NASA CONTRACTOR REPORT

NASA CR-756

NASA CR-756

N 67-23716
(ACCESSION NUMBER)

(THRU)
1
(CODE)

31
(PAGES)

CR-756
(NASA CR OR TMX OR AD NUMBER)

10
(CATEGORY)

FACILITY FORM 602

SYSTEM IDENTIFICATION AND PATTERN RECOGNITION

by Rob Roy and James Sherman

Prepared by
RENSSELAER POLYTECHNIC INSTITUTE
Troy, N. Y.
for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • APRIL 1967

29 NASA CR-756 END

3 SYSTEM IDENTIFICATION AND PATTERN RECOGNITION 6

By Rob Roy and James Sherman 9

Distribution of this report is provided in the interest of information exchange. Responsibility for the contents resides in the author or organization that prepared it.

26 A

Prepared under Grant No. NGR-33-018-014 by
1 RENSSELAER POLYTECHNIC INSTITUTE
Troy, N.Y. 3

for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 – CFSTI price $3.00

SYSTEM IDENTIFICATION AND PATTERN RECOGNITION

Rob Roy
Associate Professor
Electrical Engineering Department
Rensselaer Polytechnic Institute
Troy, New York

and

James Sherman
Research Assistant
Electrical Engineering Department
Rensselaer Polytechnic Institute
Troy, New York

Introduction

This paper presents a new technique for determining an input-output model of a system with finite settling time. This technique grew out of a realization that the problem of system identification is closely related to the problem of pattern recognition.[1,2] Therefore, before the mathematical relationships are given, an explanation of the concept of the technique is presented.

Consider a finite settling time system. The relevant input past of such a system can be represented by a set of N uniformly spaced samples, such as those obtained at the taps of a lossless delay line. The set of sample time points forms an orthogonal set of coordinates, such that the input past is a vector in this coordinate system. The output of the system is a scalar function of the input vector. The system performs a transformation of the input vector to the scalar output value. This transformation can be viewed as a hypersurface in an $N + 1$ dimensional space. The projection of the tip of the input vector to the hypersurface is the value of the system output. For a linear system the hypersurface is a hyperplane, the system output being a linear combination of the input sample values. System identification is the determination of the transformation hypersurface.

Pattern recognition is closely related to system identification. A pattern is characterized by a set of pattern features, such that the pattern can be viewed as a point in an N dimensional feature space. Each pattern belongs to a specific category. The problem is to sort the patterns into their proper categories. This is done by observing a training set of patterns, where the correct category for each pattern is known. After observing the training set, the pattern recognizer must correctly categorize test patterns whose correct category is unknown.

Pattern recognition is usually performed by determining a set of surfaces, or discriminant functions, which separate the training set into their correct categories. These surfaces are determined by first assuming a general form for the surfaces and then iteratively adjusting these surfaces after observing each member of the training set. This is called "nonparametric" training.[3]

The analogy between Pattern Recognition and System Identification is clear. Both require the determination of a hypersurface. If the input data to the system is viewed as a sequence of patterns, then the well developed techniques of pattern recognition can be applied. A general form is assumed for the system transformation and the error-correcting training algorithms are used to determine the specific hypersurface. Note that it is the viewpoint which is important, as this viewpoint naturally leads to the use of techniques from a seemingly unrelated field.

Volterra Series Representation[4]

Consider the linear system of Fig. (1). The output $y(t)$ is given by the convolution integral

$$y(t) = \int_{-\infty}^{\infty} h(\tau)\, x(t - \tau)\, d\tau \tag{1}$$

Clearly,

$$|y(t)| = \left| \int_{-\infty}^{\infty} h(\tau)\, x(t - \tau)\, d\tau \right| \leq \int_{-\infty}^{\infty} |h(\tau)|\, d\tau \cdot \sup_t |x(t)| \tag{2}$$

If

$$\int_{-\infty}^{\infty} \left| h(t) \right| dt < \infty \tag{3}$$

then a bounded input to the system produces a bounded output. Such a system is called "stable". The systems that will be considered here are those systems which are called "stable".

Next, examine the nonlinear system of Fig. (2). Since

$$z(t) = y^2(t) \tag{4}$$

the output of the system is given by

$$\begin{aligned} z(t) &= \int_{-\infty}^{\infty} h_1(\tau_1)\, x(t - \tau_1)\, d\tau_1 \int_{-\infty}^{\infty} h_2(\tau_2)\, x(t - \tau_2)\, d\tau_2 \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_1(\tau_1)\, h_2(\tau_2)\, x(t - \tau_1)\, x(t - \tau_2)\, d\tau_1\, d\tau_2 \end{aligned} \tag{5}$$

If a two dimensional kernel $h_2(\tau_1, \tau_2)$ is defined as

$$h_2(\tau_1, \tau_2) = h_1(\tau_1)\, h_1(\tau_2) \tag{6}$$

then

$$z(t) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_2(\tau_1, \tau_2)\, x(t - \tau_1)\, x(t - \tau_2)\, d\tau_1\, d\tau_2 \tag{7}$$

The two dimensional kernel $h_2(\tau_1, \tau_2)$ is called a "regular homogeneous" functional of second degree. This kernel is "realizable" if

$$h_2(\tau_1, \tau_2) = 0 \qquad \text{for either } \tau_1 \text{ or } \tau_2 < 0 \tag{8}$$

and "stable" if

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left| h_2(\tau_1, \tau_2) \right| d\tau_1\, d\tau_2 < \infty \tag{9}$$

Those functionals with realizable kernels are called volterra kernels. Kernels of this type play an important role in the analysis[5,6] and synthesis[7,8] of nonlinear systems.

Next, consider the case where the nonlinear block is an arbitrary continuous function.

$$z(t) = f\left[y(t)\right] \tag{10}$$

The function $f(y)$ can be approximated* by a finite sequence of polynomials

$$f(y) \approx f_N(y) = \sum_{i=1}^{N} a_i\, y^i \tag{11}$$

Consequently $z_N(t)$ can be expressed as

$$\begin{aligned} z_N(t) = a_o &+ a_1 \int_{-\infty}^{\infty} h_1(\tau)\, x(t-\tau)\, d\tau + a_2 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_2(\tau_1, \tau_2)\, x(t-\tau_1)\, x(t-\tau_2) d\tau_1\, d\tau_2 \\ &+ \;\cdots\cdots \qquad\qquad \cdots\cdots \\ &+ a_N \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} h_N(\tau_1 \ldots \tau_N)\, x(t-\tau_1) \ldots x(t-\tau_N)\, d\tau_1 \ldots d\tau_N \end{aligned} \tag{12}$$

or

$$z_N(t) = \sum_{i=1}^{N} a_i \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} h_i(\tau_1, \ldots, \tau_i)\, x(t-\tau_1) \ldots x(t-\tau_i)\, d\tau_1 \ldots d\tau_i \tag{13}$$

where

$$h_i(\tau_1, \ldots, \tau_i) = \prod_{j=1}^{i} h_j(\tau_j) \tag{14}$$

---

*The Weierstrass theorem assures that a sequence of polynomials exist which converge everywhere to $f(y)$. For bounded functions, this implies convergence in the mean. Thus, discontinuous nonlinearities are excluded.

