LINEAR SYSTEMS IDENTIFICATION ALGORITHMS FOR A MINICOMPUTER

by

HEINRICH ZUERCHER

Diploma in El. Eng., Federal Institute of Technology, Switzerland, 1972

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

in the Department of
Electrical Engineering

We accept this thesis as conforming to the required standard

Research Supervisor

Members of Committee

Head of Department

Members of the Department of
Electrical Engineering
THE UNIVERSITY OF BRITISH COLUMBIA

November, 1974

Department of EL. ENG.

The University of British Columbia
Vancouver 8, Canada

Date NOV 15 1974

ABSTRACT

Two methods for identifying multiple-input mutliple-output linear time-invariant discrete systems from input-output data are derived. The selector matrix principle of Gopinath is used and two special classes of selector matrices that lead to canonical system models are introduced. Very general input sequences can be applied. However, a few restrictions exist for the initial system state. The input matrix is identified column-wise. Both methods are considerably better than Gopinath's method in terms of storage requirements and numerical accuracy. Tests on a large computer are performed. One method is implemented on a minicomputer with good results.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENT

I wish to thank Dr. E.V. Bohn, my supervisor, for his advice, guidance and interest during the research and writing of my thesis.

I am thankful to Mr. Dave Holmes for his assistance with the hardware work.

The financial support received in the form of U.B.C. Graduate Fellowships from 1972 to 1974 and under the National Research Council Grant 67-3134 is gratefully acknowledged.

I. INTRODUCTION

Aim of Thesis

Recent advances in the capabilities of process computers and corresponding digital interface equipment have made the application of "modern" control techniques much more feasible. "Modern" design methods for multi-input multi-output linear time-invariant systems require a state-space model of the system and the problem of identifying such models from experimental data has therefore been the subject of intensive research in recent years. Several identification methods have been proposed and tested on large computers. They all require extensive data processing, which makes their implementation on minicomputers inefficient if not impossible. The aim of this thesis is to develop and test on-line identification algorithms that can be realized efficiently on minicomputers.

Background

It is essential for the practical usefulness of an identification method that the class of applicable input signals is not restricted to particular inputs such as impulse function, step function or white noise. The ideal case would certainly be the identification from normal operating records. Another practically relevant requirement regards the minimality of the number of significant parameters of the model and this requires the choice of a suitable canonical form, whenever this is possible. Moreover the simplicity of the link between the selected state-space representation and the set of input-output equations used for the identification heavily conditions the required amount of computation.

The above problems have been solved for single-input single-output or multi-input single-output systems [9] but no unified approach for multi-input multi-output systems that satisfies all previous requirements can be found in the literature. So far a number of methods that allow the use of general input-output records have been suggested: Gopinath [1], Budin [2], Bonivento and Guirdozi [3], Ackermann [4] and Bingulac and Djorovic [8]. Gopinath introduces the principle of a data selector matrix which enables the identification method to be formulated concisely and in a manner suitable for digital computation. However, the models found generally have no canonical structure. The computational part of this procedure, especially the determination of a selector matrix, has been improved by Budin. Bonivento and Ackermann both introduce a particular canonical system model and then identify the reduced number of model parameters using the corresponding input-output equations. However, the selection and processing of input-output data is more complicated than with Gopinath's selector matrix. Bingulac's procedure leads to a model in Jordan canonical form and requires the determination of system eigenvalues i.e. polynomial roots. This makes the treatment of noisy data very difficult, which is a serious drawback.

Outline of work

An examination of Gopinath's method in chapter II shows that a slightly more general definition of the selector matrix allows us to consider both Ackermann's and Bonivento's approach as special cases of Gopinath's selector matrix formulation. The generality of Gopinath's procedure justifies its further investigation. Two modified versions of Gopinath's method are proposed in chapter III and the implementation of one such method on a minicomputer is described in chapter IV. Finally chapter V presents experimental results and examples whereas chapter VI contains the conclusions.

II. GOPINATH'S METHOD USING SPECIAL SELECTOR MATRICES

2.1 Problem Statement

A linear time-invariant multiple-input multiple-output system can be represented by the state equations

$$x(k + 1) = A\ x(k) + B\ u(k) \tag{1}$$

$$y(k) = H\ x(k) \tag{2}$$

where x is an n x 1 state vector, y is an ℓ x 1 output vector, u is an m x 1 input vector, A is an n x n state transition matrix, B is an n x m input matrix and H is an ℓ x n output matrix. The system is thus specified by the triplet (A,B,H). (A,B,H) is assumed to be completely observable and controllable because otherwise it is impossible to identify the system from input-output data.

The problem is to find a model $(\hat{A},\hat{B},\hat{H})$ of the same dimension as (A,B,H) from a sequence of inputs u(k) and outputs y(k) such that $(\hat{A},\hat{B},\hat{H})$ simulates the input-output behaviour of (A,B,H). It is clear that the number of correct models $(\hat{A},\hat{B},\hat{H})$ is infinite and that there are input sequences u(k) which will not be sufficient to uniquely specify any realization $(\hat{A},\hat{B},\hat{H})$.

The use of a selector matrix for the processing of data splits the problem into two parts:

1) The finding of a selector matrix. This is equivalent to determining order and structure of $(\hat{A},\hat{B},\hat{H})$.

2) The identification of all parameters in $(\hat{A},\hat{B},\hat{H})$.

This thesis deals with an efficient on-line computer solution of problem part 2) and a selector matrix is, therefore, always assumed to be known. Two particular classes of selector matrices are introduced in the next section and then used throughout the thesis.

2.2 Two Special Classes of Selector Matrices

A more general class of selector matrices than the one introduced in [1] and used in [2] is defined below.

Definition 1:

$\mathcal{S}$ denotes the set of n x d matrices (n ≤ d) with the following properties:

1) $S = \{s_{ij}\}$ where $s_{ij} = 0$ or 1 (3)

2) $\forall i$, $s_{ij} = 1$ for one and only one j, say j_i (4)

3) $j_i \neq j_k$ for $i \neq k$ $\quad i, k = 1,2, \ldots, n$ (5)

This definition implies that when premultiplying a d x p matrix A with S the resulting n x p matrix SA consists of n of the rows of A ordered in accordance with the unit row vectors of S. Note that with selector matrices as defined in [1] and [2] the n rows in SA are always ordered as they are in A. However, with selector matrices as defined in Definition 1, the n rows of SA can be ordered in any possible way. We can now define the following special class of selector matrices.

Definition 2:

$\mathcal{S}_{s2}$ denotes the set of n x d matrices with the following properties:

1) $\mathcal{S}_{s2} \subset \mathcal{S}$ (6)

2) d is an integral multiple of ℓ (7)

3) $S'_{s2} = [S'_1 \; S'_2 \; \cdots \; S'_\ell]$ (8)

where $S_k = \{s_{k,ij}\}$ are n_k x d matrices with

$$j_i = k + (i - 1)\,\ell \qquad i = 1, \ldots n_k$$
$$k = 1, \ldots \ell$$

and j_i is defined as in (4)

4) $n_k \geq 1$ for $k = 1, \ldots \ell$ (9)

An example of a selector matrix S_{s2} is

$$S_{s2} = \begin{bmatrix} 1&0&0&0&0&0 \\ 0&0&1&0&0&0 \\ 0&0&0&0&1&0 \\ 0&1&0&0&0&0 \\ 0&0&0&1&0&0 \end{bmatrix} \qquad \text{where } n = 5, \ \ell = 2, \ d = 3*\ell = 6, \quad n_1 = 3, \ n_2 = 2$$

with

$$S_1 = \begin{bmatrix} 1&0&0&0&0&0 \\ 0&0&1&0&0&0 \\ 0&0&0&0&1&0 \end{bmatrix} \qquad \text{and} \qquad S_2 = \begin{bmatrix} 0&1&0&0&0&0 \\ 0&0&0&1&0&0 \end{bmatrix}$$

The advantage gained from the application of a selector matrix of type 2 is demonstrated in section 2.3. The use of a selector matrix of type 3 as defined below will be shown in section 2.4. However, it is not necessary for the understanding of the remainder of the thesis, except section 2.4, to read Definition 3 which is obtained by slightly changing (8) and (9) in Definition 2.

Definition 3:

1) $\mathcal{S}_{s3} \subset \mathcal{S}$ (10)

2) d is an integral multiple of ℓ (11)

3) $S'_{s3} = [S'_1 \ \ S'_2 \ \ldots . \ \ S'_h]$ (12)

where $S_k = \{s_{k,ij}\}$ are n_k x d matrices with

* $j_i = k + (i-1)\ell$ $\qquad i = 1, \ldots n_k$

$\qquad k = 1 \ldots . h \leq \ell$

and j_i is defined as in (4).

