recursive algorithm for the minimum-window observer. The computation time required is almost equal to the computation time required for the corresponding optimal filter (3.38). For linear time-invariant systems with a time-invariant observation matrix, the observer gain and transition matrix are constant. Thus (5.13) reduces to the observer equation

$$\hat{\underline{x}}_k = F\hat{\underline{x}}_{k-1} + \underline{g}z_k \tag{5.14}$$

and the required computation time is reduced substantially. Although the minimum-window observer is highly sensitive to measurement noise, practically excluding the direct use of (5.14) for non-deterministic problems, it is nevertheless possible to construct acceptable filter schemes from (5.13) or (5.14) for special applications. Consider, for example,

an oscillatory time-invariant system given by Eq. (5.1) with zero damping. Then a simple averager following the minimum-window observer is characterized by

$$\hat{\underline{x}}_k = F\hat{\underline{x}}_{k-1} + \underline{g}z_k$$

$$\hat{\hat{\underline{x}}}_k = \frac{1}{k}\sum_{i=1}^{k} \Phi^{k-i}\hat{\underline{x}}_i = \frac{k-1}{k}\Phi\hat{\hat{\underline{x}}}_{k-1} + \frac{1}{k}\hat{\underline{x}}_k \qquad (5.15)$$

where $\hat{\hat{\underline{x}}}_k$, the output of the combined filter, results in a filter performance almost indistinguishable from the optimum filter (3.38). This is demonstrated in Example 5.1. Note that the computational requirements are only a fraction of the computational requirements for the optimal filter. It is the author's opinion that this result merits further investigation in order to find some design rules for simple and fast observers with almost optimal performance similar to the one given in Eq. (5.15).

Example 5.1

Let the system equations (5.1) be given as

$$\left.\begin{aligned} \underline{x}_k &= \Phi\underline{x}_{k-1} = \begin{bmatrix} .9 & -.5 \\ .38 & .9 \end{bmatrix}\underline{x}_{k-1} \\ z_k &= M_k\underline{x}_k + \nu_k = [1 \quad 0]\,\underline{x}_k + \nu_k \end{aligned}\right\} \qquad (5.16)$$

where the observations are now noise-contaminated with the measurement noise ν_k, a noise sample drawn from a uniform distributed noise population of zero mean with maximum deviation of ± 0.1. Estimate the system state vector $\underline{x}_k$

using (5.15) when the system starts with $\underline{x}_o = \begin{bmatrix} 1 \\ .1 \end{bmatrix}$. Results for a typical noise sequence $\{\nu_1, \nu_2 \ldots \nu_k\}$ are shown for both state variables $x1_k$ and $x2_k$ in Figs. 5.1 and 5.2. Note that the filter response closely matches the actual system response. The estimation error defined by

$$\underline{\tilde{x}}_k = \underline{\hat{\hat{x}}}_k - \underline{x}_k \tag{5.17}$$

is compared with the estimation error of the least-square-filter in Figs. 5.3 and 5.4.

B. SLIDING-WINDOW OBSERVER FOR TIME-INVARIANT SYSTEMS

For the case of time-invariant observable systems with a time-invariant observation matrix it is possible to increase the memory of the observer to an arbitrary length ℓ where $\ell \geq n$. This results in a sliding-window observer of length ℓ, where the state vector $\underline{x}_k$ is determined from the last ℓ observations in the least-square-error sense. Consider the first set of ℓ observations according to (5.1).

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_{\ell-1} \\ z_\ell \end{bmatrix} = \begin{bmatrix} M\Phi^{-\ell+1} \\ M\Phi^{-\ell+2} \\ \vdots \\ M\Phi^{-1} \\ M \end{bmatrix} \underline{x}_\ell \tag{5.18}$$