If $f(y)$ is analytic in a given region, then $f(y)$ can be expanded in a power series

$$f(y) = \sum b_i \, y^i \tag{15}$$

and

$$z(t) = \sum_{i=1}^{\infty} b_i \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h_i(\tau_1, \ldots, \tau_i) \, x(t-\tau_1) \ldots x(t-\tau_i) \, d\tau_1 \ldots d\tau_i \tag{16}$$

Since the power series (15) will converge for all

$$|y(t)| < \varepsilon$$

the functional power series will converge for all

$$|x(t)| < \frac{\varepsilon}{\int_{-\infty}^{\infty} |h(\tau)| \, d\tau} \tag{17}$$

Systems which can be represented by a functional power series with a nonzero radius of convergence are called "analytic systems".[9] Although the limit of (13) and expression (16) are the same in the region of convergence of the functional power series, expression (16) is restricted in its range of validity.

If the system was time varying, then Eq. (13) would be extended to the more general expression

$$z_N(t) = \sum_{i=1}^{N} a_i \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h_i(t, \tau_1 \ldots \tau_i) \, x(\tau_1) \ldots x(\tau_i) \, d\tau_1 \ldots d\tau_i \tag{18}$$

The systems which will be investigated are the class of nonlinear systems whose output depends to an arbitrarily small extent on the remote past. In other words, finite settling time systems. Wiener[10] showed that any nonlinear system with finite settling time could be characterized by a linear network which characterized the input past, followed by a zero memory nonlinearity. Dr. Wiener used a

Laguerre network to produce an orthogonal representation of the input past, then followed this network with a set of Hermite polynomials which represented the zero memory nonlinearity. This cascade of two operations is essentially a specific form of the functional approach of volterra. In this case the values of the functionals depend on the values of a real function over a finite interval. The functions are continuous and square integrable over a finite interval. This approach was also studied by Cameron and Martin.[11]

Consider the case where the representation of the input past consists of a set of sample values. Thus

$$\underline{X}(t) = \text{input vector} = \text{col}\left[x_1(t) \quad x_2(t) \; \ldots\ldots \; x_n(t)\right]$$

$$x_i(t) = x(t - (i-1)T)$$

$$T = \text{sampling interval}$$

$$nT = \text{settling time of system} \tag{19}$$

Furthermore, the input will be assumed to be piecewise constant

$$x(t) = x_i \qquad (i-1)T \leq t < i\,T \qquad |x_i| \leq X \tag{20}$$

This type of input is inherent in a digital computer controlled system.

Under these assumptions, Eq. (13) becomes[12]

$$z_N(t) = \sum_{i=1}^{N} \sum_{k_1=1}^{n} \cdots \sum_{k_i=1}^{n} H_{k_1 \ldots k_i}(t)\, x_{k_1} \cdots x_{k_i} \tag{21}$$

where

$$H_{k_1 \ldots k_i}(t) = \begin{cases} 0 & t < mT \\ \displaystyle\int_{k_1T}^{M} \cdots \int_{k_iT}^{M} h_i(t, \tau_1 \cdots \tau_i)\, d\tau_1 \ldots d\tau_i & mT \leq t < (m+1)T \\ \displaystyle\int_{k_1T}^{(k_1+1)T} \cdots \int_{k_iT}^{(k_i+1)T} h_i(t, \tau_1 \ldots \tau_i)\, d\tau_1 \ldots d\tau_i & (m+1)T \leq t \end{cases} \tag{22}$$

$$m = \max \{k_1, k_2, \ldots, k_i\}$$

$$M = \begin{cases} (k_i+1)T & \text{for} \quad k_i < m \\ t & \text{for} \quad k_i = m \end{cases}$$

If Eq. (21) is expanded, taking into account the symmetry of the kernels, then the form of the transformation surface is seen.

$$z_N(t) = \sum_{j=1}^{n} H_j(t)\, x_j + \sum_{i=1}^{n} \sum_{j=i}^{n} H_{ij}(t)\, x_i\, x_j + \sum_{i=1}^{n} \sum_{j=i}^{n} \sum_{k=j}^{n} H_{ijk}(t)\, x_i\, x_j\, x_k$$

$$+ \ldots\ldots + \underbrace{\sum_{i=1}^{n} \cdots \sum_{q=}^{n}}_{N \text{ sums}} H_{i \ldots q}(t)\, x_i \ldots x_q \tag{23}$$

Note that if $N = 1$ (linear system) the transformation surface is a hyperplane, for $N = 2$ a quadric surface, or in general, an N^{th} order polynomial type surface.

Φ Learning Machine[13]

The term " Φ learning machine" refers to the generic form of a pattern recognition device. The general block diagram of this device is shown in Fig. (3). The first operation is a transformation of the input vector (pattern) $\underline{X}$ into a vector $\underline{F}$ in Φ space. Vector $\underline{F}$ is a set of linearly independent functions $f_i(\underline{X})$. The coordinates in Φ space are a set of functions which span the space. Specific examples of Φ functions are: ($\underline{X}$ has d dimensions)

1. Linear functions $f_i(\underline{X}) = x_i \qquad i = 1, \ldots d$
2. Quadric functions $f_i(\underline{X})$ has the form $x_k^{\,n}\, x_\ell^{\,m}$

   $k, \ell = 1, \ldots, d \qquad \text{and} \qquad n, m = 0 \text{ and } 1$

3. r^{th} order polynomial functions: $f_i(\underline{X})$ has the form

$$x_{k_1}^{n_1} \; x_{k_2}^{n_2} \; . \; . \; . \; x_{k_r}^{n_r} \qquad \text{for}$$

$$k_1, k_2, \ldots, k_r = 1, \ldots, d \qquad \text{and} \qquad n_1, n_2, \ldots, n_r = 0 \text{ and } 1$$

If the original vector $\underline{X}$ was defined in a d dimensional space, the vector $F(\underline{X}) = \left\{ f_1(\underline{X}), f_2(\underline{X}), \ldots, f_M(\underline{X}) \right\}$ is defined in an M dimensional space where

$$M = \begin{pmatrix} d + r \\ r \end{pmatrix} - 1 \qquad r = \text{order of polynomial} \tag{24}$$

The second operation is a linear summation of the functions $f_i(\underline{X})$. The function

$$\Phi(\underline{X}) = \sum_{i=1}^{m} w_i \, f_i(\underline{X}) \tag{25}$$

represents a hyperplane in Φ space, and an r^{th} order polynomial surface in the original $\underline{X}$ space.