4) $n_k \geq 1$ for $k = 1, \ldots . h$ (13)

An example of a selector matrix S_{s3} is

$$S_{s3} = \begin{bmatrix} 1&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0 \end{bmatrix} \qquad \text{where } n = 5, \ \ell = 2, \ d = 5*\ell = 10, \quad n_1 = 5, \ n_2 = 0, \ h = 1$$

2.3 The Algorithm with Selector Matrix Type 2

Gopinath's derivation starts out from the assumption that a selector matrix S exists such that

$$S \begin{bmatrix} H \\ HA \\ \vdots \\ HA^{n^*-1} \end{bmatrix} = T \text{ where T is nonsingular} \tag{14}$$

where

n^* = observability index of (A,B,H)

$n^* \leq n$

Theorem 1 states that there always exists a selector matrix of type 2 that satisfies (14).

Theorem 1:

If the system (A,B,H) is completely observable and the matrix H has the rank ℓ, i.e. $\rho[H] = \ell$, then there exists an $S \in \mathcal{S}_{s2}$ of dimension $n \times \ell n^*$ such that

$$S \begin{bmatrix} H \\ HA \\ \vdots \\ HA^{n^*-1} \end{bmatrix} = T \quad \text{where T is nonsingular} \tag{15}$$

The proof for Theorem 1 is described in Appendix 1. The next step in the procedure is the introduction of a basis for the model $(\hat{A},\hat{B},\hat{H})$ such that

$$S \begin{bmatrix} \hat{H} \\ \hat{H}\hat{A} \\ \vdots \\ \hat{H}\hat{A}^{n^*-1} \end{bmatrix} = I_n \tag{16}$$

(14) and (16) imply directly that

$$\begin{aligned} \hat{A} &= T\,A\,T^{-1} \\ \hat{B} &= T\,B \\ \hat{H} &= H\,T^{-1} \end{aligned} \tag{17}$$

It is well known that a state transformation of (A,B,H) with the transformation matrix T described in (15) leads to a model $(\hat{A},\hat{B},\hat{H})$ with the following observable canonical form:

$$\hat{A} = \left[\begin{array}{ccc|ccc|c|ccc}
0 & 1 & 0 & 0 & \cdots & 0 & & 0 & \cdots & 0 \\
\vdots & \ddots & & \vdots & & & & \vdots & & \\
0 & & 1 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\
\multicolumn{3}{c|}{-\alpha_{11}-} & \multicolumn{3}{c|}{-\alpha_{12}-} & & \multicolumn{3}{c}{-\alpha_{1\ell}-} \\
\hline
0 & \cdots & 0 & 0 & 1 & 0 & & 0 & \cdots & 0 \\
\vdots & & & \vdots & \ddots & & & \vdots & & \\
0 & \cdots & 0 & 0 & & 1 & & 0 & \cdots & 0 \\
\multicolumn{3}{c|}{-\alpha_{21}-} & \multicolumn{3}{c|}{-\alpha_{22}-} & & \multicolumn{3}{c}{-\alpha_{2\ell}-} \\
\hline
& & & & \vdots & & & & & \\
\hline
0 & \cdots & 0 & 0 & \cdots & 0 & & 0 & 1 & 0 \\
\vdots & & \vdots & \vdots & & & & \vdots & \ddots & \\
0 & \cdots & 0 & 0 & \cdots & 0 & & 0 & \cdots & 1 \\
\multicolumn{3}{c|}{-\alpha_{\ell 1}-} & \multicolumn{3}{c|}{-\alpha_{\ell 2}-} & & \multicolumn{3}{c}{-\alpha_{\ell\ell}-}
\end{array}\right]
\begin{array}{c}
\uparrow \\ n_1 \\ \\ \downarrow \\ \uparrow \\ n_2 \\ \\ \downarrow \\ \\ \uparrow \\ n_\ell \\ \\ \downarrow
\end{array}
\qquad
\hat{B} = \left[\begin{array}{c}
-b_1- \\ -b_2- \\ \vdots \\ \\ -b_n-
\end{array}\right]$$

$$\hat{H} = \left[\begin{array}{ccc|ccc|c|ccc}
1 & 0 & \cdots & \cdots & \cdots & \cdots & & \cdots & \cdots & 0 \\
0 & \cdots & 0 & 1 & 0 & \cdots & & \cdots & \cdots & 0 \\
\vdots & & & & & & & & & \vdots \\
0 & \cdots & \cdots & \cdots & \cdots & 0 & & 1 & 0\cdots & 0
\end{array}\right] \tag{18}$$

Because of Theorem 1 we can without loss of generality continue the derivation of Gopinath's input output equation assuming that $(\hat{A},\hat{B},\hat{H})$ satisfies (18). Simple manipulation of (1), (2) and the equivalence of (A,B,H) and $(\hat{A},\hat{B},\hat{H})$ yield

$$\bar{y}(k) = \left[\begin{array}{c} \hat{H} \\ \hat{H}\hat{A} \\ \vdots \\ \hat{H}\hat{A}^{n^*-1} \end{array}\right] \hat{x}(k) + R_1\, \bar{u}(k) \tag{19}$$

where

$$\begin{aligned} \bar{y}'(k) &= [y'(k)\ \ y'(k+1)\ \ \ldots\ \ y'(k+n^*-1)] \\ \bar{u}'(k) &= [u'(k)\ \ u'(k+1)\ \ \ldots\ \ u'(k+n^*-1)] \end{aligned} \tag{20}$$

and

$$R_1 = \begin{bmatrix} 0 \cdots & & & 0 \\ \hat{H}\hat{B} & 0 \cdots & & \vdots \\ \hat{H}\hat{A}\hat{B} & \hat{H}\hat{B} & 0 \cdots & \vdots \\ \vdots & & & \\ \hat{H}\hat{A}^{n^*-2} & \cdots\cdots\cdots & \hat{H}\hat{B} & 0 \end{bmatrix}. \tag{21}$$

The multiplication of (19) by S, using the equations (16) and (18), gives

$$S\ \bar{y}(k) = \hat{x}(k) + SR_1\ \bar{u}(k) \tag{22}$$

where

$$SR_1 = \left[\begin{array}{lllll} 0 \cdots & & & \cdots & 0 \\ b_1 & & & & \vdots \\ b_2 & b_1 & & & \\ \vdots & \vdots & & & \\ b_{n_1-1} & b_{n_1-2} & \cdots & b_1 \quad \cdots & 0 \\ \hline 0 \cdots & & & \cdots & 0 \\ b_{n_1+1} & & & & \vdots \\ \vdots & & & & \\ b_{n_1+n_2-1} & b_{n_1+n_2-2} \cdots & b_{n_1+1} & \cdots & 0 \\ \hline \hline 0 \cdots & & & & 0 \\ b_{n-n_\ell+1} & & & & \vdots \\ \vdots & & & & \\ b_{n-1} & \cdots\cdots & b_{n-n_\ell+1} & & 0 \end{array}\right] \tag{23}$$

Substituting $\hat{x}(k)$ and $\hat{x}(k+1)$ from (22) into

$$\hat{x}(k+1) = \hat{A}\ \hat{x}(k) + \hat{B}\ u(k)$$

yields Gopinath's direct input-output relation

$$S\ \bar{y}(k+1) = [\hat{A}R] \begin{bmatrix} S\ \bar{y}(k) \\ \bar{u}(k) \end{bmatrix} \tag{24}$$

where

$$R = -\hat{A}SR_1 + S \begin{bmatrix} \hat{H}\hat{B} & 0 \cdots & & \\ \hat{H}\hat{A}\hat{B} & \hat{H}\hat{B} & 0 \cdots & \\ \vdots & & & \\ \hat{H}\hat{A}^{n^*-1}\hat{B} & \cdots & & \hat{H}\hat{B} \end{bmatrix} \tag{25}$$