or

$$\underline{z}_\ell = A\, \underline{x}_\ell,$$

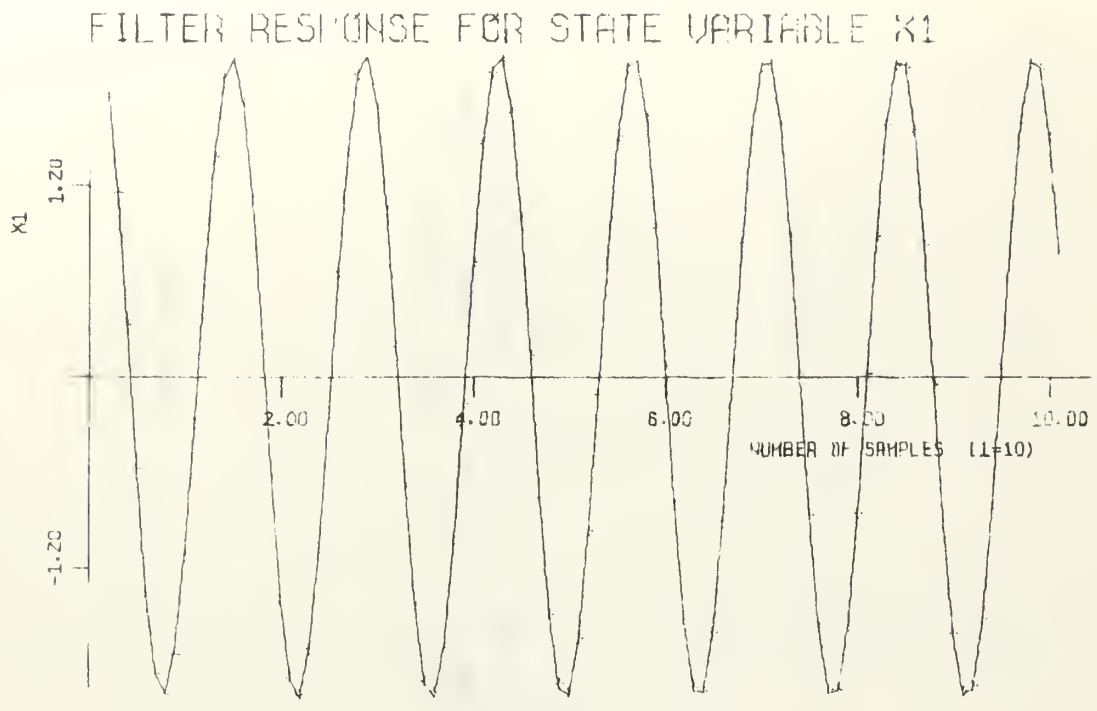

FIG. 5.1 EX. 5.1

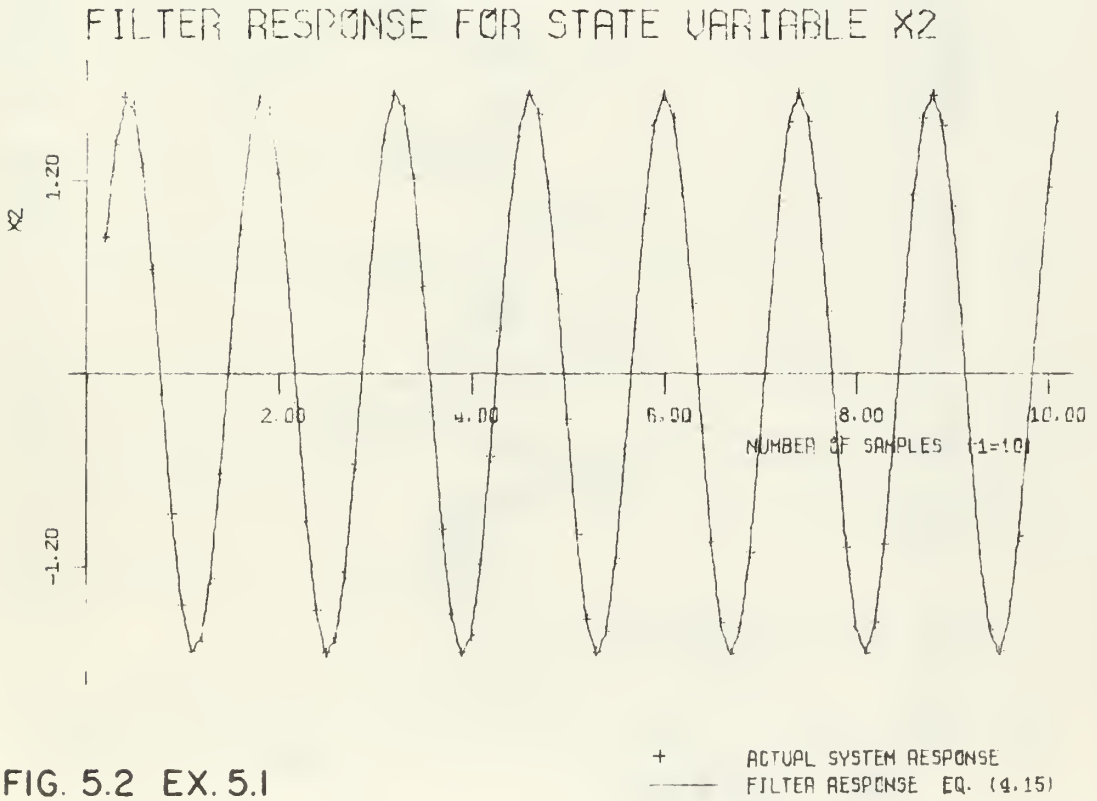

FIG. 5.2 EX. 5.1

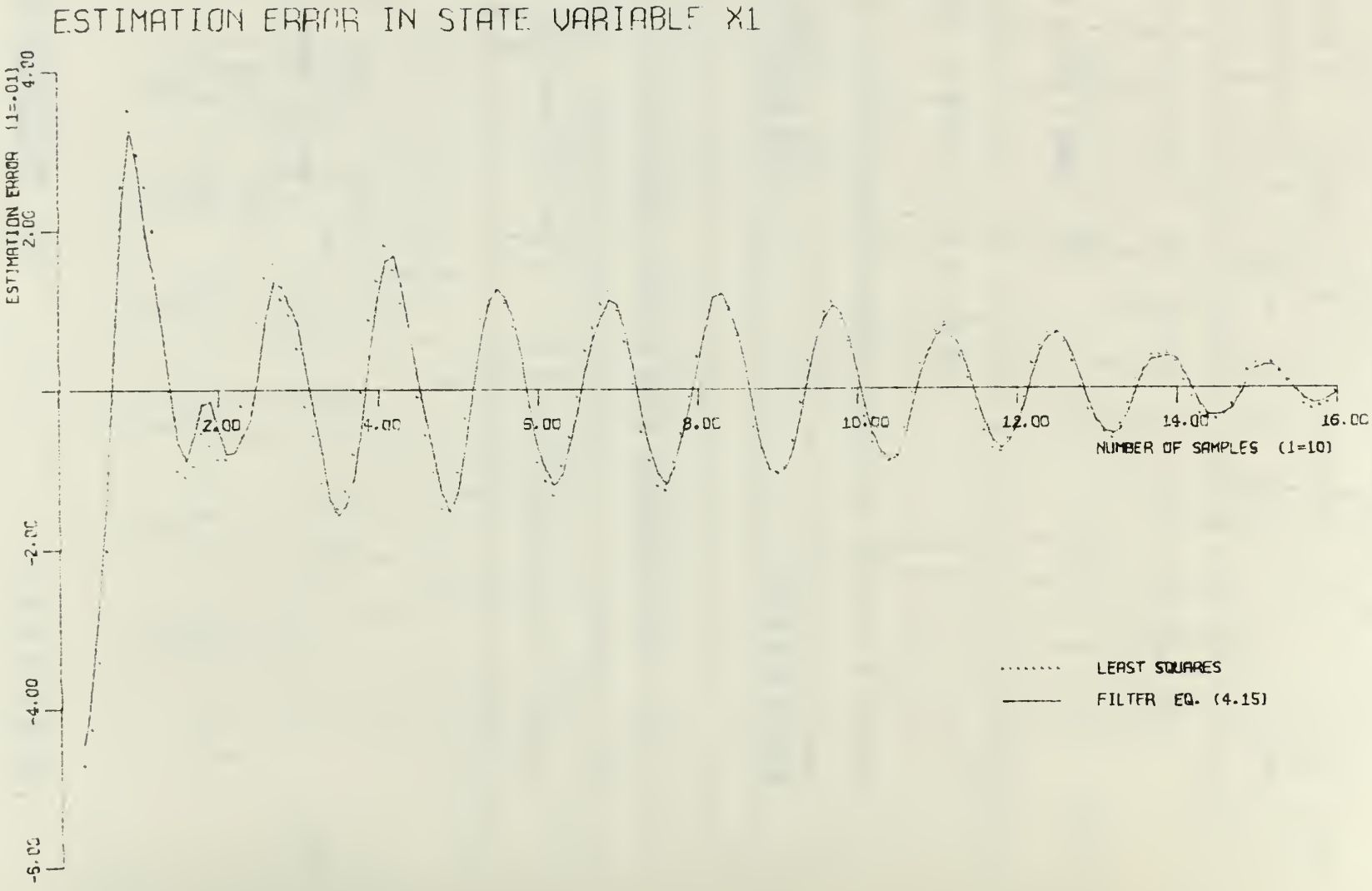
ESTIMATION ERROR IN STATE VARIABLE X1
ESTIMATION ERROR (1=.01)
4.00
2.00
-2.00
-4.00
-6.00
2.00
4.00
6.00
8.00
10.00
12.00
14.00
16.00
NUMBER OF SAMPLES (1=10)
LEAST SQUARES
FILTER EQ. (4.15)

FIG. 5.3 EX. 5.1

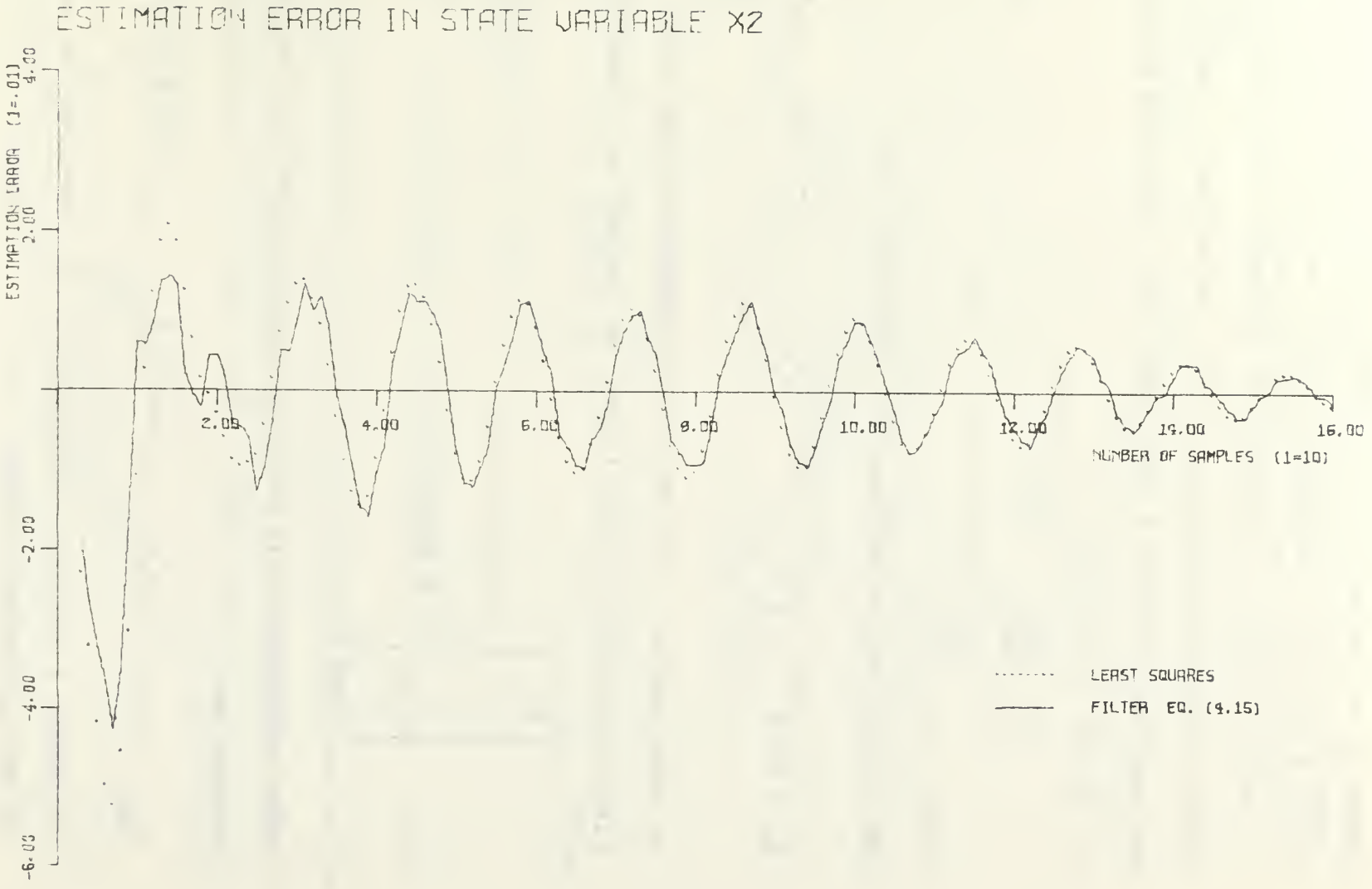
ESTIMATION ERROR IN STATE VARIABLE X2
ESTIMATION ERROR (1=.01)
4.00
2.00
-2.00
-4.00
-6.00
2.00
4.00
6.00
8.00
10.00
12.00
14.00
16.00
NUMBER OF SAMPLES (1=10)
LEAST SQUARES
FILTER EQ. (4.15)