In the transformed space, or Φ space, the separating surface $\Phi(\underline{X})$ is adjustable by an iterative error correcting algorithm. Consequently the use of a transformation to a nonlinear space considerably eases the conceptual and computational difficulties in achieving a given separating hypersurface in the original linear space. The general procedure is quite similar to that of multiple regression,[13] where a least squares fit to a given surface is achieved.

Equivalence of Volterra Series and Φ Machine

The equivalence between a volterra series expansion for a nonlinear system and a Φ learning machine will now be demonstrated. The case that will be considered is the time invariant expansion (13).

$$z_N(t) = \sum_{j=1}^{n} H_j x_j + \sum_{i=1}^{n} \sum_{j=i}^{n} H_{ij} x_i x_j + \sum_{i=1}^{n} \sum_{j=i}^{n} \sum_{k=j}^{n} H_{ijk} x_i x_j x_k + \cdots$$

$$+ \sum_{i=1}^{n} \cdots \sum_{q=}^{n} H_{i \ldots q} x_i \cdots x_q \tag{26}$$

This represents an N^{th} order polynomial hypersurface in $\underline{X}$ space. On the other hand, this is exactly the same type of surface implemented by an N^{th} order Φ machine

$$\Phi(x) = \sum_{j=1}^{d} w_j x_j + \sum_{i=1}^{d} \sum_{j=i}^{d} w_{ij} x_i x_j + \cdots + \sum_{i=1}^{d} \cdots \sum_{q=}^{d} w_{i \ldots q} x_i \ldots x_q \tag{27}$$

Therefore, the techniques used in determining the separating surfaces for pattern recognition can be directly applied to the problem of determining the transformation surface for nonlinear systems. This is indeed a useful analogy, as the techniques for pattern recognition are well developed.

Training Procedure

The training procedure uses an error-correcting algorithm to train the Φ-machine. This algorithm iteratively adjusts the weight vector of the linear portion of the Φ-machine based upon the normal operating record of the system.[14]

The algorithm uses the following nomenclature:

y_i = output of the system at the i^{th} iteration.

$\underline{X}_i$ = input vector to the system at the i^{th} iteration.

$z_i = \Phi(\underline{X}_i)$ = output of Φ-machine at the i^{th} iteration.

$\underline{W}_i$ = weight vector of Φ-machine at the i^{th} iteration.

$\underline{F}(\underline{X}) = \text{col}\left[f_1(\underline{X}), \ldots, f_M(\underline{X})\right]$

$\underline{F}_i = \underline{F}(\underline{X}_i)$ = output of Φ -processor at the i^{th} iteration

$||\underline{F}_i|| = \underline{F}_i^T \underline{F}_i$ (squared Euclidian norm).

α = convergence factor of algorithm.

The sequences of steps used with the algorithm is as follows:

1. Set the initial weight vector. A zero weight vector is adequate.
2. Determine y_i and $\underline{X}_i$.
3. Generate $\underline{F}_i$ and z_i.

$$z_i = \underline{W}_i \cdot \underline{F}_i = \underline{W}_i^T \underline{F}_i \tag{28}$$

4. Calculate new weight vector using the following error-correcting algorithm.[3,14]

$$\underline{W}_{i+1} = \underline{W}_i + \frac{\alpha(y_i - z_i)}{||\underline{F}_i||} \underline{F}_i \tag{29}$$

where $0 < \alpha < 2$.

5. Repeat starting with step 2.

This procedure generates a sequences of $\underline{W}_i$'s whose components will converge to the kernels of the Volterra representation of the system under certain conditions.

The system shown in Fig. (4) will be used to illustrate this algorithm. The input-output relationship for this system is

$$\begin{aligned} y\left[NT\right] &= 17\, x\left[NT\right] + 17\, x\left[(N-1)T\right] \\ &= (17,\ 17)\begin{pmatrix} x\left[NT\right] \\ x\left[(N-1)T\right] \end{pmatrix} \end{aligned} \tag{30}$$

The correct weight vector is col (17, 17). Assume an input sequence such as

N	0	1	2	3	4	5	6	7	8
x [NT]	0	1	1	1	-4	2	-1	0	-1

Table 1 lists the results at each iteration. The sequence of weight vectors is plotted in Fig. (5). This sequence converges to the correct weight vector. The weight vector error is the difference between the correct weight vector and the Φ-machine weight vector. The sequence of weight vector errors is also plotted in Fig. (5). The difference between two successive weight vectors, $\underline{W}_{i+1} - \underline{W}_i$, is the projection of the weight vector error, $\underline{E}_i$, on the input vector $\underline{X}_i$. Therefore the weight vector error cannot increase. The weight vector error will remain the same, if two successive input vectors are linear dependent. The input vectors for $i = 2, 3$ and for $i = 5, 6$ are examples of this.

Convergence

Consider the system of Fig. (6). The output of the system is corrupted by additive noise N. $\underline{W}_x$ is the weight vector whose components are the kernels of Eq. (26). Equation (26) can be written as:

$$z_N(t) = \underline{W}_* \cdot \underline{F}(\underline{X}) = \underline{W}_*^T \, \underline{F}(\underline{X}) \tag{31}$$

For systems that are approximated by Eq. (26) the truncated terms of their Volerra representations are combined with the additive noise N. Therefore N_i is the additive noise corresponding to z_i.

The Φ-machine learns the mean of the noise along with the steady state value of the process. Thus, the noise N_i has zero mean but otherwise arbitrary characteristics. Then,

$$y_i = \underline{W}_*^T \, \underline{F}_i + N_i \tag{32}$$

Now the convergence properties of the system in Fig. (4) will be studied. Let $\underline{E}_i$ be the weight vector error.

$$\underline{E}_i = \underline{W}_* - \underline{W}_i \tag{33}$$

Define convergence of the learning machine as the norm of $\underline{E}_i$ approaching zero as i increases without bound. From Eqs. (28), (29), (32) and (33)

$$\underline{W}_{i+1} = \underline{W}_i + \frac{\alpha}{\| \underline{F}_i \|} (\underline{W}_*^T \underline{F}_i + N_i - \underline{W}_i^T \underline{F}_i) \underline{F}_i \tag{34}$$

$$\underline{E}_{i+1} = \underline{E}_i - \frac{\alpha}{\| \underline{F}_i \|} \underline{F}_i \underline{F}_i^T \underline{E}_i - \frac{\alpha N_i}{\| \underline{F}_i \|} \underline{F}_i$$