(24) is an external description in the sense that the variables u(k), y(k), can be measured externally. Equation (24) does not involve a system state. With relation (18), (23) and (25) it can be verified that

$$[\hat{AR}] = \left[\begin{array}{c|c|c|c|c|c}
\uparrow & 0\;1\quad 0 & 0\;\cdots\;0 & & 0\;\cdots\;0 & 0\;\cdots\qquad 0 \\
n_1 & \vdots\;\ddots & \vdots & \cdots & \vdots & \vdots \\
\downarrow & 0\qquad 1 & 0\;\cdots\;0 & & 0\;\cdots\;0 & 0\;\cdots\qquad 0 \\
 & -\alpha_{11}- & -\alpha_{12}- & & -\alpha_{1\ell}- & -\rho_1- \\
\hline
\uparrow & 0\;\cdots\;0 & 0\;1\quad 0 & & 0\;\cdots\;0 & 0\;\cdots\qquad 0 \\
n_2 & \vdots & \ddots & \cdots & \vdots & \vdots \\
\downarrow & 0\;\cdots\;0 & 0\qquad 1 & & 0\;\cdots\;0 & 0\;\cdots\qquad 0 \\
 & -\alpha_{21}- & -\alpha_{22}- & & -\alpha_{2\ell}- & -\rho_2- \\
\hline
 & \vdots & & & & \\
\hline
\uparrow & 0\;\cdots\;0 & 0\;\cdots\;0 & & 0\;1\quad 0 & 0\;\cdots\qquad 0 \\
n_\ell & \vdots & & & \vdots\;\ddots & \vdots \\
\downarrow & 0\;\cdots\;0 & 0\;\cdots\;0 & & 0\;\cdots\;1 & 0\;\cdots\qquad 0 \\
 & -\alpha_{\ell 1}- & -\alpha_{\ell 2}- & & -\alpha_{\ell\ell}- & -\rho_\ell-
\end{array}\right] \tag{26}$$

All significant parameters of $[\hat{AR}]$ appear in the rows n_1, n_2+n_1, ... n when a selector matrix of type 2 is used. Therefore the number of parameters that have to be estimated is reduced from $(n + mn^*)n$ to $(n + mn^*)\ell$.

The determination of $[\hat{AR}]$ is straightforward. Using (24) we have

$$S[\bar{y}(k+1)\;\ldots\;\bar{y}(k+n+mn^*)] = [\hat{A}\,|R]\left[\begin{array}{c} S\bar{y}(k)\;\ldots\;S\bar{y}(k+n+mn^*-1) \\ \hline \bar{u}(k)\;\ldots\;\bar{u}(k+n+mn^*-1)\end{array}\right] \tag{27}$$

which can be written compactly as

$$S\;Y_{n^*}(k+1) = [\hat{AR}]\;U_{n^*}(k) \tag{28}$$

Equation (26) allows us to introduce a reduced selector matrix

$$S_{red} = \left[\begin{array}{c} s_{n_1} \\ s_{n_1+n_2} \\ s_{n_1+n_2+\ldots n_\ell}\end{array}\right] \quad ; \quad S_{red} \text{ is } \ell \times n^*\ell \tag{29}$$

where s_i is the ith row of S.

Substituting S_{red} into (28) we get

$$S_{red}\ Y_{n*}(k+1) = \begin{bmatrix} \alpha_{11} & \alpha_{12} \cdots \alpha_{1\ell} & \rho_1 \\ \alpha_{21} \cdots & & \\ \vdots & & \\ \alpha_{\ell 1} \cdots & \alpha_{\ell\ell} & \rho_\ell \end{bmatrix} U_{n*}(k) \tag{30}$$

which has the form

$$S_{red}\ Y_{n*}(k+1) = [\hat{A}_{red}\ R_{red}]\ U_{n*}(k) \tag{31}$$

Note that (31) represents the input-output equations derived by Bonivento and Guirdozi in [3]. Theorem 2 states conditions under which a unique solution for $[\hat{A}R]$ and therefore $[\hat{A}_{red}\ R_{red}]$ can be found.

Theorem 2:

If (A,B,H) is completely observable and controllable and $S \in \mathcal{S}_{s2}$ and u(k) are random variables with joint nonlattice distribution, then $U_{n*}(k)$ is "almost surely" nonsingular, i.e. there exists a unique solution for $[\hat{A}R]$ with probability one.

The proof of this theorem is given in Appendix 1. Once $[\hat{A}R]$ is known, then $\hat{B}$ is found as described in [1].

$$\hat{B} = R_o + \hat{A}R_1 + \ldots\ \hat{A}^{n*-1}R_{n*-1} \tag{32}$$

where

$$R = [R_o R_1\ \ldots\ R_{n*-1}] \text{ with each } R_i \text{ an n x m matrix.} \tag{33}$$

Using equation (18), it can be verified that $\hat{H}$ is found in the following manner

$$h_{ij} = s_{ji} \qquad \text{for } i = 1, \ldots \ell \text{ and } j = 1, \ldots n \tag{34}$$

where $H = \{h_{ij}\}$ and $S = \{s_{ij}\}$

2.4. The Algorithm with Selector Matrix Type 3

As in the preceding section it can be shown that there always exists a selector matrix of type 3 such that equations (28) and (32) lead to

$$\hat{A} = \left[\begin{array}{c|c|c|c}
0\ \ 1 & & & \\
\vdots\ \ \ddots & 0 & \cdots & 0 \\
0\quad\ 1 & & & \\
-\alpha_{11}- & & & \\
\hline
0 \ldots 0 & 0\ \ 1\quad 0 & & \\
\vdots & \vdots\ \ \ddots & & 0 \\
0 \ldots 0 & 0 \ldots 1 & & \\
-\alpha_{21}- & -\alpha_{22}- & & \\
\hline
 & \vdots & & \\
\hline
0 \ldots 0 & 0 \ldots 0 & & 0\ \ 1\quad 0 \\
\vdots & & & \ddots \\
0 \ldots 0 & 0 \ldots 0 & & 0\quad\ 1 \\
-\alpha_{h1}- & -\alpha_{h2}- & & -\alpha_{hh}-
\end{array}\right]
\begin{array}{c}
\updownarrow n_1 \\ \\ \updownarrow n_2 \\ \\ \updownarrow n_h
\end{array}
\qquad
\hat{B} = \left[\begin{array}{c}
-\!-\!-\ b_1\ -\!-\!- \\
-\!-\!-\ b_2\ -\!-\!- \\
\vdots \\
\\
-\!-\!-\ b_n\ -\!-\!-
\end{array}\right] \tag{35}$$

Note that in this particular case equation (31) represents the input-output representation derived by Ackermann in [4]. The number of significant parameters in $\hat{A}$ is reduced because $\hat{A}$ is in lower triangular form. However, the output matrix $\hat{H}$ contains more significant parameters.

$$\hat{H} = \left[\begin{array}{c}
1\ 0 \ldots \qquad\qquad\qquad\qquad\qquad 0 \\
0 \ldots \qquad 1\ 0 \ldots \quad .. \qquad\qquad 0 \\
\\
\qquad\qquad\qquad\qquad\qquad 1\ 0 \ldots 0 \\
\hdashline
-\!-\!-\!-\!-\!-\ h_{h+1}\ -\!-\!-\!-\!-\!- \\
\vdots \\
-\!-\!-\!-\!-\!-\ h_{\ell}\ -\!-\!-\!-\!-\!-
\end{array}\right] \tag{36}$$

Equation (34) can still be used to find the first h rows of the matrix $\hat{H}$. The finding of the bottom rows requires the solution of the following system of linear equations

$$\begin{bmatrix}
y_{h+1}(k) \ldots & y_{h+1}(k+n-1) \\
y_{h+2}(k) \ldots & \\
\vdots & \\
y_{\ell}(k) \ldots & y_{\ell}(k+n-1)
\end{bmatrix}
=
\begin{bmatrix}
h_{h+1} \\ h_{h+2} \\ \vdots \\ h_{\ell}
\end{bmatrix}
*\left[\ \hat{x}(k)\ \ \hat{x}(k+1) \ldots \hat{x}(k+n-1)\right] \tag{37}$$

where $y_j(k)$ is the jth component of $y(k)$.

The model states $\hat{x}(k)$, $\hat{x}(k+1)$....can be found with equation (22) after $\hat{A}$ and $\hat{B}$ have been computed.

2.5. Summary

It has been shown that a selector matrix of type 2 in conjunction with equations (28), (32) and (34) leads directly to a canonical observable model $(\hat{A},\hat{B},\hat{H})$ as in (18). It has also been proved that such a selector matrix exists for any observable and controllable system with

$$\rho[H] = \ell$$

and that a unique solution of (28) is guaranteed for a random input sequence.

Similarly, a selector matrix of type 3 and the equations (28), (32), (34) and (37) yield a model as in (35) and (36). The proof that a selector matrix of type 3 always exists and that a unique solution of (28) and (37) is guaranteed for a random input sequence is analogous to the proof for selector matrix type 2.

III. Modified Versions of Gopinath's Method

3.1. Method A

A reduction of the matrix dimensions in equation (28) is achieved by decomposing the identification experiment into m intervals. A different input sequence is used during each interval and the data for one interval are used to identify $\hat{A}$ and one column of $\hat{B}$.