FIG. 5.4 EX. 5.1

where $\underline{z}_\ell$ is the observation vector of the last ℓ observation, $\underline{x}_\ell$ is the n dimensional state vector at time instant ℓ, and A is the ℓ x n matrix of constants relating observations and state vector $\underline{x}_\ell$. Since the system is assumed to be observable matrix A has rank r = n. Then the solution to (5.18) is given by

$$\hat{\underline{x}} = PA^T\underline{z}_\ell \tag{5.19}$$

where $P = [A^TA]^{-1}$

Eq. (5.21) may be written more explicitly as

$$\hat{\underline{x}}_\ell = P\{(\Phi^{-\ell+1})^TM^Tz_1 + (\Phi^{-\ell+2})^TM^Tz_2 + \ldots + (\Phi^{-1})^TM^Tz_{\ell-1} + M^Tz_\ell\} \tag{5.20}$$

At time instant $\ell+1$ Eq. (5.18) takes the form of Eq. (5.21) since the memory of the filter is limited to ℓ observations.

$$\begin{bmatrix} z_2 \\ z_3 \\ \vdots \\ z_\ell \\ z_{\ell+1} \end{bmatrix} = A\,\underline{x}_{\ell+1} \tag{5.21}$$

Then analogous to (5.20), the solution of (5.21) may be written as

$$\hat{\underline{x}}_{\ell+1} = P\{(\Phi^{-\ell+1})^TM^Tz_2 + (\Phi^{-\ell+2})^TM^Tz_2 + \ldots + (\Phi^{-1})^TM^Tz_e + M^Tz_{e+1}\}$$

To obtain a recursive algorithm it is necessary to combine (5.22) and (5.20). Thus

$$\hat{\underline{x}}_\ell - P(\Phi^{-\ell+1})^TM^Tz_1 = P\Phi^T\{(\Phi^{-\ell+1})^TM^Tz_2 + \ldots + (\Phi^{-1})^TM^Tz\} \tag{5.23}$$

Substituting $z_1 = M\Phi^{-\ell+1}\hat{\underline{x}}_\ell$, which is true for noiseless observations, and premultiplying (5.24) with $P(\Phi^{-1})^T P^{-1}$ yields

$$P\{(\Phi^{-\ell+1})^T M^T z_2 + \ldots + (\Phi^{-1})^T M^T z_\ell\}$$

$$= P(\Phi^{-1})^T\{P^{-1} - (\Phi^{-\ell+1})^T M^T M\Phi^{-\ell+1}\}\ \hat{\underline{x}}_\ell$$

$$= P\{(\Phi^{-1})^T A^T A\Phi^{-1} - (\Phi^{-\ell})^T M^T M\Phi^{-\ell}\}\Phi\hat{\underline{x}}_\ell$$

$$= P\{(\Phi^{-1})^T[(\Phi^{-\ell+1})^T M^T M\Phi^{-\ell+1} + (\Phi^{-\ell+2})^T M^T M\Phi^{-\ell+2} + \ldots$$
$$+ M^T M]\Phi^{-1} - (\Phi^{-\ell})^T M^T M\Phi^{-\ell}\}\ \Phi\hat{\underline{x}}_\ell$$

$$= P\{(\Phi^{-\ell+1})^T M^T M\Phi^{-\ell+1} + \ldots + \Phi^{-T} M^T M\Phi^{-1}\}\Phi\hat{\underline{x}}_\ell$$

$$= P\{A^T A - M^T M\}\ \Phi\hat{\underline{x}}_\ell$$

$$= [I - (PM^T)M]\ \Phi\hat{\underline{x}}_\ell \qquad (5.24)$$

Identifying $\underline{g} = PM^T$ (5.24) may be written

$$P\{(\Phi^{-\ell+1})^T M^T z_2 + \ldots + (\Phi^{-1})^T M^T z_\ell\} = [I - \underline{g}M]\Phi\hat{x}_\ell \qquad (5.25)$$

It is interesting to note that $\underline{g}$ is given by the last column of A^+, as a generalization of (5.9b) of the previous section. Substituting (5.25) into (5.22) yields the desired recursive relationship

$$\hat{\underline{x}}_{\ell+1} = [I - \underline{g}M]\ \Phi\hat{\underline{x}}_\ell + \underline{g}z_{\ell+1} \qquad (5.26)$$

where the estimate $\hat{\underline{x}}_{\ell+1}$ is no longer dependent on the observation z_1. It then follows that the next estimate

$$\hat{\underline{x}}_{\ell+2} = [I-\underline{g}M]\,\Phi\underline{x}_{\ell+1} + \underline{g}_{\ell+2} \tag{5.27}$$

is no longer dependent upon the first two observations z_1 and z_2. Therefore the general recursive formulation for all discrete times $k > \ell$ is given by

$$\hat{\underline{x}}_{k-1} = F\hat{x}_k + \underline{g}z_{k+1} \tag{5.28}$$

where $F = [I-gM]\Phi$ and $\underline{g}$ remain constant. The complete algorithm for the general rectangular sliding-window observer, with the first state estimation available after ℓ observations are processed, is then

$$\begin{aligned} \hat{\underline{x}}_\ell &= [A^TA]^{-1}A^T\,\underline{z}_\ell \qquad k = \ell \\ \hat{\underline{x}}_{k+1} &= F\hat{x}_k + \underline{g}z_{k+1} \qquad k > \ell \end{aligned} \tag{5.29}$$

The estimation error

$$\tilde{\underline{x}}_k = \hat{\underline{x}}_k - \underline{x}_k \tag{5.30}$$

obeys the same dynamic relation as $\hat{\underline{x}}_k$. Thus

$$\tilde{\underline{x}}_{k+1} = F\tilde{\underline{x}}_k + \underline{g}\nu_{k+1} \tag{5.31}$$

If the covariance matrix of $\tilde{\underline{x}}_k$ is denoted by C_k it follows from (5.31) that

$$C_{k+1} = \tilde{x}_{k+1}\tilde{x}^T_{k+1} = FC_kF^T + \underline{g}R\underline{g}^T \tag{5.32}$$

where R is the variance of the measurement noise. Let

$$\varepsilon_k = |\sqrt{C_k(1,1)}\ \ | + |\sqrt{C_k(2,2)}\ |$$

where the sequence of matrices, C_k, is obtained from (5.32), $R = 1$, and $C_k(i,i)$ denotes the i^{th} diagonal element of the covariance matrix. ε_k is a relative measure of the expected absolute estimation error at time instant k. A comparison in terms of this relative absolute estimation error for a few sliding-windows of the system in Example 5.1 with the corresponding error of the least-square-error filter is shown in Fig. 5.5.

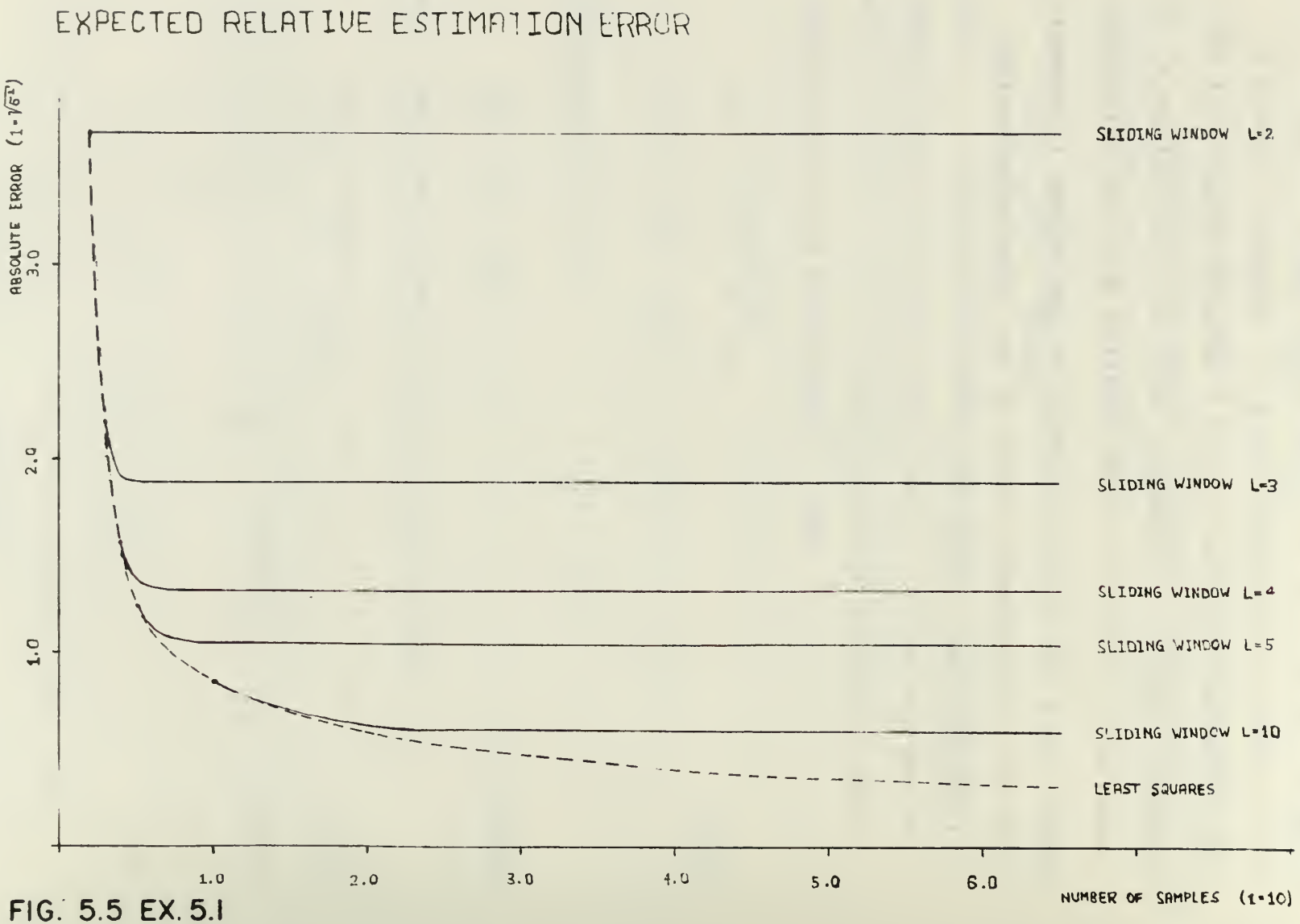
EXPECTED RELATIVE ESTIMATION ERROR
ABSOLUTE ERROR $(1-\sqrt{6^2})$
3.0
2.0
1.0
SLIDING WINDOW L=2
SLIDING WINDOW L=3
SLIDING WINDOW L=4
SLIDING WINDOW L=5
SLIDING WINDOW L=10
LEAST SQUARES
1.0
2.0
3.0
4.0
5.0
6.0
NUMBER OF SAMPLES (1=10)