Let

$$A_i = I - \frac{\alpha}{\| \underline{F}_i \|} \underline{F}_i \underline{F}_i^T$$

and

$$b_i = \frac{\alpha N_i}{\| \underline{F}_i \|}$$

Then

$$\underline{E}_{i+1} = A_i \underline{E}_i - b_i \underline{F}_i$$

Taking the norm of $\underline{E}_{i+1}$ gives

$$\| \underline{E}_{i+1} \| = \underline{E}_i^T A_i^2 \underline{E}_i - 2 b_i \underline{F}_i^T A_i \underline{E}_i + b_i^2 \| \underline{F}_i \| \tag{35}$$

$$A_i^2 = I - \frac{2\alpha}{\| \underline{F}_i \|} \underline{F}_i \underline{F}_i^T + \frac{\alpha^2}{\| \underline{F}_i \|^2} \underline{F}_i \underline{F}_i^T \underline{F}_i \underline{F}_i^T$$

$$= I - \frac{\alpha(2-\alpha)}{\| \underline{F}_i \|} \underline{F}_i \underline{F}_i^T$$

$$\underline{F}_i^T A_i \underline{E}_i = \underline{F}_i^T \underline{E}_i - \frac{\alpha}{\| \underline{F}_i \|} \underline{F}_i^T \underline{F}_i \underline{F}_i^T \underline{E}_i$$

$$= (1-\alpha) \underline{F}_i^T \underline{E}_i$$

Separating the terms containing N_i gives

$$\| \underline{E}_{i+1} \| = \left\{ 1 - \frac{\alpha(2-\alpha)}{\|\underline{F}_i\| \, \|\underline{E}_i\|} (\underline{F}_i^T \underline{E}_i)^2 \right\} \| \underline{E}_i \| + \frac{\alpha N_i}{\|\underline{F}_i\|} (\alpha N_i + 2(\alpha-1) \underline{F}_i^T \underline{E}_i) \tag{36}$$

Let

$$c_i = \frac{\alpha(2-\alpha)}{\|\underline{F}_i\| \, \|\underline{E}_i\|} (\underline{F}_i^T \underline{E}_i)^2 \tag{37}$$

where $0 \leq c_i \leq 1$ since $0 < \alpha < 2$. Let

$$d_i = \frac{\alpha N_i}{\|\underline{F}_i\|} (\alpha N_i + 2(\alpha-1) \underline{F}_i^T \underline{E}_i) \tag{38}$$

Then

$$\begin{aligned} \| \underline{E}_{i+1} \| &= (1-c_i) \, \| \underline{E}_i \| + d_i \\ &= \prod_{j=1}^{i} (1-c_j) \| \underline{E}_1 \| + \sum_{j=1}^{i-1} d_j \prod_{k=j+1}^{i} (1-c_k) + d_i \end{aligned} \tag{39}$$

Note that

$$0 \leq \prod_{j=1}^{i} (1-c_j) \leq 1 \qquad \text{since} \qquad 0 \leq c_i \leq 1 \tag{40}$$

Consider the case where $N_i = 0$ for all i.

$$\| \underline{E}_{i+1} \| = \prod_{j=1}^{i} (1-c_j) \| \underline{E}_1 \| \qquad \text{since} \qquad d_i \equiv 0 \tag{41}$$

Eliminate the possibility that c_i ever equals one because it is quite improbable and c_i equaling one implies immediate convergence, $\| \underline{E}_{i+1} \| = 0$. Thus, one has only to consider c_i less than one. Then a necessary and sufficient condition for convergence of the limit of Eq. (41) to zero as i increases is that the sum of the c_i diverges.

$$\sum_{j=1}^{\infty} c_j = \infty \tag{42}$$

Therefore, the input vector must probe the input vector space so that an infinity of c_i are nonzero. Also, the c_i cannot approach zero too quickly. By Eqs. (37) and (42)

$$\sum_{j=1}^{\infty} \frac{(\underline{F}_i^T \underline{E}_i)^2}{\|\underline{F}_i\| \, \|\underline{E}_i\|} = \infty \tag{43}$$

Thus, the sequence of $\underline{F}_i$'s must probe its vector space in all directions infinitly often in the training sequence. Therefore, the sequence of $\underline{X}_i$'s must probe its vector space in both magnitude and direction.

Consider the case where the noise N_i is nonzero and assume that condition (42) is satisfied. If N_i does not approach zero as i increases the error vector norm cannot approach zero.

Let $\overline{\|\underline{E}_i\|}$ be the conditional expected value of the error norm given the sequence of $\underline{F}_i$'s. Only d_i depends on N_i in Eq. (39). The conditional expected value of d_i is

$$\bar{d}_i = \frac{\alpha^2}{\|\underline{F}_i\|} \overline{N_i^2} \tag{44}$$

This exists assuming $\underline{F}_i \neq \underline{0}$ and that $\overline{N_i^2}$, the variance of the noise, exists. Taking the conditional expected value of Eq. (39) gives

$$\overline{\|\underline{E}_{i+1}\|} = \prod_{j=1}^{i} (1-c_j) \, \|\underline{E}_1\| + \sum_{j=1}^{i-1} \bar{d}_j \prod_{k=j+1}^{i} (1-c_k) + \bar{d}_i \tag{45}$$

The Φ-machine is converging if $\overline{\|\underline{E}_i\|}$ decreases as i increases otherwise it is diverging. Assume that $\bar{d}_i$ is bounded for all i and that there are only a finite number of c_i less than a preset positive constant.

$$\bar{d}_i \leq \bar{d}_{max}, \quad \text{all } i \qquad 0 \leq c_i < \epsilon \tag{46}$$

for $N-1$ values of i.

A bound on $\overline{\|\underline{E}_i\|}$ is found by operating on Eq. (45).

$$\overline{\|\underline{E}_{i+1}\|} \le \prod_{j=1}^{i} (1-c_i)\ \|\underline{E}_1\| + \bar{d}_{max} \left[1 + \sum_{j=1}^{i-1} \prod_{k=j+1}^{i} (1-c_k) \right]$$

$$\le \prod_{j=1}^{i} (1-c_i)\ \|\underline{E}_1\| + \bar{d}_{max} \left[1 + N-1 + \sum_{j=1}^{i-N} \prod_{k=j+1}^{i+1-N} (1-\epsilon) \right]$$

$$\le \prod_{j=1}^{i} (1-c_i)\ \|\underline{E}_1\| + \bar{d}_{max} \left[N + \sum_{j=1}^{i-N} (1-\epsilon)^j \right]$$

Taking the limit as $i \to \infty$

$$\|\underline{E}_\infty\| \le \bar{d}_{max} \left[N + \frac{1}{\epsilon} \right] \tag{47}$$

This is a very conservative bound. However, it is proportional to the variance of the noise for stationary noise characteristics. Therefore the error norm will decrease as i increases if the noise variance is not too large.