After interchanging columns in R and components in $\overline{u}(k)$ equation (24) can be split up in the following way

$$S\overline{y}(k+1) = \hat{A}S\overline{y}(k) + \overline{R}_1\overline{u}_1(k) + \ldots \overline{R}_m\overline{u}_m(k) \qquad (38)$$

where

$$\overline{R}_i = [r_i \; r_{i+m} \; \cdots \; r_{i+(n^*-1)m}]$$

r_i is the ith column of R (39)

and

$$\overline{u}_i'(k) = [u_i(k) \; u_i(k+1) \; \ldots \; u_i(k+n^*-1)]$$

$u_i(k)$ is the ith component of u(k)

Assuming that

$$\overline{u}_j(k) \equiv 0 \qquad \text{for } j \neq i \qquad (40)$$

we get the following partial input-output equation analogous to (24)

$$S\overline{y}(k+1) = [\hat{A}\overline{R}_i] \begin{bmatrix} S\overline{y}(k) \\ \overline{u}_i(k) \end{bmatrix} \qquad \text{for } i = 1, \ldots m \qquad (41)$$

The determination of $[\hat{A}\overline{R}_i]$ is straightforward. We have

$$S[\overline{y}(k+1)\ldots \overline{y}(k+n+n^*)] = [\hat{A}\overline{R}_i] \left[\begin{array}{c} S\overline{y}(k)\ldots S\overline{y}(k+n+n^*-1) \\ \hline \overline{u}_i(k)\ldots \overline{u}_i(k+n+n^*-1) \end{array}\right] \qquad (42)$$

$$\text{for } i = 1, \ldots m$$

which can be written compactly as

$$SY_{n^*}(k+1) = [\hat{A}\overline{R}_i]\, U_{in^*}(k) \qquad \text{for } i = 1, \ldots m \qquad (43)$$

A unique solution for $[\hat{A}\overline{R}_i]$ exists whenever $U_{in^*}(k)$ is nonsingular. We first assume that the one-input system (A, b_i, H) is controllable, where b_i

is the ith column of B. Then theorem 2 applies to (A,b_i,H), i.e. a unique solution for $[\hat{A}\bar{R}_i]$ exists with probability one whenever the input $u_i(j)$ is random. Most likely not all the m systems (A,b_i,H) will be controllable. If (A,b_i,H) is not controllable, general conditions that guarantee $U_{in^*}(k)$ to be nonsingular have not been found. However, whatever input sequence $u_i(j)$ is used, the starting condition

$$x(k) \neq 0, \text{ and therefore } \hat{x}(k) \neq 0 \qquad (44)$$

is necessary for $U_{in^*}(k)$ to be nonsingular. The proof follows directly from an equation analogous to (4) in Appendix 1.

Once $[\hat{A}\bar{R}_i]$ is determined the column b_i of $\hat{B}$ is found directly from

$$b_i = \bar{r}_o + \hat{A}\bar{r}_1 + \dots \hat{A}^{n^*-1}\bar{r}_{n^*-1} \qquad (45)$$

where

$$\bar{R}_i = [\bar{r}_o \ \bar{r}_1 \dots \bar{r}_{n^*-1}] \text{ with each } \bar{r}_i \text{ an n x 1 vector.}$$

Relation (45) follows from (32) and (39).

3.2. Method B

Particular input sequences allow a modification of (A,B,H) into an augmented system $(\underline{A},\underline{H})$ eliminating the input matrix. The identification of the augmented system with Gopinath's method will give $\hat{A}$ directly and is simplified because the term R in the input-output equation (24) disappears. $\hat{B}$ is then identified columnwise with a different algorithm which does not require any knowledge about $\hat{A}$.

3.2.1. $\hat{A}$-Algorithm

For any input sequence of the type

$$u(k+1) = C\,u(k) \quad \text{for } k = 1, \dots \qquad (46)$$

where C is m x m, the dynamic behaviour of system (1), (2) can be described by the following augmented state equations

$$\underline{x}(k+1) = \underline{A}\ \underline{x}(k) \qquad (47)$$

$$\underline{y}(k) = \underline{H}\ \underline{x}(k)$$

where

$$\underline{x}'(k) = [u'(k) \quad x'(k)]$$

$$\underline{y}'(k) = [u'(k) \quad \underline{y}'(k)]$$

and

$$\underline{A} = \left[\begin{array}{c|c} C & 0 \\ \hline B & A \end{array}\right] , \qquad \underline{H} = \left[\begin{array}{c|c} Im & 0 \\ \hline 0 & H \end{array}\right] . \qquad (48)$$

It is obvious that

$$\rho[\underline{H}] = m + \rho[H] \qquad (49)$$

We define the following augmented (m+n) x (m+ℓ)n* selector matrix

$$\underline{S} = \left[\begin{array}{c|c|c|c|c|c} Im & & & 0 & & \\ \hline 0 & S_1 & 0 & S_2 \ldots & 0 & S_{n^*} \\ \underleftrightarrow{m} & & \underleftrightarrow{m} & & \underleftrightarrow{m} & \end{array}\right] , \qquad (50)$$

where $S = [S_1 \; S_2 \; \ldots \; S_{n^*}]$ with S_i an n x ℓ matrix, is the selector matrix for the original system (A,B,H) as used in (15) and (24).

It is now easy to show that selector matrix $\underline{S}$ can be used to identify an augmented model $(\hat{\underline{A}},\hat{\underline{H}})$ with Gopinath's method. Multiplying $\underline{S}$ with the observability matrix of $(\underline{A},\underline{H})$ and using (15), (48) and (50) gives

$$\underline{S}\begin{bmatrix} \underline{H} \\ \underline{HA} \\ \vdots \\ \underline{HA}^{n^*-1} \end{bmatrix} = \underline{S}\left[\begin{array}{c|c} I_m & 0 \\ \hline 0 & H \\ \hline C & 0 \\ \hline X & HA \\ \hline C^2 & 0 \\ \vdots & \\ \hline C^{n^*-1} & 0 \\ \hline X & HA^{n^*-1} \end{array}\right] = \left[\begin{array}{c|c} I_m & 0 \\ \hline X & S\begin{bmatrix} H \\ HA \\ \vdots \\ HA^{n^*-1} \end{bmatrix} \end{array}\right] = \left[\begin{array}{c|c} I_m & 0 \\ \hline X & T \end{array}\right] = \underline{T} \qquad (51)$$

Because T is nonsingular $\underline{T}$ must be nonsingular too, i.e. the augmented system $(\underline{A},\underline{H})$ as defined in (48) is observable. Furthermore $\underline{S}$ can be used to identify model $(\hat{\underline{A}},\hat{\underline{H}})$. Analogous to (17) we obtain

$$\hat{\underline{A}} = \underline{T}\,\underline{A}\,\underline{T}^{-1} \tag{52}$$

which leads with (48) and (51) to

$$\hat{\underline{A}} = \left[\begin{array}{c|c} C & 0 \\ \hline X & T\,A\,T^{-1} \end{array}\right] = \left[\begin{array}{c|c} C & 0 \\ \hline X & \hat{A} \end{array}\right] \tag{53}$$

where $\hat{A}$ is identical to $\hat{A}$ in (18) and (24).

Because $(\underline{A},\underline{H})$ has no input matrix, equation (24) is reduced to

$$\underline{S}\,\bar{\underline{y}}(k+1) = \hat{\underline{A}}\,\underline{S}\,\bar{\underline{y}}(k) \tag{54}$$

and equation (24) becomes

$$\underline{S}\,\underline{Y}_{n*}(k+1) = \hat{\underline{A}}\,\underline{S}\,\underline{Y}_{n*}(k) \tag{55}$$

with

$$\underline{Y}_{n*}(k) = [\bar{\underline{y}}(k)\ \ \bar{\underline{y}}(k+1)\ \ \dots\ \ \bar{\underline{y}}(k+n+m-1)]$$

where $\bar{\underline{y}}(k)$ is defined corresponding to (20) and (48). A unique solution for $\hat{\underline{A}}$ is found whenever

$$\underline{S}\,\underline{Y}_{n*}(k) = \left[\begin{array}{c} u(k)\ \ Cu(k)\ \ \dots\ \ C^{n+m-1}u(k) \\ \hdashline S\bar{y}(k)\ \ S\bar{y}(k+1)\dots S\bar{y}(k+n+m-1) \end{array}\right] \tag{56}$$

is nonsingular.