FIG. 5.5 EX. 5.1

VI. SUMMARY AND RECOMMENDATIONS FOR FURTHER STUDY

(1) The normalized least-square-error solution of a set of inconsistent linear equatons has been established as an alternative to the usual least-square-error solution. The solution is obtained by minimizing a weighted least-square-error criterion and presented in recursive form for obtaining sequentially, the solution of a growing set of equations. The technique is illustrated by some estimation and identification problems. Further investigation is required to establish a decision criterion to determine whether the normalized least-square-error or the least-square-error solution method is to be preferred in engineering problems. It is expected that this decision criterion, in the case of problems involving discrete state estimation or parameter identification, will depend not only upon the system equations themselves but also upon the nature of the measurement noise.

(2) Complete recursive algorithms for the normalized least-square-error solution and the least-square-error solution based upon the concept of the best approximate solution [5] are presented.

(3) A finite step algorithm for the calculation of the pseudo inverse and the best approximate solution of a fixed set of linear equations is proposed. This result has

advantage over previously published recursive computation methods in that the algorithm presented starts unconditionally.

(4) Finite-memory, linear, observation filters in recursive form are proposed for the sequential state determination of a sampled-data system. Design rules for these observation filters, when the system is time-invariant, are given:

(a) for the minimum-window and averager observer where the system states as determined from the minimum set of past data, are smoothed by a simple averager. This procedure is shown to be very efficient for an oscillatory system with zero damping. Further investigation is recommended to improve the estimation for general systems, using the minimum-window observer together with a weighted averager. It is expected that optimal weighting can be approached using a scalar weighting factor in the recursive form.

(b) for the sliding-winding observer of memory length ℓ, $\ell > n$, where the n states of the system are determined from the last ℓ measurements in the least-square-error sense.

(5) In the Appendix, the recursive algorithm of the best approximate solution is applied to the numerical solution of a set of nonlinear equations. The results are promising in that solutions are obtained when other well known linearization or gradient techniques fail. The theory behind

this approach has not been investigated fully and it is recommended that further mathematical work be persued to establish rigorous proofs and conditions of convergence.

APPENDIX A

A. ITERATIVE SOLUTION OF A SET OF NONLINEAR EQUATIONS

The solution of simultaneous nonlinear equations is often impossible to obtain analytically, and graphical or iterative methods for a computer solution have to be employed. In addition, the solution of a class of nonlinear differential equations, as discussed in Section B, reduces, after integration, to the solution of nonlinear algebraic equations at each step of the integration process [16]. The most commonly used iterative methods are based upon linearization techniques, i.e., Newton-Raphson method and linear interpolation, or upon some gradient method whereby the iterative approximation to the solution is sequentially improved such that some error measure is forced to decrease [16]. All these methods however may not converge, or they may converge to a solution at infinity. In addition, the values of the functions as well as their gradients may have to be calculated at each iterative approximation. This calculation might be impossible if the approximation is outside the range or domain of one or more of the functions, or if one or more of the functions has a discontinuity at that particular approximation point for the solution. Further complications arise from the fact that the set of nonlinear equations may have more than one solution and the above iteration schemes may converge (if they converge at all) to an undesirable solution point.

Using the solution methods for a set of linear equations, as discussed in Chapter IV, a new iterative procedure is developed. This procedure circumvents the problems and drawbacks of the foregoing methods, and converges to a single point if one or more solution points exist. If this point is no solution point the initial estimate has to be displaced in the direction of the preferred solution point, where the preferred or desired solution is defined (in accordance with the concept of the minimum-norm solution) as the solution which is closest to the initial estimate. The class of nonlinear functions included in the iteration process in general are all functions which can be represented in polynomial form or which have power-series expansions.

1. Development of the Method

Let a set of n nonlinear functions in n unknowns be given as

$$\underline{h}(\underline{x}) = \begin{bmatrix} h_1(\underline{x}) \\ h_2(\underline{x}) \\ \vdots \\ h_n(\underline{x}) \end{bmatrix} = \underline{0} \qquad \text{(A.1)}$$

Using the polynomial form, or a power-series representation, (A.1) may be written as

$$A\underline{x} + B\underline{g}(\underline{x}) = \underline{c} \qquad \text{(A.2)}$$

where A and B are constant matrices of dimensions n x n and n x m respectively. The elements of the n x 1 vector

$\underline{c} = [c_1 \ldots c_n]^T$ are the negatives of the constant terms in the polynomial or series form. The vector $\underline{g}(\underline{x}) = [g_1(\underline{x}) \ldots g_m(\underline{x})]^T$ has dimensions m x 1. Its elements, the functions $g_i(\underline{X})$ are the nonlinear remainder terms of the corresponding function $h_i(\underline{x})$, or parts thereof, chosen such that the functions $g_i(\underline{x})$ are defined for all $\underline{x}$.

Range-or-domain-limited functions $g_i(\underline{x})$ can be accepted only if it is possible to reformulate these functions in terms of variables for which they are always defined. As an example consider the equation

$$y = \ell n x = 0 \tag{A.3a}$$

which has no real solution for x 0. However the same equation may be written as

$$e^y - x = 0 \tag{A.3b}$$

or

$$\{1 + y + \frac{y^2}{2!} + \frac{y^3}{3!} + \ldots\} - x = 0 \tag{A.3c}$$

or

$$[-i + 1] \begin{bmatrix} x \\ y \end{bmatrix} + [1] \, g_1(y) + 1 = 0 \tag{A.3d}$$

where

$$g_1(y) = \frac{y^2}{2!} + \frac{y^3}{3!} + \ldots. \tag{A.3e}$$

Now $g_1(y)$ is defined for all y and the domain-limited Eq. (A.3a) is acceptable in the iteration process in the form

of Eq. (A.3d). This separation of the nonlinear part of h_i from the linear and constant part has the advantage that the discontinuities of h_i do not appear in g_i. Thus, for example, if

$$2\,yx + x - 1 = 0 \tag{A.4a}$$

or

$$[1 \quad 0]\begin{bmatrix} x \\ y \end{bmatrix} + [2]\,xy = 1 \tag{A.4b}$$

y is not defined for $x = 0$. However the function

$$g_2(x,y) = x.y \tag{A.4c}$$

is defined for all x and y.