Tests

The identification method was studied by simulation on an IBM 360 Model 50. All noise signals were generated by pseudo-random number generators. The cubic system shown in Fig. (7) was used in tests of the identification procedure. The input, $x(t)$, used was correlated guassian noise passed through a sampler and a zero order hold. The Φ-machine was trained to the average output over the sampling interval. The sampling interval used for all tests was 1 sec.

Polynomial terms of greater than third order are not needed for the system shown in Fig. (7). This system also has a null steady state output for a null input. The settling time of the system was approximated as 10 sec. The additive noise was assumed to have zero mean. Thus, the weight vector has 285 components.

The correct weight vector, $\underline{W}_*$, was determined by using special input sequences. These input sequences were chosen so that the corresponding $\underline{F}_i$'s would form a linearly independent set. The corresponding outputs of the system were used to form the following equations.

$$y_i = \underline{W}_*^T \underline{F}_i \qquad i = 1, 285$$

These equations were solved for $\underline{W}_*$.

Two types of tests were made. In the first type the Φ-machine was trained to the average output over the sampling interval of the cubic system. In the second type the Φ-machine was trained to the output of the system shown in Fig. (6). The $\underline{W}_*$ used was the one for the cubic system. The N_i was sampled uncorrelated gaussian noise with zero mean. Only the variance of the input was varied in both tests. Variances of ten, one, and one-tenth were used. The input noise had zero mean and exponential correlation, $\rho = 0.707$.

The following error measures were used. The normalized weight error was used to judge the extent of convergence of the Φ-machine. The normalized weight error is given by

$$\|E_i\| = \frac{\|\underline{W}_* - \underline{W}_i\|}{\|\underline{W}_*\|}$$

The R.M.S. error between system output and Φ-machine output was used to illustrate the performance of the Φ-machine as a model for the system. This error measure is given by

$$e(j, N) = \sqrt{\frac{1}{N}\sum_{i=j}^{j+N} (y_i - z_i)^2}$$

Graphs of the normalized weight error verses the number of iterations are in Fig. (8) for the cases where the input variance was ten. The identification procedure diverged for the second type of test when additive uncorrelated noise with a variance of fifty was used. However, this procedure convergedfor the first type of test where the additive noise was correlated and had a variance of 176. Thus, the identification procedure can withstand greater noise variances, if the noise is correlated.

The results from 1000 iterations are in Tables 2-4. λ is an approximate exponential convergence rate. An exponential function, $e^{-\lambda i}$, was fitted to the curve of normalized weight error verses the number of iterations for the last forty iterations. In the cases where the variance of the input was one-tenth, the identification procedure did not obtain any significant amount of convergence. The rms error did not decrease significantly in these cases. Thus, the input must not be to small. The amount of decrease in rms error is well correlated with the amount of convergence. The cases that did not converge seem to approach a constant weight vector after a large initial error. The convergence rates are all the same order of magnitude for the cases where the convergence was good.

Conclusions

This paper has presented a new method for system identification. Based upon the relaxation technique used in pattern recognition, this method of identification produces the hypersurface of the input-output transformation. It is shown that the describing hypersurface is equivalent to a volterra series representation of the system and that the identification technique produces the kernels of the volterra representation.

The relaxation (or error-correcting) technique will converge to the correct solution, even when the measurements are corrupted by noise. Any final error in the weight vectors (constants of the hypersurface) is shown to be bounded by a constant which is directly proportional to the variance of the measurement noise. In addition, experimental results have shown that the range of acceptable noise variance under which the system will converge can be greatly increased if the noise is correlated. Both theoretical and experimental results are presented to support these conclusions.

Acknowledgement

This study was supported by the National Aeronautics and Space Administration under Research Grant NGR-33-018-014.

References

1. Roy, R. and R. W. Miller, Nonlinear Process Identification Using Decision Theory, IEEE Transactions on Automatic Control, Oct. 1964, Vol. AC-9, No. 4, pp. 538-540.

2. Roy, R. and C. Schley, Process Identification Using a Mode Learning Machine, Proceedings of the 7th Joint Automatic Control Conference, Univ. of Washington, Seattle, Washington, Aut. 17-19, 1966, pp. 639-648.

3. Nilsson, N. J., Learning Machines, McGraw-Hill Book Co., N.Y., 1965.

4. Volterra, V., Theory of Functionals and of Integral and Integro-Differential Equations, Dover Publications, New York (1959).

5. Smets, H. B., Analysis and Synthesis of Nonlinear Systems, IRE Transactions Professional Group on Circuit Theory, CT-7, No. 4, pp. 459-469, Dec. 1960.

6. Barrett, J. F., The Use of Functionals in the Analysis of Nonlinear Physical Systems, Statistical Advisory Unit, Report No. 1/57, Ministry of Supply, Great Britain, 1957.

7. VanTrees, H. L., Synthesis of Optimum Nonlinear Control Systems, The M.I.T. Press, Cambridge, Mass., 1962.

8. Shen, D. W. C., Artificial Intelligence in a Hierachy of Nonlinear Systems, IEEE Special Publication S-142 Artificial Intelligence, Jan. 1963, pp. 18-30.

9. Brilliant, M. B., Theory of the Analysis of Nonlinear Systems, Technical Report 345, Research Laboratory of Electronics, MIT, March 1958.

10. Wiener, N., Nonlinear Problems in Random Theory, The Technology Press of M.I.T., Cambridge, Mass., and John Wiley and Sons, New York, 1949.

11. Cameron, R. H., and Martin, W. T., The Orthogonal Development of Nonlinear Functionals in Series of Fourier-Hermite Functionals, Annals of Mathematics, 48, No. 2, pp. 385-389, April 1947.

12. Zaborsky, J., and Humphrey, W., Control Without Model or Plant Identification, Proceedings of the 5th Joint Automatic Control Conference, Stanford University, Stanford, California, 1964, pp. 361-366.

13. Ralston, A., and Wilf, H., Mathematical Methods for Digital Computers, "Multiple Regression Analysis," by M. A. Efroymson, John Wiley and Sons, Inc., 1960, pp. 191-203.

14. Nagumo, J., and Noda, A., A Learning Method for System Identification, Proceedings of the 1966 National Electronics Conference, Vol. 22, pp. 584-589.