Sufficient and practical relevant conditions for C, u(k) and initial system state x(k) such that (56) is satisfied have not been found. However, it is necessary that matrix C be cyclic (definition in Appendix 1). This is no serious restriction. "The condition of being non-cyclic is caused by having two identical subsystems embedded in one system and yet completely decoupled from each other"[15]. In spite of an arbitrary choice of C and u(k) in all numerical examples a unique $\hat{\underline{A}}$ was always found, i.e. the algorithm is practical even without sufficient conditions for C, u(k) and x(k).

3.2.2. $\hat{B}$-Algorithm

Simple manipulation of (1) and (2) gives

$$\bar{y}(k+1) = \begin{bmatrix} \hat{H}\hat{A} \\ \hat{H}A^{2} \\ \vdots \\ \hat{H}\hat{A}^{n*} \end{bmatrix} \hat{x}(k) + \begin{bmatrix} \hat{H}\hat{B} & & \\ \hat{H}\hat{A}\hat{B} & \hat{H}\hat{B} & \\ \vdots & & \\ \hat{H}\hat{A}^{n*-1}\hat{B} & \dots & \hat{H}\hat{B} \end{bmatrix} \bar{u}(k) \qquad (57)$$

where $\bar{y}(k)$ and $\bar{u}(k)$ are defined in (20).

It is assumed now that $(\hat{A},\hat{B},\hat{H})$ is initially in the steady state, in the sense that

$$\begin{aligned} \hat{x}_s &= \hat{A}\hat{x}_s + \hat{B}u_s \\ y_s &= \hat{H}\hat{x}_s \end{aligned} \qquad (58)$$

Let

$$\begin{aligned} y_o(k) &\triangleq y(k) - y_s \\ \hat{x}_o(k) &\triangleq \hat{x}(k) - \hat{x}_s \\ u_o(k) &\triangleq u(k) - u_s \end{aligned} \qquad (59)$$

Substituting $y, \hat{x}$ and u in (57) and using (58) gives

$$\begin{bmatrix} y_o(1) \\ \vdots \\ y_o(n*) \end{bmatrix} = \begin{bmatrix} \hat{H}\hat{B} & & \\ \hat{H}\hat{A}\hat{B} & \hat{H}\hat{B} & \\ \vdots & & \\ \hat{H}\hat{A}^{n*-1}\hat{B} & \dots & \hat{H}\hat{B} \end{bmatrix} \begin{bmatrix} u_o(0) \\ \vdots \\ u_o(n*-1) \end{bmatrix} \qquad (60)$$

To simplify the notation $y(k)$, $u(k)$ shall denote $y_o(k)$ and $u_o(k)$ in the rest of this section.

In the case of m different input sequences of length n* equation (60) can be written in the form

$$\begin{bmatrix} Y_1 \\ \vdots \\ Y_{n*} \end{bmatrix} = \begin{bmatrix} \hat{H}\hat{B} & & \\ \hat{H}\hat{A}\hat{B} & \hat{H}\hat{B} & \\ \vdots & & \\ \hat{H}\hat{A}^{n*-1}\hat{B} & \dots & \hat{H}\hat{B} \end{bmatrix} \begin{bmatrix} U_o \\ \vdots \\ U_{n*-1} \end{bmatrix} \qquad (61)$$

where

$$\begin{aligned} Y_k &= [y_1(k)\ y_2(k)\ \dots\ y_m(k)] \quad \text{is an } \ell \times m \text{ matrix} \\ U_k &= [u_1(k)\ u_2(k)\ \dots\ u_m(k)] \quad \text{is an } m \times m \text{ matrix} \end{aligned} \tag{62}$$

and

$$\begin{aligned} &y_i(k),\ k = 1, 2, \dots n^* \quad \text{represents the ith output sequence} \\ &u_i(k),\ k = 0, 1, 2, \dots n^*-1 \quad \text{represents the ith input sequence} \end{aligned} \tag{63}$$

The starting values $u_k(0)$ of the input sequences are chosen such that they span the whole input space, i.e.

$$U_o \text{ is non-singular} \tag{64}$$

With (61) we compute sequentially the following terms

$$\begin{aligned} \hat{H}\hat{B} &= Y_1\, U_o^{-1} \\ \hat{H}\hat{A}\hat{B} &= (Y_2 - \hat{H}\hat{B}U_1)\, U_o^{-1} \\ \hat{H}\hat{A}^{n^*-1}\hat{B} &= (Y_{n^*} - \hat{H}\hat{A}^{n^*-2}\hat{B}U_1 - \dots - \hat{H}\hat{A}\hat{B}\, U_{n^*-2} - \hat{H}\hat{B}U_{n^*-1})U_o^{-1} \end{aligned} \tag{65}$$

Recalling (16) one gets $\hat{B}$ directly by use of (65)

$$S \begin{bmatrix} \hat{H}\hat{B} \\ \hat{H}\hat{A}\hat{B} \\ \vdots \\ \hat{H}\hat{A}^{n^*-1}\hat{B} \end{bmatrix} = S \begin{bmatrix} \hat{H} \\ \hat{H}\hat{A} \\ \vdots \\ \hat{H}\hat{A}^{n^*-1} \end{bmatrix} \hat{B} = \hat{B} \tag{66}$$

Equation (65) becomes particularly simple when m stepfunctions are used as input sequences such that

$$U_o = U_1 = \dots = U_{n^*-1} = U \tag{67}$$

and

$$U = \begin{bmatrix} U_{s1} & & & 0 \\ & U_{s2} & & \\ & & \ddots & \\ 0 & & & U_{sm} \end{bmatrix} \tag{68}$$

Substituting (67) and (68) into (65) gives

$$\begin{aligned} \hat{H}\hat{B} &= Y_1\, U^{-1} \\ \hat{H}\hat{A}\hat{B} &= (Y_2 - Y_1)\, U^{-1} \\ \hat{H}\hat{A}^{n^*-1}\hat{B} &= (Y_{n^*} - Y_{n^*-1})\, U^{-1} \end{aligned} \tag{69}$$

and for (66)

$$S \begin{bmatrix} Y_1 \\ Y_2 - Y_1 \\ \vdots \\ Y_{n^*} - Y_{n^*-1} \end{bmatrix} U^{-1} = \hat{B} \tag{70}$$

Partitioning (70) into m columns gives

$$S \begin{bmatrix} y_i(1) \\ y_i(2) - y_i(1) \\ \vdots \\ y_i(n^*) - y_i(n^*-1) \end{bmatrix} \frac{1}{u_{si}} = b_i \quad \text{for } i = 1,\ldots m \tag{71}$$

where b_i is the ith column of $\hat{B}$.

Note that the matrix $\hat{A}$ need not be known in order to solve (65) and (66) or (70) and (71) for $\hat{B}$.

3.3 Comparison with Gopinath's Method

Four aspects of identification that are important in practical applications are discussed for Gopinath's method and method A and B.

1) order of input-output equations

2) input restrictions

3) number of experiments

4) initial conditions

1) The order of the systems of linear input-output equations (28), (43) and (55) varies from method to method. Low order equations require less computer storage and time and give more accurate results (less numerical errors). Method A and B are always better than Gopinath's method in this respect. Method B is the best one in case of few output components. The following maximum matrix dimensions are obtained in equation (28), (43)

and (55) respectively for a fourth order system with a maximum of 4 inputs and 4 outputs.

$$\max\ (n + mn^*) = 20 \quad \text{for Gopinath's method}$$

$$\max\ (n + n^*) = 8 \quad \text{for Method A}$$

$$\max\ (n + m) = 8 \quad \text{for method B.}$$

2) Only Gopinath's method facilitates theoretically the identification of a model $(\hat{A},\hat{B},\hat{H})$ from normal operating records but better results will certainly be achieved with the application of chosen input signals. All three methods allow the use of simple input sequences like the square wave illustrated in Figure 3.1.

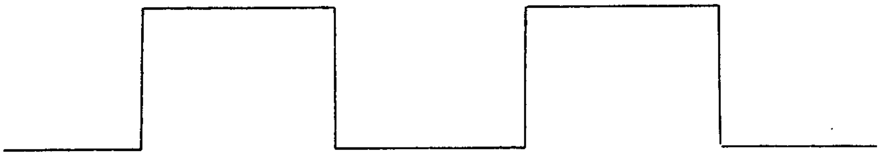

Figure 3.1.

The following restrictions for the input sequences have to be considered though:

The lower mn* rows in $U_{n^*}(k)$ in equation (28) have to be linearly independent if we use Gopinath's method. The same is necessary for the lower n* rows in $U_{in^*}(k)$ in equation (43) for method A whereas for method B equation (46) has to be satisfied. Tests showed that the occurrence of a singular data matrix from such restricted input sequences is unlikely (Experiment 1 and 2).