Let

$$\underline{x} = \underline{x}_o + \Delta\underline{x} \tag{A.5}$$

where $\underline{x}$ is the desired solution, $\underline{x}_o$ is the initial estimate for $\underline{x}$, and $\Delta\underline{x}$ is the necessary correction to $\underline{x}_o$ in order to satisfy Eq. (A.1). Eq. (A.2) may be rewritten as

$$A\Delta\underline{x} + B[\underline{g}(\underline{x}_o+\Delta\underline{x})-\underline{g}(\underline{x}_o)] = \underline{c} - A\underline{x}_o - B\underline{g}(\underline{x}_o) \tag{A.6a}$$

or

$$[A + BD\,(\underline{x}_o, \Delta\underline{x})]\,\Delta\underline{x} = \underline{c}' \tag{A.6b}$$

where $\underline{c}'$ is a vector of constants defined as the right side of Eq. (A.6a) and the elements of the m x m matrix D are defined from the total difference quotients of the functions

$g_i(\underline{x})$, $(i = 1,2,\ldots,m)$. Thus

$$[D(\underline{x}_o,\Delta\underline{x})]\ \Delta\underline{x} = \underline{g}(\underline{x}_o + \Delta\underline{x}) - \underline{g}(\underline{x}_o) \qquad (A.6c)$$

This may be best explained with an example. Again consider Eq. (A.4c). Then

$$\begin{aligned} & g_2(x_o + \Delta x, y_o + \Delta_y) - g_2(x_o, y_o) \\ & = (x_o + \Delta x)(y_o + \Delta y) - x_o y_o \\ & = x_o \Delta y + y_o\ \Delta x + \Delta x \Delta y \end{aligned} \qquad (A.7a)$$

which may be written as

$$\begin{bmatrix} y_o + \frac{\Delta y}{2} & x_o + \frac{\Delta x}{2} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \qquad (A.7b)$$

Then the terms $(y_o + \frac{\Delta y}{2})$ and $(x_o + \frac{\Delta x}{2})$ are elements of the matrix D. In general all the elements in the matrix D are dependent on the initial estimate, which is a constant vector during the iteration process, and the value of $\underline{\Delta x}$, which is unknown. The algorithm proposed sequentially updates an estimate for the elements of $D(\underline{x}_o,\underline{\Delta x})$ until the true values are obtained. Then the last approximation for $\underline{\Delta x}$ is the desired correction for the solution given in (A.5). From (A.1), and the given initial estimate $\underline{x}_o$, calculate A, B and $\underline{c}'$ as defined above. Let $\underline{\Delta x}_k$ denote the k^{th} approximation for $\underline{\Delta x}$ and set $\underline{\Delta x}_o = 0$. Then solve iteratively.

$$[A + BD\ (\underline{x}_o,\ \underline{\Delta x}_k)]\ \underline{\Delta x}_{k+1} = \underline{c}' \qquad (A.8)$$

Using (4.25) for $\underline{\Delta x}_{k+1}$ until (A.5) satisfies all equations in (A.1). The solution is obtained from (4.25) because the matrix in (A.8) may become singular at some step in the iteration process. This describes the basic technique, whereby the solution is obtianed by iterative approximation of the exact difference quotients of the functions $g_i(\underline{x})$ and not by iterative improvements of a previous approximation to the solution. If the process (A.8) converges and the error

$$\varepsilon = \sum_{i=1}^{n} |h_i| \qquad (A.9)$$

is acceptably small, the solution (A.5) usually is the desired solution closest to the initial estimate. However, if process (A.8) converges to a value of $\underline{\Delta x}$ for which ε is not small, then $\underline{x} = \underline{x}_o + \underline{\Delta x}$ lies between two or more solutions of (A.1). In this case a displacement of the initial estimate is necessary and the iteration has to be repeated. As shown in subsequent examples the values for the elements of $\underline{\Delta x}_k$ oscillate in a damped manner about their exact value. In order to increase the rate of convergence and, more importantly, to force convergence if the damping of these oscillations is slightly negative (which would eventually lead to divergence) additional damping may be introduced. This is accomplished by using a weighted average between $\underline{\Delta x}_{k-1}$ and $\underline{\Delta x}_k$ for the computation of the elements of $D(\underline{x}_o, \underline{\Delta x}_k)$. Then solve iteratively starting with $\underline{\Delta \hat{x}}_o = 0$ using (4.25)

$$\left\{ \begin{array}{l} [A + BD(\underline{x}o, \underline{\Delta x}_k]\ \underline{\Delta x}_{k+1} = \underline{c}' \\ \\ \underline{\Delta \hat{x}}_{k+1} = \alpha \underline{\hat{x}}_k + (1-\alpha)\ \underline{\Delta x}_{k+1} \end{array} \right\} \qquad \text{(A.10)}$$

where α, $0 \leq \alpha < 1$, is the coefficient of additional damping. α is determined from the rate of convergence of the process (A.10) starting with $\alpha = 0$. If after a few steps of the iteration are completed the convergence rate is considered too small, α may be increased and the process reinitiated.

This completes the development and discussion of the new algorithm for the solution of a set of nonlinear equations. In order to show the power of the iteration method (A.10) two examples, where other techniques may fail, are given below.

2. Experimental Results

Example 1

Find the point of itersection of the two curves

$$\left. \begin{array}{l} y + 2xy + x = 1 \\ y - xy + x^2 = 1 \end{array} \right\} \qquad \text{(A.11)}$$

closest to the given point (x_o, y_o).

According to (A.6a) and (A.6b) this set of equations may be written as

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 2 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} xy \\ x^2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad \text{(A.12a)}$$

or

$$\left\{\begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix} + \begin{bmatrix}2 & 0\\ -1 & 1\end{bmatrix}\begin{bmatrix}y_o+\frac{y}{2} & x_o+\frac{\Delta x}{2}\\ 2x_o+\Delta x & 0\end{bmatrix}\right\}\begin{bmatrix}\Delta x\\ \Delta y\end{bmatrix}$$

$$=\begin{bmatrix}1\\ 1\end{bmatrix} - \begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix}\begin{bmatrix}x_o\\ y_o\end{bmatrix} - \begin{bmatrix}2 & 0\\ 0 & -1\end{bmatrix}\begin{bmatrix}x_o y_o\\ x_o^2\end{bmatrix}$$

(A.12b)

Eq. (A.12b) is now in the desired form for (A.10).

Starting from the initial estimates

$$\text{a)}\ \begin{bmatrix}x_o\\ y_o\end{bmatrix} = \begin{bmatrix}0\\ 0\end{bmatrix}, \quad \text{b)}\ \begin{bmatrix}x_o\\ y_o\end{bmatrix} = \begin{bmatrix}-2\\ 1\end{bmatrix}, \quad \text{c)}\ \begin{bmatrix}x_o\\ y_o\end{bmatrix} = \begin{bmatrix}.8\\ 0\end{bmatrix}$$

with no additional damping (A.10) yields the following sequences for $\underline{\Delta x}_k$

$$\text{a)}\ \begin{bmatrix}-1.0\\ 2.0\end{bmatrix}, \begin{bmatrix}.33\\ 1.78\end{bmatrix}, \begin{bmatrix}-.6\\ 2.0\end{bmatrix}, \cdots \begin{bmatrix}-.1595\\ 1.703\end{bmatrix}$$

$$\text{b)}\ \begin{bmatrix}.25\\ 2.25\end{bmatrix}, \begin{bmatrix}.3975\\ 1.93\end{bmatrix}, \cdots, \begin{bmatrix}.3993\\ 1.895\end{bmatrix}$$

$$\text{c)}\ \begin{bmatrix}.883\\ -.263\end{bmatrix}, \begin{bmatrix}.516\\ -.052\end{bmatrix} \cdots, \begin{bmatrix}.5993\\ -.1051\end{bmatrix}$$

Thus the following solutions are obtained

$$\text{a)}\ \begin{bmatrix}x\\ y\end{bmatrix} = \begin{bmatrix}-.1595\\ 1.703\end{bmatrix}$$

b) $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1.399 \\ -.105 \end{bmatrix}$

c) $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1.399 \\ -.105 \end{bmatrix}$

The graphs of the functions in (A.11) and the three solutions are shown in Fig. A.1. Fig. A.2 shows the oscillation of the sequential values $\underline{\Delta x}$ for case (a). Fig. A.4 illustrates the dependence of the rate of convergence upon the additional damping α where it is assumed that the solution is obtained whenever $\varepsilon < 10^{-4}$.

Note that in case (b) none of the iteration methods that depend upon function values could have initiated an iteration and that in case (c) other methods would have converged to a different solution point.