TABLE 1

i	1	2	3	4	5	6	7	8
y_i	17	34	34	-51	-34	17	-17	-17
z_i	0	17	34	-93.5	-13	17	-13.6	-14.2
$\underline{X}^T$	1, 0	1, 1	1, 1	-4, 1	2, -4	-1, 2	0, -1	-1, 0
$\underline{W}_i^T$	0, 0	17, 0	25.5, 8.5	25.5, 8.5	15.5, 11	14.2, 13.6	14.2, 13.6	14.2, 17
$c_i\underline{X}_i^T$	17, 0	8.5, 8.5	0, 0	-10, 2.5	-1.3, 2.6	0, 0	0, 3.4	2.8, 0
$\underline{W}_{i+1}^T$	17, 0	25.5, 8.5	25.5, 8.5	15.5, 11	14.2, 13.6	14.2, 13.6	14.2, 17	17, 17

where $\underline{F}_i = \underline{X}_i$, $\alpha = 1$, and $c_i = (y_i - z_i)/||\underline{X}_i||$

TABLE 2

Input Noise Variance = 10

Type of test	First	Second	Second	Second	Second	Second	Second
Variance of additive noise	176	0.0	0.5	1.0	5.0	10.0	50.0
e(1, 40)	141.1	185.2	187.8	190.4	212.5	242.5	512.0
e(960, 40)	16.9	21.9	21.8	21.9	29.4	46.3	211.4
e(1, 1000)	52.9	65.5	66.0	66.7	74.0	88.0	253.7
$\|\|E_{1000}\|\|$	.3863	.2983	.2979	.2981	.3151	.3698	1.189
$\lambda \times 1000$	0.3836	1.768	1.738	1.606	1.163	.5173	*

*Divergent case

TABLE 3

Input Noise Variance = 1.0

Type of test	First	Second	Second	Second	Second	Second
Variance of additive noise	.176	0.0	.01	.05	.1	.5
e(1, 40)	.2152	.2569	.2625	.3012	.3749	1.201
e(960, 40)	.0201	.0306	.0396	.1352	.2668	1.331
e(1, 1000)	.0707	.0825	.0876	.1625	.2900	1.384
$\|\|E_{1000}\|\|$	.3810	.3229	.3532	.8107	1.529	7.505
λ x 1000	1.687	1.858	1.106	-1.45	*	*

*Divergent case

TABLE 4

Input Noise Variance = 0.1

Type of test	First	Second	Second	Second	Second
Variance of additive noise	.000176	0.0	.001	.005	.01
e(1, 40)	.00036	.00043	.00173	.00820	.01634
e(960, 40)	.00034	.00038	.00161	.00809	.01622
e(1, 1000)	.00050	.00057	.00159	.00756	.01512
$\lVert \mathbf{E}_{1000} \rVert$	.9571	.9492	.9725	1.276	1.901
λ x 1000	1.557	.188	-.352	*	*

*Divergent case

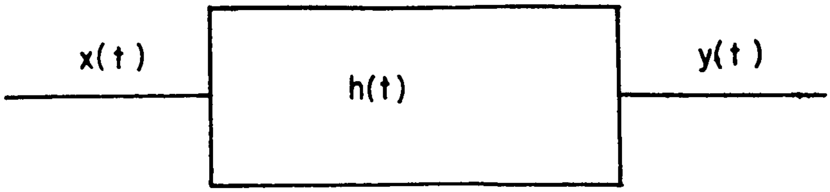


FIGURE 1

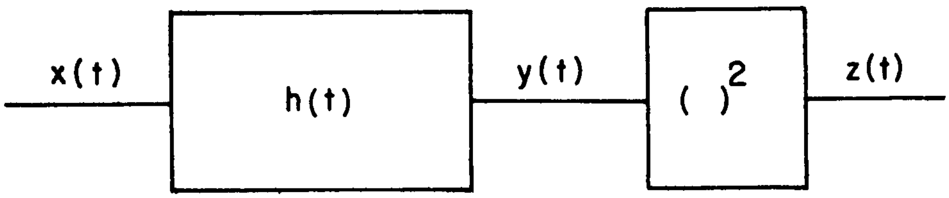
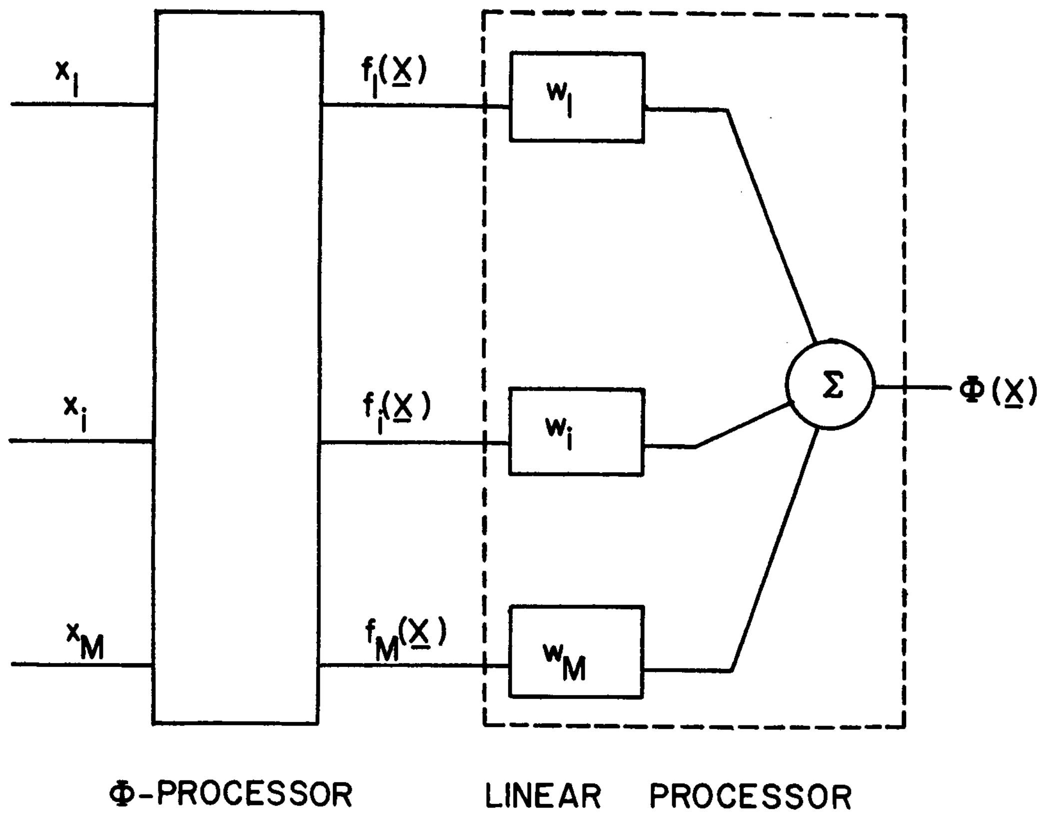