3) The number of necessary experiments is:

1 for Gopinath's method

1 with m intervals for method A

m + 1 for method B.

4) No initial conditions for the system are required by Gopinath's method whereas in case A equation (44) has to be satisfied. With method B the system has to be in the steady-state when the identification of $\hat{B}$ is started. This restriction is balanced off by the advantage that the identification of $\hat{A}$ and $\hat{B}$ is decoupled in case B.

In conclusion, it may be said that there is a trade off between experimental and computational requirements. We obtain small matrices and numerical decoupling but require several experiments for method B on the one hand and obtain large matrices with only one experiment for Gopinath's method on the other hand.

IV. IDENTIFICATION OF REAL-TIME SYSTEMS WITH A MINICOMPUTER

4.1 Experiment

Identification method B has been implemented on a NOVA minicomputer in order to identify models of linear systems realized on the DONNER 3400 analog computer. The experimental setup is shown in Figure 4.1. The NOVA as a central unit supervises the identification experiment, in particular it controls the A/D conversion and initiates and resets the input sequences. The computer accepts known system parameters (dimension, etc.) from the teletype and computes the model parameters from the measured data after all the samples have been taken. The control logic generates the desired input sequence at a rate given by the external signal generator. The state of the sampling logic tells the waiting computer when a sample can be taken and when the experiment can be started.

4.2. Software for Minicomputer

Because of the 4K memory limitation of the standard NOVA emphasis was put on efficient storage use rather than fast data processing whenever these two goals were incompatible. On-line computation of $\hat{A}$ and $\hat{B}$ would have been possible but would not have resulted in storage savings as all input-output data had to be stored anyway for teletype output and performance evaluation. Lt proved to be simpler to acquire all experimental data first and then compute $\hat{A}$ and $\hat{B}$ cff line. The program flow chart is shown in Figure 4.2. The experiment supervision and data acquisition is performed by the same subroutine in both case $\hat{A}$ and case $\hat{B}$ even though it is represented by two different boxes in the flow chart.

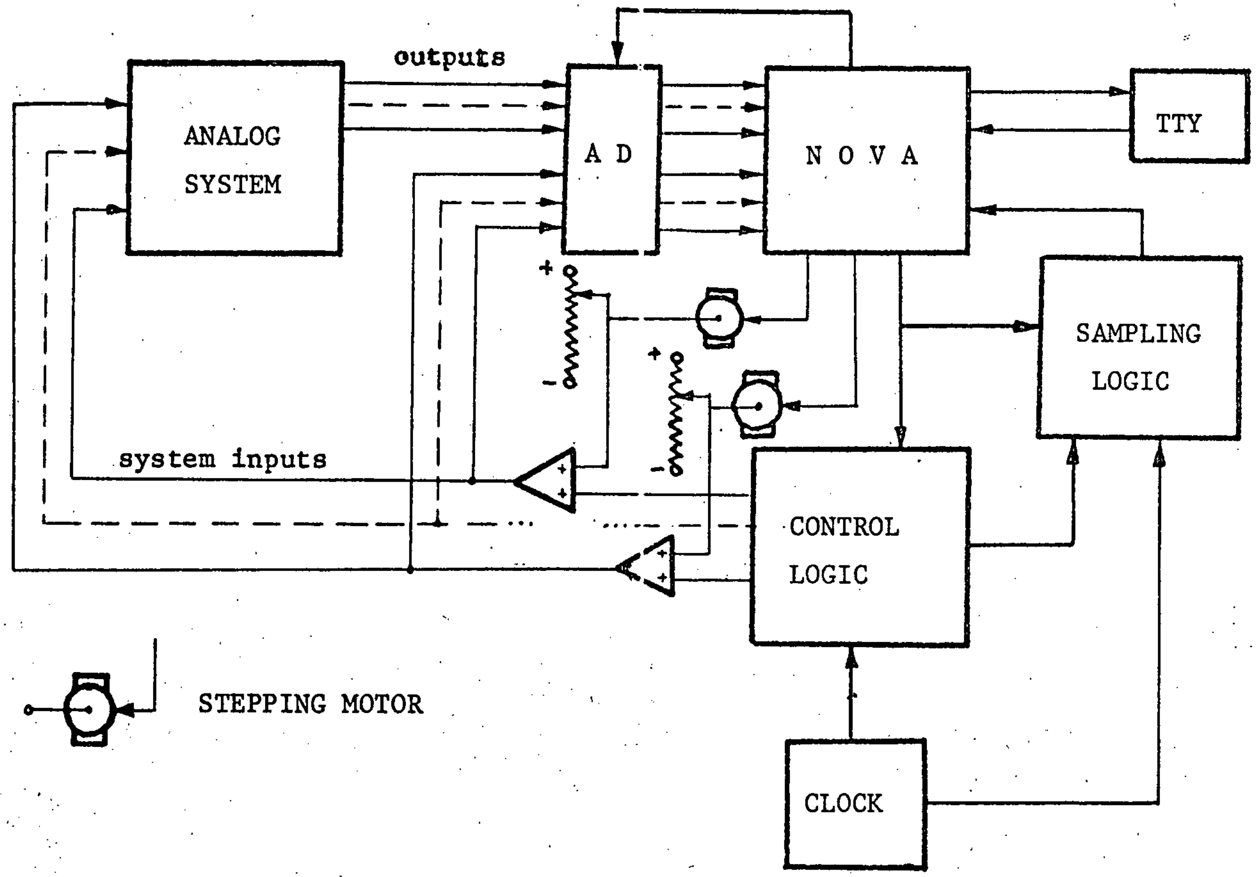
outputs
ANALOG SYSTEM
A D
N O V A
TTY
SAMPLING LOGIC
CONTROL LOGIC
CLOCK
system inputs
STEPPING MOTOR

Figure 4.1.

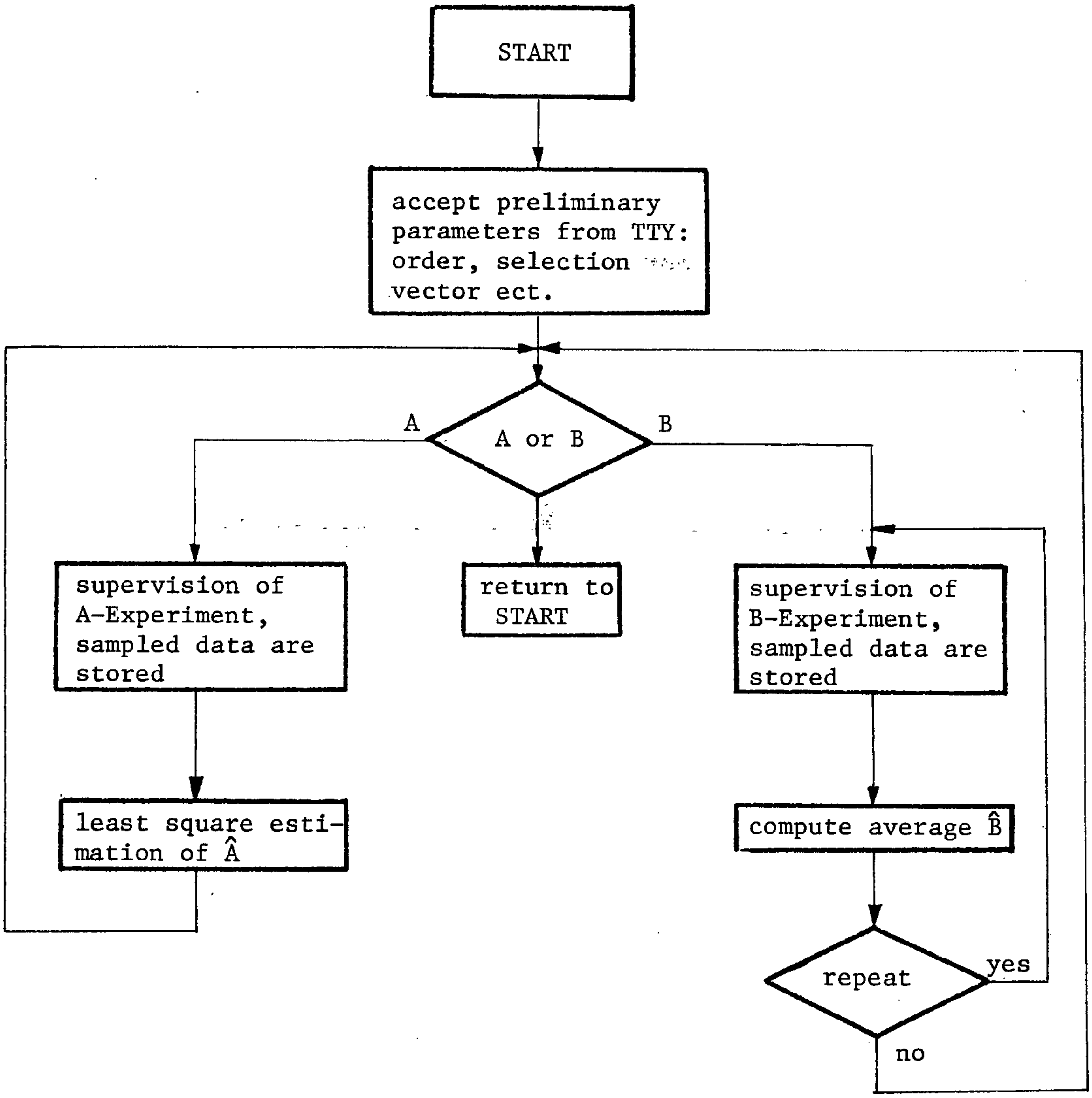
START
accept preliminary parameters from TTY: order, selection vector ect.
A or B
A
B
supervision of A-Experiment, sampled data are stored
return to START
supervision of B-Experiment, sampled data are stored
least square estimation of $\hat{A}$
compute average $\hat{B}$
repeat
yes
no

Figure 4.2.