Example 2

Find the point of intersection of the two curves

$$\left\{\begin{array}{l} y + 2xy + x = 1 \\ \quad 2xy + x = -2 \end{array}\right\} \qquad \text{(A.13a)}$$

closest to the point of (x_o, y_o)

The solution $\begin{bmatrix} x^* \\ y^* \end{bmatrix} = \begin{bmatrix} -2/7 \\ +3 \end{bmatrix}$, obtained directly from substitution is the only finite solution. Thus, no matter what the initial estimate is, the desired solution is $[x^* \quad y^*]^T$. The set of Eq. (A.13) may be rewritten as

ILLUSTRATION TO EXAMPLE 1

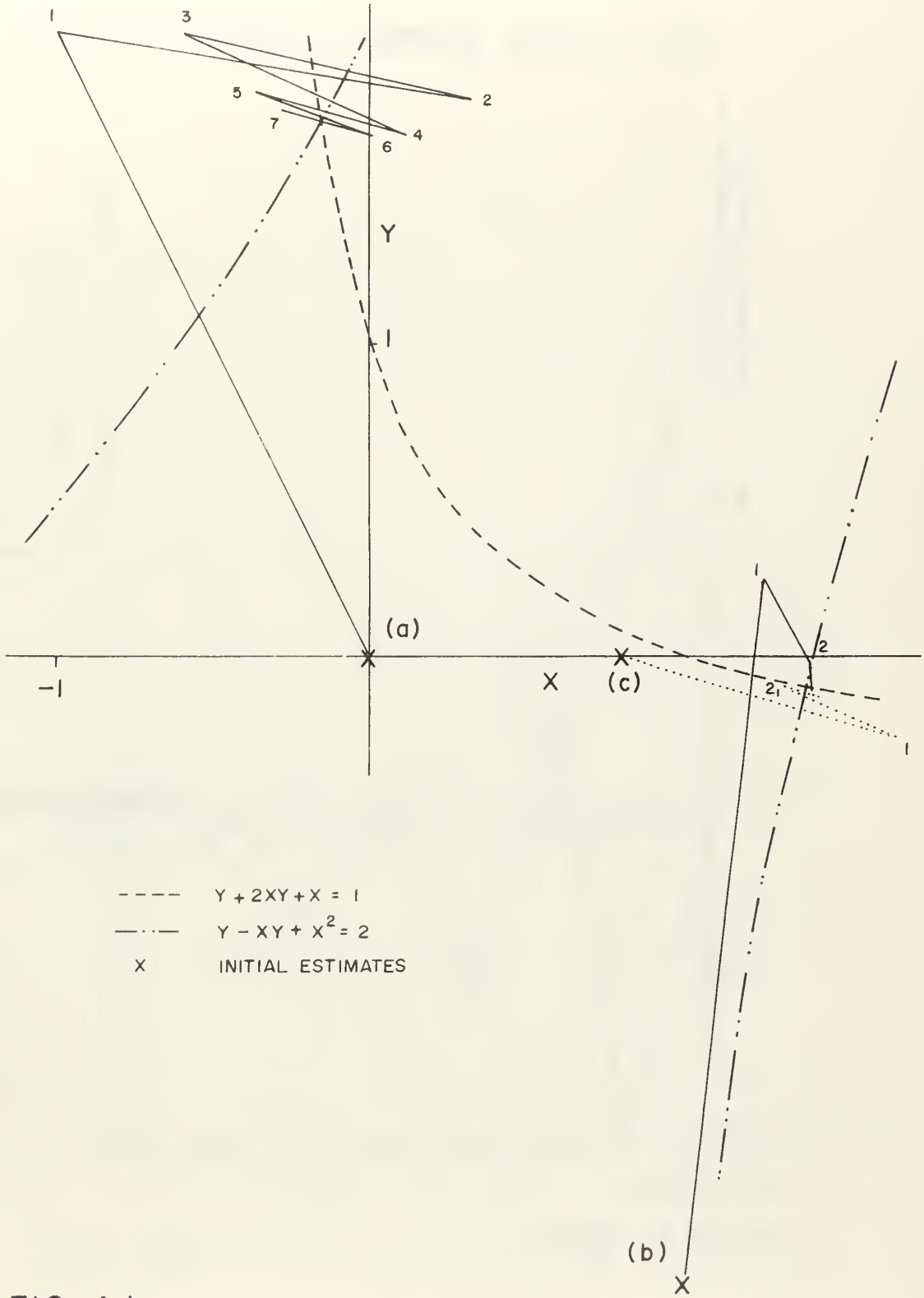

FIG. A.1

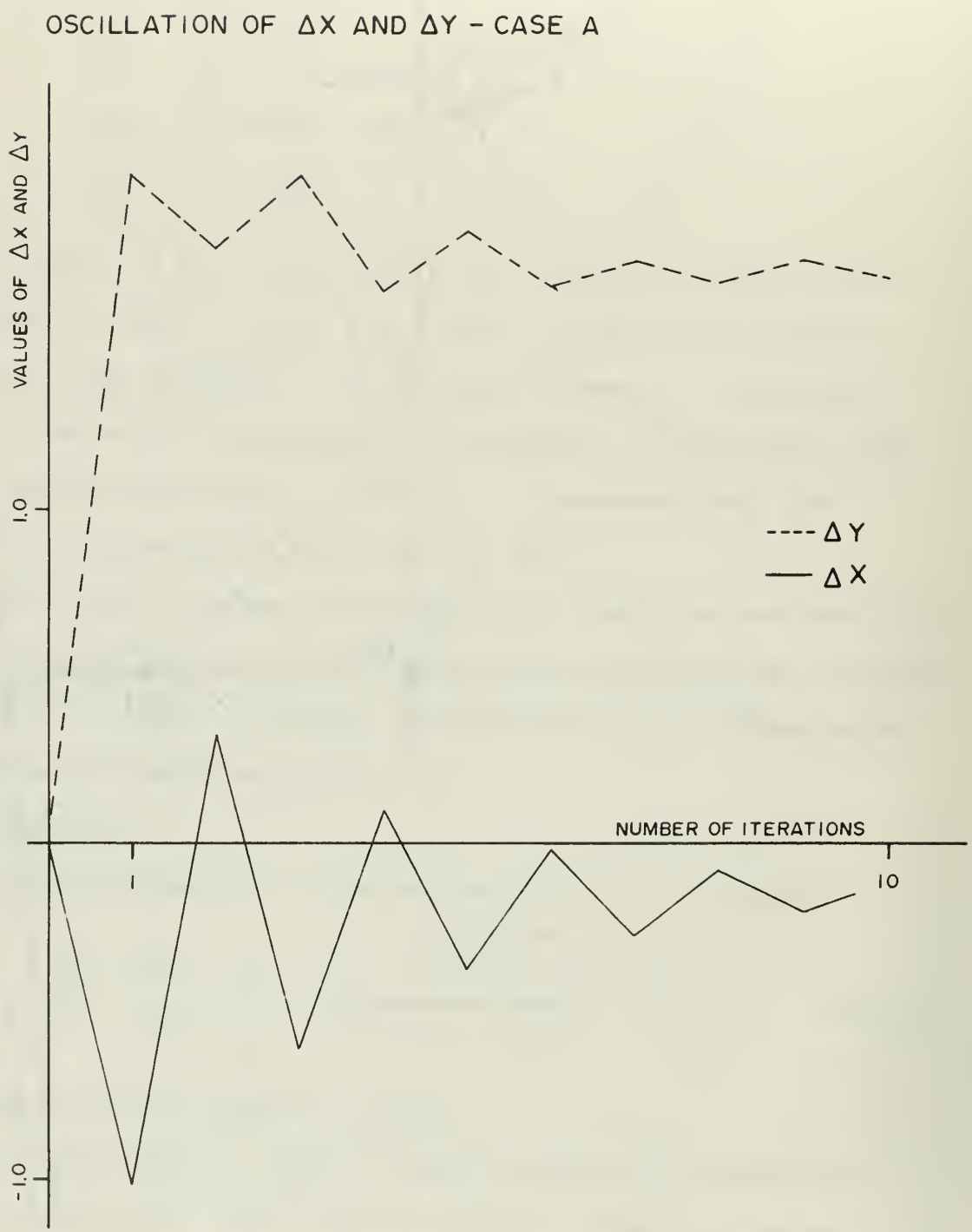
OSCILLATION OF ΔX AND ΔY - CASE A
VALUES OF ΔX AND ΔY
1.0
-1.0
---- ΔY
— ΔX
NUMBER OF ITERATIONS
1
10

FIG. A.2 EX. 1

CONVERGENCE AS A FUNCTION OF
ADDITIONAL DAMPING - EXAMPLE I.