FIGURE 2



Φ- MACHINE

FIGURE 3

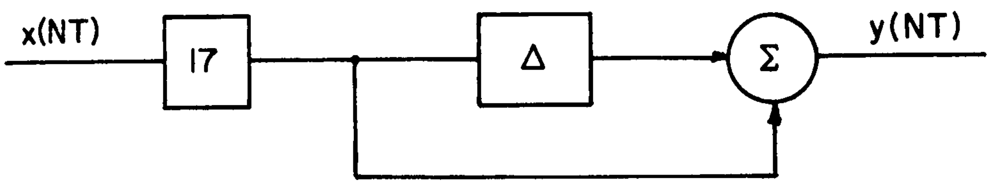
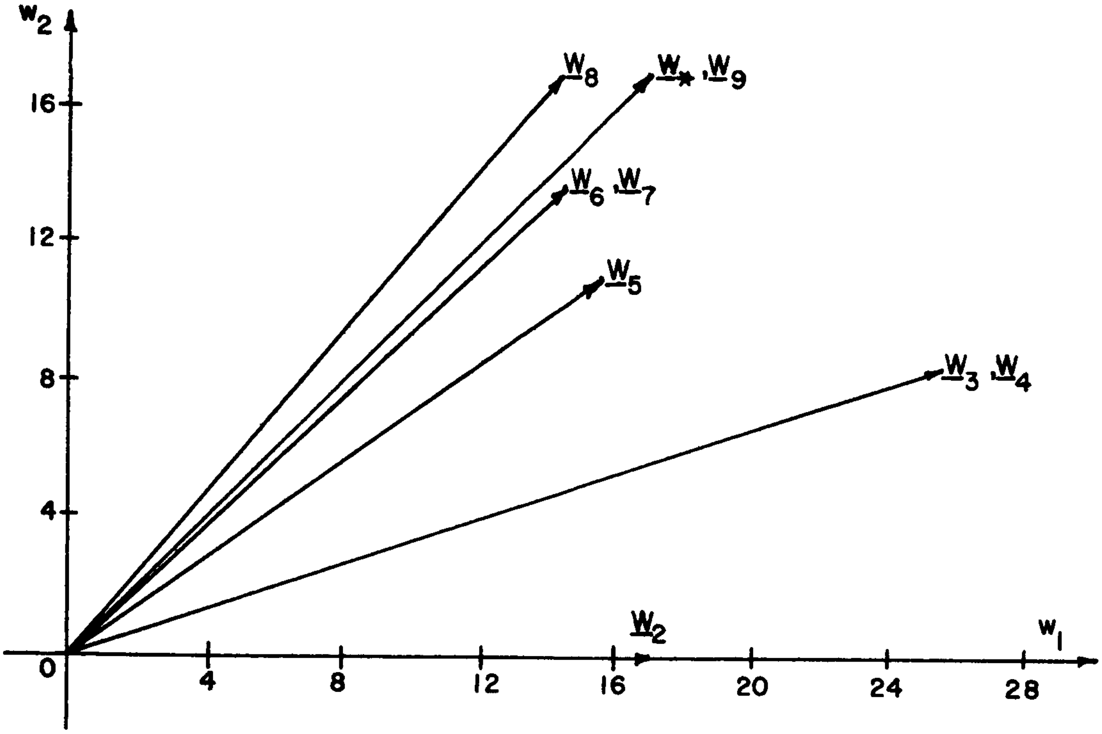
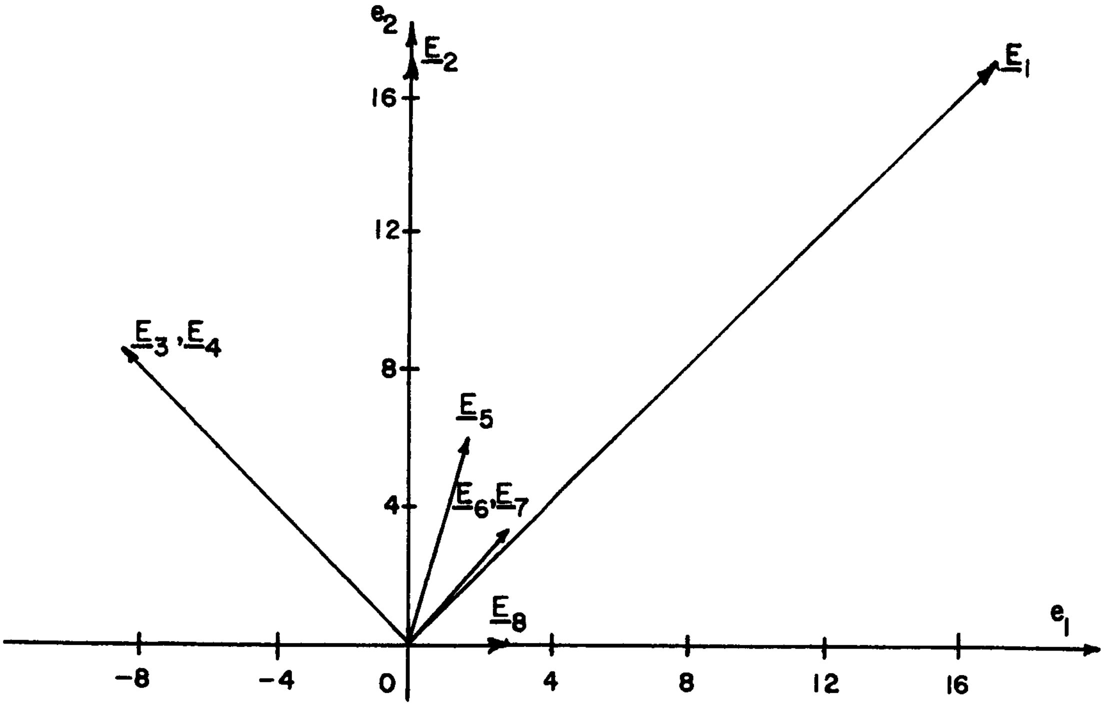