The computation of $\hat{\underline{A}}$ from equation (55) proved to be insufficient even for the small disturbances described in Section 4.3. In order to reduce the influence of measurement noise more data has to be included into the parameter identification. Equation (55) can be generalized to

$$Z = \hat{\underline{A}}D + V \qquad (72)$$

where Z and D are (n+m) x d with d = 1,2,... and V is an array of noise terms. The least squares estimate of $\hat{\underline{A}}$ is then

$$\hat{\underline{A}}_e = ZD' \, [DD']^{-1} \quad \text{for} \quad d \geq (m+n) \qquad (73)$$

The least squares program described in [7] was used to solve equation (73) recursively. The algorithm starts out with d = 1 and then updates $\hat{\underline{A}}_e$ with each new data vector. New data vectors are checked for linear independence until the first unique solution (d = m+n) is reached. This check enables us to detect cases of singular or nearly singular data matrices. (The recursive least squares algorithm and the linear independence check are presented in Appendix 2.)

Matrix $\hat{B}$ is computed columnwise with formula (71), therefore only step inputs as given by (67) and (68) can be used. In the presence of measurement noise we have

$$S \begin{bmatrix} y_i(1) \\ y_i(2) - y_i(1) \\ \vdots \\ y_i(n^*) - y_i(n^*-1) \end{bmatrix} \frac{1}{u_{si}} = \hat{b}_i = b_i + v \quad \text{for } i = 1,\ldots m \qquad (74)$$

where $\hat{b}_i$ is the estimate of column b_i and v is a noise vector.

If we assume that

$$E[\hat{b}_i - b_i] = E[v] = 0 \tag{75}$$

and $$E[vv'] = R$$

then it is possible to reduce the covariance matrix R by averaging N independent estimates. We compute

$$\hat{b}_i = \frac{1}{N} \sum_{j=1}^{N} \hat{b}_{ij} \tag{76}$$

where $\hat{b}_{ij}$ = jth estimate of b_i

$$v_j = \hat{b}_{ij} - b_i$$

$$E[v_j v_k'] = \begin{cases} R & \text{for } j = k \\ 0 & \text{for } j \neq k \end{cases} \tag{77}$$

and get

$$E[\hat{b}_i - b_i] = E[v] = 0$$

$$E[vv'] = \frac{R}{N} = R_s \tag{78}$$

The program enables the user to repeat the identification of $\hat{B}$ several times and it computes the average of all parameter estimates as a final result.

The program diverges in one point from the described theory. The selector matrix is replaced by the less general selection vector because it saves storage and computing time and is easier to initialize. Whereas selector matrices can select rows out of a data matrix ordered in any possible way the selection vectors select the rows as they are ordered in the data matrix, e.g. s = [1011] indicates that out of 4 rows number 1,3 and 4 are selected in this sequence. The consequence of this is that model $(\hat{A},\hat{B},\hat{H})$ will not be in the canonical form described in (18). Example

1 and 2 in Experiment 1 show, however, that the canonical form is easily obtained by interchanging of rows and columns in $\hat{A}$, rows in $\hat{B}$ and columns in $\hat{H}$.

4.3 Sources of Error

The experimentally found model is inaccurate even for an exactly linear system because of noise contaminated system outputs, quantization errors in the A/D conversion from continuous signals to 12-bit binary numbers and computational errors (truncation, round off). The relative parameter error caused by all the above effects was less than one percent in all experiments of Chapter 5. A further error source are nonideal input signals as indicated by the dashed lines in Figure 4.3.

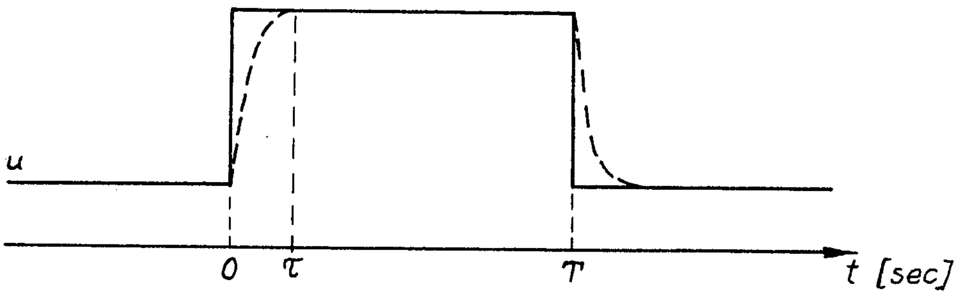

Figure 4.3.

The computation of model parameters is based on the assumption of ideal input step functions and sampling of input and output immediately after the steps. The measuring of an incorrect input amplitude can be prevented by delaying the sampling of the input by τ and the influence of the nonideal shape on the output can be neglected as long as the condition

$$\tau << T$$

is satisfied. A delay of the input sample was built into the program and set at $\tau = 0.0032$ seconds for the experiments in chapter V.

4.4. Storage Allocation and Time Requirements

The storage allocation in the program is such that at the most fourth order systems with up to four inputs and four outputs can be identified. The 4000 available storage locations are apportioned as follows

1215	for floating point arithmetic and operating software (binary loader etc.)
1255	for complete identification package (method B)
1125	for data storage
405	empty
4000	total

It was calculated that about 4300 storage locations would be required for the identification of fifth order systems. An estimation of the storage needed by Gopinath's procedure for fourth order systems was carried out under the assumption that only the least squares algorithm of the complete program would be used. It was found that about 7500 storage locations would be required, i.e. twice as much as with method B or A.

The used recursive least squares algorithm completes the cycle for one data vector for a fourth order system in about three seconds which indicates the length of the shortest possible sampling interval for real-time identification of $\hat{A}$.

V. EXPERIMENTS AND RESULTS

The purpose of the following experiments was to test the programs written for the proposed identification methods A and B and to study various properties of the algorithms. Linear systems were either realized on the analog computer DONNER 3400 or simulated on the IBM 370.

Models identified on the large computer were tested by comparing their step response with the one of the original system over the first 15 sampling intervals whereas models computed on the NOVA with real system data were compared with models found from a disturbance free system simulation on the IBM 370.

Experiment 1

The following system was simulated and identified on the IBM 370.

$$A = \begin{bmatrix} 0 & 1 & -1 & 0 & -1 \\ -1 & 1 & 1 & 0 & 2 \\ 1 & 0 & 1 & -2 & 0 \\ 1 & 0 & 1 & -2 & 1 \\ 1 & 0 & -1 & 0 & -3 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \qquad H = \begin{bmatrix} 1 & 0 & 1 & -1 & -1 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix} \quad (0)$$

The results achieved with four different selector matrices are shown in Table 5.1. Both identification methods were applied in all four cases, method A with an input sequence α, α, 0, 0, α, α, 0, 0,...and Method B with a step input. The equivalence of the models (1), (2), (3), (4) and system (0) was then verified by matching their step responses. The errors caused by round-off were less than 10^{-6} for model parameters and step response, i.e. both method A and B gave identical and correct models.