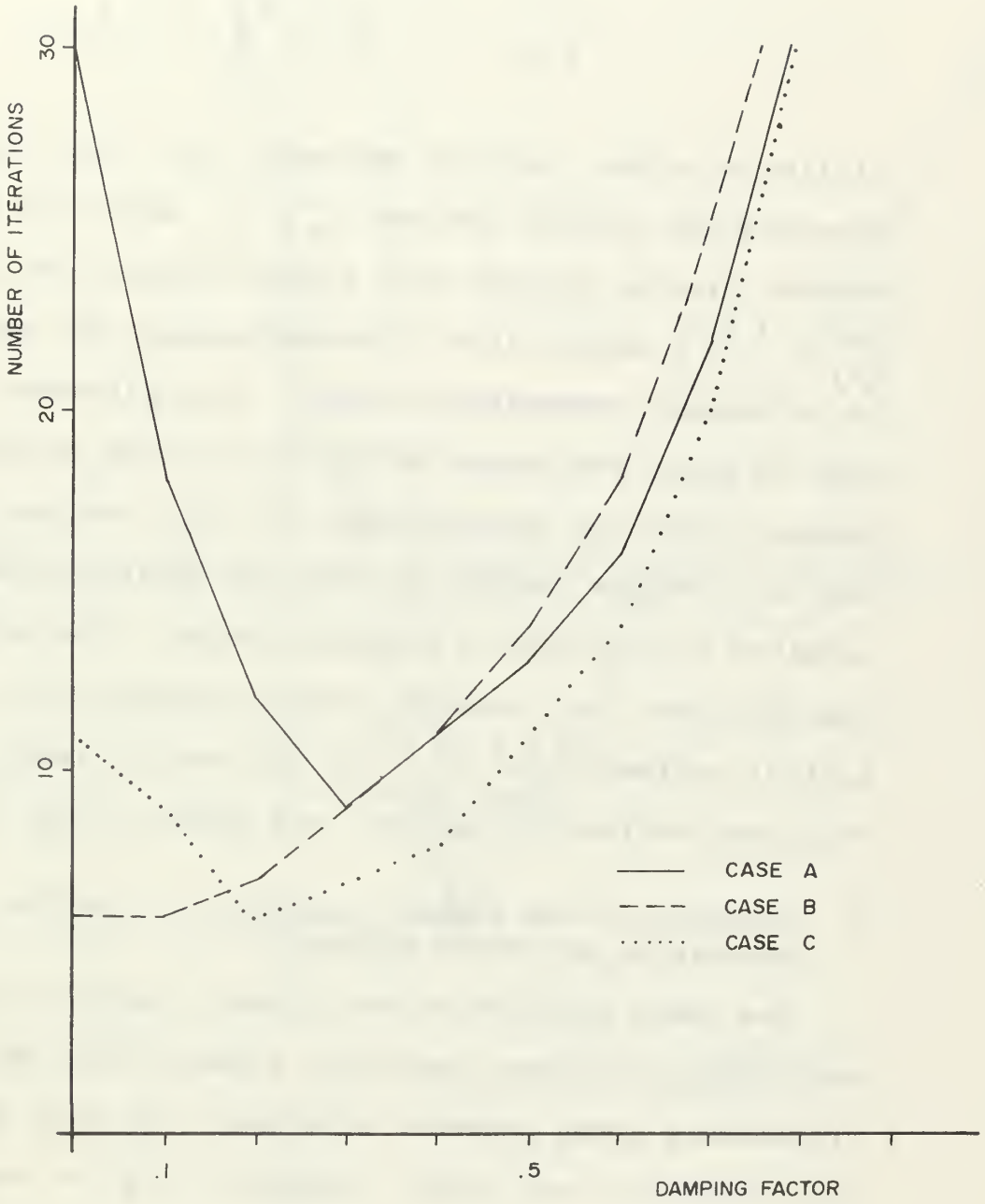

FIG. A.3

$$\left\{\begin{bmatrix}1 & 1\\ 1 & 0\end{bmatrix} + \begin{bmatrix}2\\ 2\end{bmatrix}\begin{bmatrix}y_o+\frac{\Delta y}{2} & x_o+\frac{\Delta x}{2}\end{bmatrix}\right\}\begin{bmatrix}\Delta x\\ \Delta y\end{bmatrix}$$

$$= \begin{bmatrix}1\\ -2\end{bmatrix} - \begin{bmatrix}1 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}x_o\\ y_o\end{bmatrix} - \begin{bmatrix}2\\ 2\end{bmatrix} x_o y_o \qquad \text{(A.13b)}$$

similar to other iteration methods, (A.9) will diverge whenever the initial estimate $x_o > 0$. While the other methods diverge because they iterate towards the solution $\begin{bmatrix}x\\ y\end{bmatrix} = \begin{bmatrix}\infty\\ -.5\end{bmatrix}$, method (A.9) diverges because the oscillations in Δx become increasingly larger. This situation is remedied by using the method of Eq. (A.10) with an additional damping factor of approximately $\alpha = .7$. Now the iteration for Δx converges rapidly so that the desired solution is obtained in only 8 to 9 iteration steps. The solution for the first few iteration steps, starting with the initial estimate $\begin{bmatrix}x_o\\ y_o\end{bmatrix} = \begin{bmatrix}1\\ -2\end{bmatrix}$ as well as the graphs of the functions defined by (A.13a), are shown in Fig. A.4.

B. SOLUTION OF THE DYNAMIC RESPONSE OF CIRCUITS CONTAINING NON-LINEAR RESISTIVE ELEMENTS*

The exact solution of the dynamic response of a circuit containing non-linear resistive elements such as diodes or transistors often presents problems, even when the non-linear characteristics are known, because it may not be possible to solve for the required variables explicitly [6]. The

---

*The material in this section up to Eq. (A.21) has been published in the Proceedings of the Second Annual Princeton Conference on Information Sciences and Systems, 1968.

ILLUSTRATION TO EXAMPLE 2

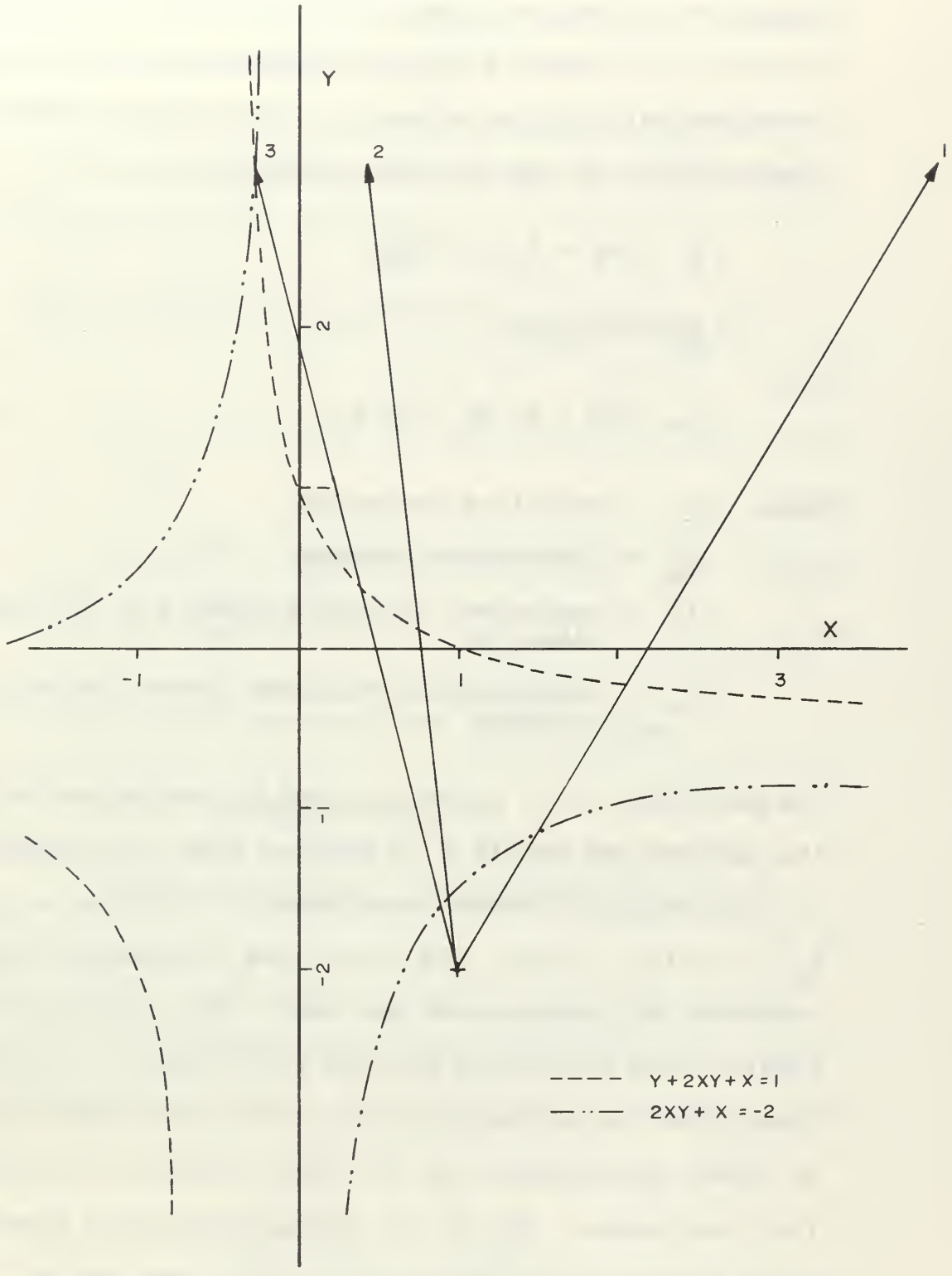

FIG. A.4

non-linear resistive elements are considered as dependent current (or voltage) sources, dependent upon state or other variables in the network. The network is then characterized by the following equations.

$$\dot{\underline{x}} = A\underline{x} + B_1\underline{u}_s + B_2\underline{i}_n \tag{A.14}$$

$$\underline{i}_n = N\,\underline{f}(\underline{v}_n) \tag{A.15}$$

$$\underline{v}_n = R\underline{x} + S_1\,\underline{u}_s + S_2\,\underline{i}_n \tag{A.16}$$

where $\underline{x}$ = the state variables

$\underline{u}_s$ = independent sources

$\underline{i}_n$ = equivalent current sources for the non-linear elements

$\underline{v}_n$ = corresponding voltages across the non-linear elements

The matrices A, B_1, B_2, R, S_1, and S_2 are determined from the network and matrix N is defined from the characteristics of the nonlinear network components. The roles of $\underline{v}_n$ and $\underline{i}_n$ in (A.14), (A.15), and (A.16) are interchanged when dependent voltage sources are used. Eq. (A.14) is the state equation for the linear part of the circuit. Eq. (A.15) formulates the characteristics of the non-linear components as given, for example, by the Ebers and Moll [20] equations for transistors. Eq. (A.16) gives the circuit constraints (loop equations for equivalent current sources and nodal equations for equivalent voltage sources) between the

voltage across the nonlinear elements and their currents, the states, and the known sources.