FIGURE 4



WEIGHT VECTOR SEQUENCE



WEIGHT VECTOR ERROR SEQUENCE

FIGURE 5

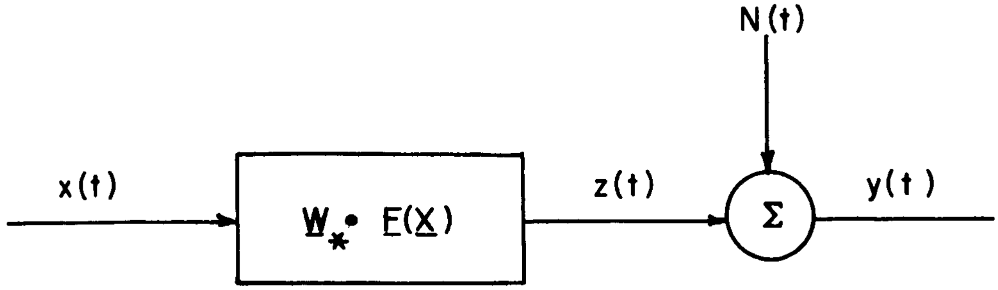
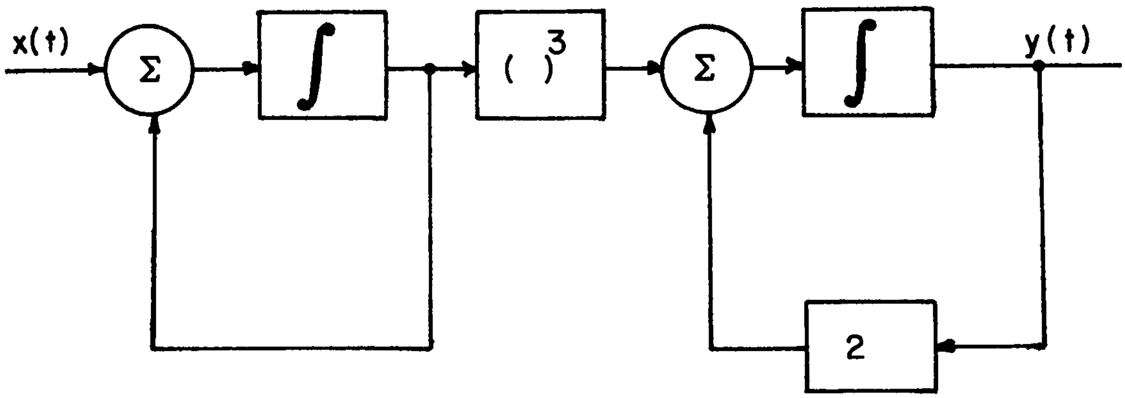


FIGURE 6



CUBIC SYSTEM

FIGURE 7

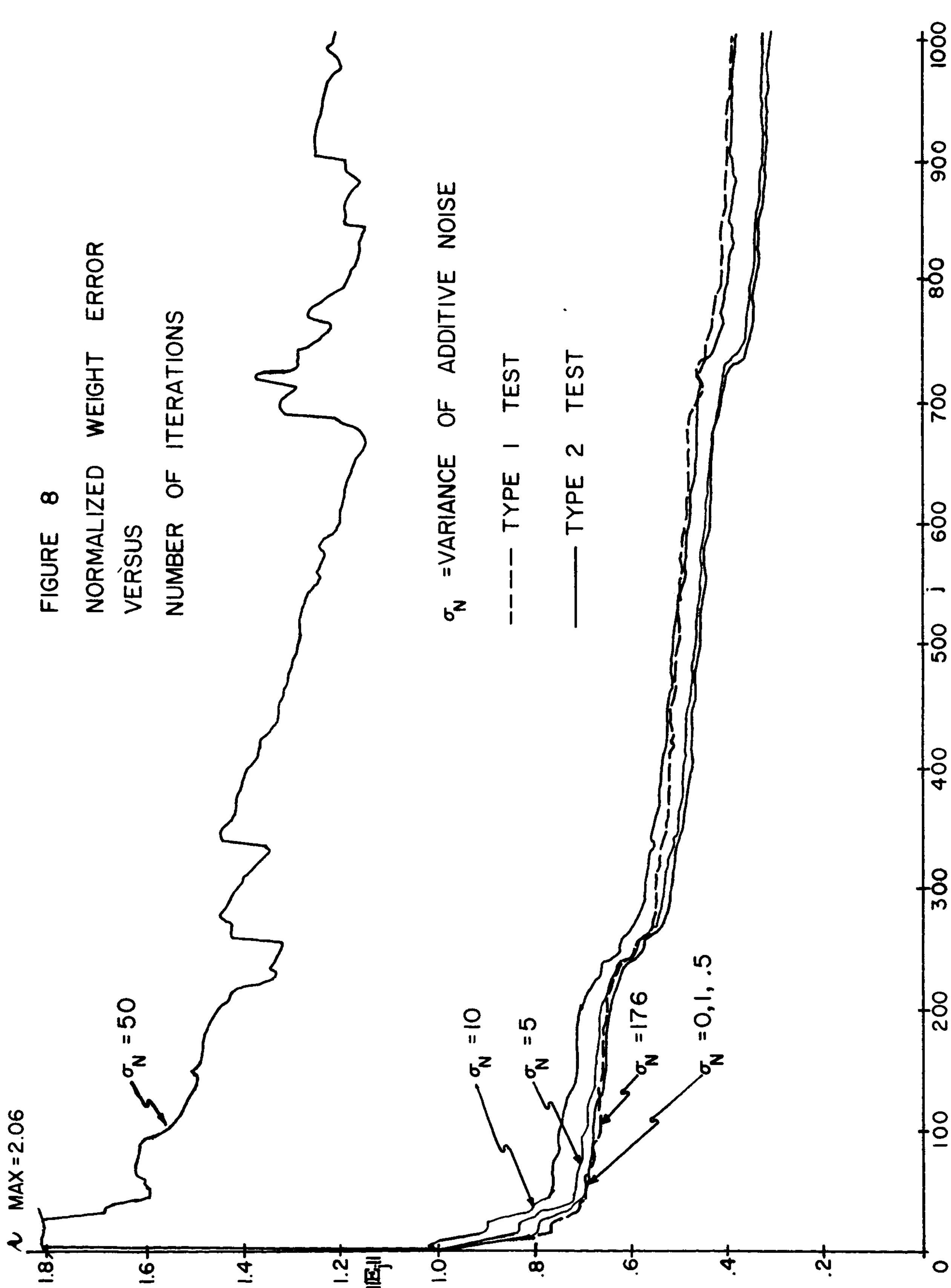


FIGURE 8
NORMALIZED WEIGHT ERROR
VERSUS
NUMBER OF ITERATIONS
σ_N =VARIANCE OF ADDITIVE NOISE
---- TYPE 1 TEST
—— TYPE 2 TEST
MAX=2.06
σ_N =50
σ_N =10
σ_N =5
σ_N =176
σ_N =0,1, .5
$\|E_j\|$
j
1.8
1.6
1.4
1.2
1.0
.8
.6
.4
.2
0
100
200
300
400
500
600
700
800
900
1000

10-5-67

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Notes, and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION DIVISION

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546