For a discrete time solution the sources, $\underline{u}_s$ and $\underline{i}_n$, may be approximated as piecewise linear functions. Thus the solution to (A.14) is given by

$$\underline{x}_{k+1} = \Phi\underline{x}_k+\Gamma_1 B_1\underline{u}_s(k)+\Gamma_2 B_1\underline{u}_s(k+1)+\Gamma_1 B_2\underline{i}_n(k)+\Gamma_2 B_2\underline{i}_n(k+1) \tag{A.17}$$

where

$$\Phi = e^{AT} \tag{A.18}$$

$$\Gamma_1 = A^{-1}[e^{AT} - \frac{A^{-1}}{T}(e^{AT} - I)] \tag{A.19}$$

$$\Gamma_2 = A^{-1}[\frac{A^{-1}}{T}(e^{AT} - I) - I] \tag{A.20}$$

Substituting (A.15) and (A.17) in (A.16) yields

$$\underline{V}_n(k+1) = R\,\underline{x}(k)+R_1[B_1\underline{u}_s(k)+B_2\underline{i}_n(k)] + (S_1+R_2 B_1)\underline{u}_s(k+1) + (S_2+R\Gamma_2 B_2)N\,\underline{f}[\underline{v}_n(k+1)] \tag{A.21}$$

Knowing the values $\underline{x}(k)$, $\underline{u}_s(k+1)$, $\underline{i}_n(k)$, Eq. (A.21) represents n simultaneous non-linear equations which have to be solved for $\underline{v}_n(k+1)$. Eq. (A.21) may be written in the form

$$A'\underline{v}_n(k+1) + B'\begin{bmatrix} g_1(\underline{v}_n) \\ \vdots \\ g_m(\underline{v}_n) \end{bmatrix} = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} \tag{A.22}$$

For diodes or transistors the functions $g_i(\underline{v}_n)$ are function of one variable only and have the general form

$$g_i(v_i) = (e^{kv_i} - 1) \tag{A.23}$$

Thus the method (A.10) is directly applicable and yields with the estimate $\underline{v}_n(k)$ the solution closest to this point $\underline{v}_n(k+1)$. Using $\underline{v}_n(k+1)$ in (A.15) and (A.16) enables $\underline{i}_n(k+1)$ to be calculated. Using these as initial values in (A.21) permits the calculation cycle to be reiterated.

LIST OF REFERENCES

1. R. Deutsch, "Estimation Theory," Prentice-Hall, Inc., 1965.

2. R. E. Kalman, "Fundamental Study of Adaptive Control Systems," Technical Report No. ASD-TR-61-27, Vol.I, April, 1962.

3. R. C. K. Lee, "Optimal Estimation, Identification, and Control," Research Monograph No. 28, The Massachusetts Institute of Technology, 1964.

4. R. Penrose, "A Generalized Inverse for Matrices," Proceedings of the Cambridge Philosophical Society, Vol. 51, 1955.

5. R. Penrose, "On Best Approximate Solutions of Linear Matrix Equations," Proceedings of the Cambridge Philosophical Society, Vol. 52, 1955.

6. H. E. Koenig, Y. Tokad, H. K. Kesaran, "Analysis of Discrete Physical Systems," McGraw-Hill Book Co., New York, N.Y., 1967, pp. 150-151.

7. C. H. Wells, "Minimum Norm Control of Discrete Systems," Doctorial Dissertation, Washington University, St. Louis, Missouri, 1966.

8. I. Zaborszky, R. G. Marsh, R. E. Janitch, M. K. Chidambera, and E. E. Buder, "Theoretical and Exponential Research on Digital Adaptive Control System," NASA CR-669, 1967.

9. F. Ayres, "Matrices," Schaum's Outline Series, 1962.

10. J. I. Nagumo and A. Noda, "A Learning Method for System Identification," IEEE Trans. AL Vol.AC-12, No. 3, June, 1967.

11. D. G. Luenberger, "Observing the State of a Linear System," IEEE Trans. Military Electronics, Vol. MIL-8, April, 1964.

12. B. E. Bona, "Applications of Observers and Optimum Filters to Inertial Systems," X8-1007/501, Autonetics Division of North American Rockwell Corp., April 1968.

13. B. E. Bona, "Some Observer Problems," Internal Letter, North American Rockwell Corp., Aerospace and System Group, March, 1969.

14. F. H. Branin, "Computer Methods of Network Analysis," Proc. IEEE, Vol. 55, No. 11, Nov. 1967.

15. J. Katzenelson and L. H. Seitleman, "An Iterative Method for Solution of Networks of Nonlinear Monotone Resistors," Trans. IEEE, Vol. CT-13, Sept.1966, pp. 317-322.

16. D. A. Calahan, "Computer-Aided Network Design," McGraw-Hill Book Co., New York, N.Y., Preliminary Edition, McGraw-Hill Cook Co., 1961.

17. E. Mishkin and L. Braun, "Adaptive Control Systems," McGraw-Hill Book Co., 1961.

18. V. W. Eveleigh, "Adaptive Control and Optimization Techniques," McGraw-Hill Book Co., 1967.

19. P. Eykhoff, "Process Parameter and State Estimation," Autonetica, Vol. 4, pp. 205-233, 1968.

20. I. I. Ebbers and J. L. Moll, "Large-Signal Behavior of Junction Transistors," Proc, IRE, pp. 1761-1772, Dec. 1954.

21. W. A. Brown, "Optimization of a Sampled-Data System using a Sum-of-Absolute-Error Index," Proc. Hawii International Conf. on Systems and Sciences, pp. 359-362.

22. T. N. E. Greyville, "Some Applications of the Pseudo-inverse of a Matrix," SIAM Review, Vol. 2, No. 1, Jam. 1960, pp. 15-22.

23. L. D. Pyle, "Generalized Inverse Computations using the Gradient Projection Method," J. Assoc. Comp. Mach., Vol. 11, No. 4 (Oct. 1964); pp. 422-428.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	20
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Professor Sydney R. Parker Dept. of Electrical Engineering Naval Postgraduate School Monterey, California 93940	5
4.	Dr. Blair Bona Autonetics Division North American Rockwell Corp. 3370 Miraloma Avenue Anaheim, California 92803	1
5.	Assoc. Professor Glen A. Meyers Dept. of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
6.	LCDR Manfred H. F. Werther Ascheffel, Kr. Eckernförde GERMANY	5

DOCUMENT CONTROL DATA - R & D

Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1 ORIGINATING ACTIVITY (Corporate author)
Naval Postgraduate School
Monterey, California 93940

2a. REPORT SECURITY CLASSIFICATION
Unclassified

2b. GROUP

3 REPORT TITLE
RECURSIVE ALGORITHM FOR THE BEST APPROXIMATE SOLUTION OF LINEAR EQUATIONS WITH APPLICATIONS TO SYSTEM IDENTIFICATION AND STATE ESTIMATION

4 DESCRIPTIVE NOTES (Type of report and, inclusive dates)
Doctor of Philosophy

5 AUTHOR(S) (First name, middle initial, last name)
Manfred Hermann Fritz Werther, Lieutenant Commander, Fed. German Navy

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
October 1969	143	23

8a. CONTRACT OR GRANT NO	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT
This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

The sequential solution, in recursive form, of a growing set of linear equations, based upon the least-square-error and a weighted least-square-error criterion, are developed. For comparison these results are applied to the discrete-time solution of several estimation and identification problems. Recursive algorithms for pseudo inversion and the best approximate solution of a set of linear equations are included. Finally, efficient state estimation procedures for time-invariant systems using a sliding-window obwerver are presented.

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
RECURSIVE ALGORITHM						
PSEUDO INVERSE						
SYSTEM IDENTIFICATION						
ESTIMATION						
SYSTEM OBSERVER						
LEAST-SQUARE-ERROR						

thesW472
Recursive algorithm for the best approxi
3 2768 000 99697 9
DUDLEY KNOX LIBRARY