Selector Matrices	Selection Vector	$\hat{A}$	$\hat{H}$	
$S_1 = \left[\begin{array}{cccccc} 1&0&0&0&0&0 \\ 0&0&1&0&0&0 \\ 0&0&0&0&1&0 \\ \hline 0&1&0&0&0&0 \\ 0&0&0&1&0&0 \end{array}\right]$ $n^* = 3$	—	$\hat{A}_1 = \left[\begin{array}{ccc\|cc} 0&1&0&0&0 \\ 0&0&1&0&0 \\ 1&0&-1&-1&1 \\ \hline 0&0&0&0&1 \\ 1&0&-1&1&-2 \end{array}\right]$	$\hat{H}_1 = \begin{bmatrix} 1&0&0&0&0 \\ 0&0&0&1&0 \end{bmatrix}$	(1)
$S_2 = \begin{bmatrix} 1&0&0&0&0&0 \\ 0&1&0&0&0&0 \\ 0&0&1&0&0&0 \\ 0&0&0&1&0&0 \\ 0&0&0&0&1&0 \end{bmatrix}$ $n^* = 3$	$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$	$\hat{A}_2 = \begin{bmatrix} 0&0&1&0&0 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \\ 1&1&0&-2&-1 \\ 1&-1&0&1&-1 \end{bmatrix}$	$\hat{H}_2 = \begin{bmatrix} 1&0&0&0&0 \\ 0&1&0&0&0 \end{bmatrix}$	(2)
$S_3 = \left[\begin{array}{cccccccc} 1&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&1&0 \\ \hline 0&1&0&0&0&0&0&0 \end{array}\right]$ $n^* = 4$	—	$\hat{A}_3 = \left[\begin{array}{cccc\|c} 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \\ 4&1&-4&-4&-2 \\ \hline -1&0&1&1&1 \end{array}\right]$	$\hat{H}_3 = \begin{bmatrix} 1&0&0&0&0 \\ 0&0&0&0&1 \end{bmatrix}$	(3)
$S_4 = \begin{bmatrix} 1&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0 \end{bmatrix}$ $n^* = 5$	$\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$	$\hat{A}_4 = \begin{bmatrix} 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \\ -2&3&3&-2&-3 \end{bmatrix}$	$\hat{H}_4 = \begin{bmatrix} 1&0&0&0&0 \\ 2&0.5&-2&-2&-0.5 \end{bmatrix}$	(4)

Table 5.1.

The four models (1), (2), (3) and (4) demonstrate the relation between selector matrix and model structure. Selector matrix S_1 and S_3 are of type 2 and the structures of model (1) and (3) verify equation (18) and (34). Selector matrix S_1 leads to two coupled subsystems of second and third order in $\hat{A}_1$ but with S_3 the subsystems in $\hat{A}_3$ are of fourth and first order. Matrix $\hat{A}_4$ is in lower triangular form because S_4 is of type 3. Model (4) demonstrates that the complete system (0) is observable through the first output component, consequently $\hat{A}_4$ contains only one subsystem and the second row of $\hat{H}_4$ becomes "general" according to (36). Model (1), (3) and (4) verify that the selector matrix principle allows one to choose the order and number of subsystems within certain limits.

The Selector matrices S_1 and S_2 select the same rows in a different order where S_1 is of type 2 and S_2 has a corresponding selection vector. The models (1) and (2) are hence very closely related. An exchange of rows and columns in $\hat{A}_1$ after the following scheme

$$2 \rightarrow 3, \quad 3 \rightarrow 5, \quad 4 \rightarrow 2, \quad 5 \rightarrow 4$$

gives $\hat{A}_2$. In $\hat{B}_1$ only the rows and in $\hat{H}_1$ only the columns have to be exchanged in order to get $\hat{B}_2$ and $\hat{H}_2$.

Experiment 2

The following continuous third order system with two inputs and two outputs was realized on the analog computer.

$$\dot{x} = \begin{bmatrix} 0.0 & 1.0 & 0.0 \\ -0.04 & -0.1 & 0.0 \\ 0.0 & 0.0 & -0.08 \end{bmatrix} x + \begin{bmatrix} 2.0 & 2.0 \\ 0.1 & 0.0 \\ 1.0 & 0.0 \end{bmatrix} u$$

$$y = \begin{bmatrix} -1.0 & 1.0 & 1.0 \\ 1.0 & 0.0 & 0.0 \end{bmatrix} x \tag{5}$$

A sampling period period T = 3 seconds, n* = 2 and the selection vector [1 1 1 0] were chosen for the identification of a discrete model. The input sequence for the $\hat{A}$-identification is shown below

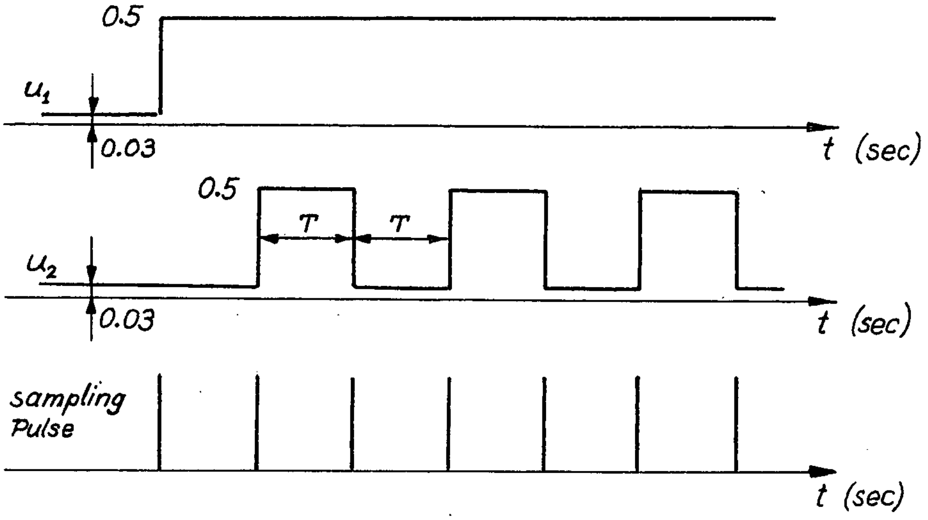

Figure 5.1.

Steps on input 1 respectively 2 were used for the identification of column 1 respectively 2 of $\hat{B}$.

The model computed on the NOVA with real data as described in Chapter 3 was

$$\hat{A} = \begin{bmatrix} 0.00000 & -0.00000 & 1.00000 \\ 0.72637 & 0.69602 & -0.92843 \\ -0.58098 & 0.24383 & 1.53274 \end{bmatrix} \quad \hat{B} = \begin{bmatrix} -3.44633 & -6.03377 \\ 6.03390 & 5.68480 \\ -3.48022 & -4.76172 \end{bmatrix} \tag{6}$$

$$\hat{H} = \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \end{bmatrix}$$

System (5) was also simulated and identified on the IBM 370. The resulting model was

$$\hat{A} = \begin{bmatrix} 0.00000 & 0.00000 & 1.00000 \\ 0.73002 & 0.70000 & -0.92806 \\ -0.58131 & 0.24084 & 1.52562 \end{bmatrix} \quad \hat{B} = \begin{bmatrix} -3.47356 & -5.98833 \\ 6.06758 & 5.67127 \\ -3.46499 & -4.74163 \end{bmatrix}$$

$$\hat{H} = \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \end{bmatrix} \tag{7}$$

Figure 5.2. shows that the free motion ($u = 0$) of the analog system and the simulated system are nearly identical. The relative difference between (6) and (7) exceeds 1% for only one parameter in $\hat{A}$. This discrepancy is caused by the difference between the analog system and its simulation as well as the error sources mentioned in Section 4.3.

Repetitions of the experiment with changed input amplitudes $u_1 = u_2 = 0.35$ or with exchanged u_1 and u_2 gave results of the same accuracy. Experiments were also carried through under the assumption of incorrect system orders 4 and 2. The case $n = 4$ with real system and simulated data resulted in linear dependent data vectors. Consequently, no model parameters could be computed. Model parameters were always found in the case $n = 2$. However, the parameters depended on the input amplitudes and the initial state of the system and were therefore meaningless.

<u>Experiment 3</u>

Model (6) found in experiment 2 can be used for the design of a digital controller for the analog system (5). A suboptimal control

$$u(k) = -\hat{F}\,\hat{x}(k) \tag{8}$$

was chosen to approximately minimize the performance function

$$J = \hat{x}'(N)Q_o\hat{x}(N) + \sum_{k=0}^{N-1} \hat{x}'(k)Q_1\hat{x}(k) + u'(k)Q_2u(k)$$

with

$$N = 15$$
$$Q_o = Q_1 = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & 1 \end{bmatrix} \quad \text{and} \quad Q_2 = \begin{bmatrix} 20 & 0 \\ 0 & 20 \end{bmatrix} \tag{9}$$

where $\hat{x}(k)$ is the observed state of model $(\hat{A},\hat{B},\hat{H})$.

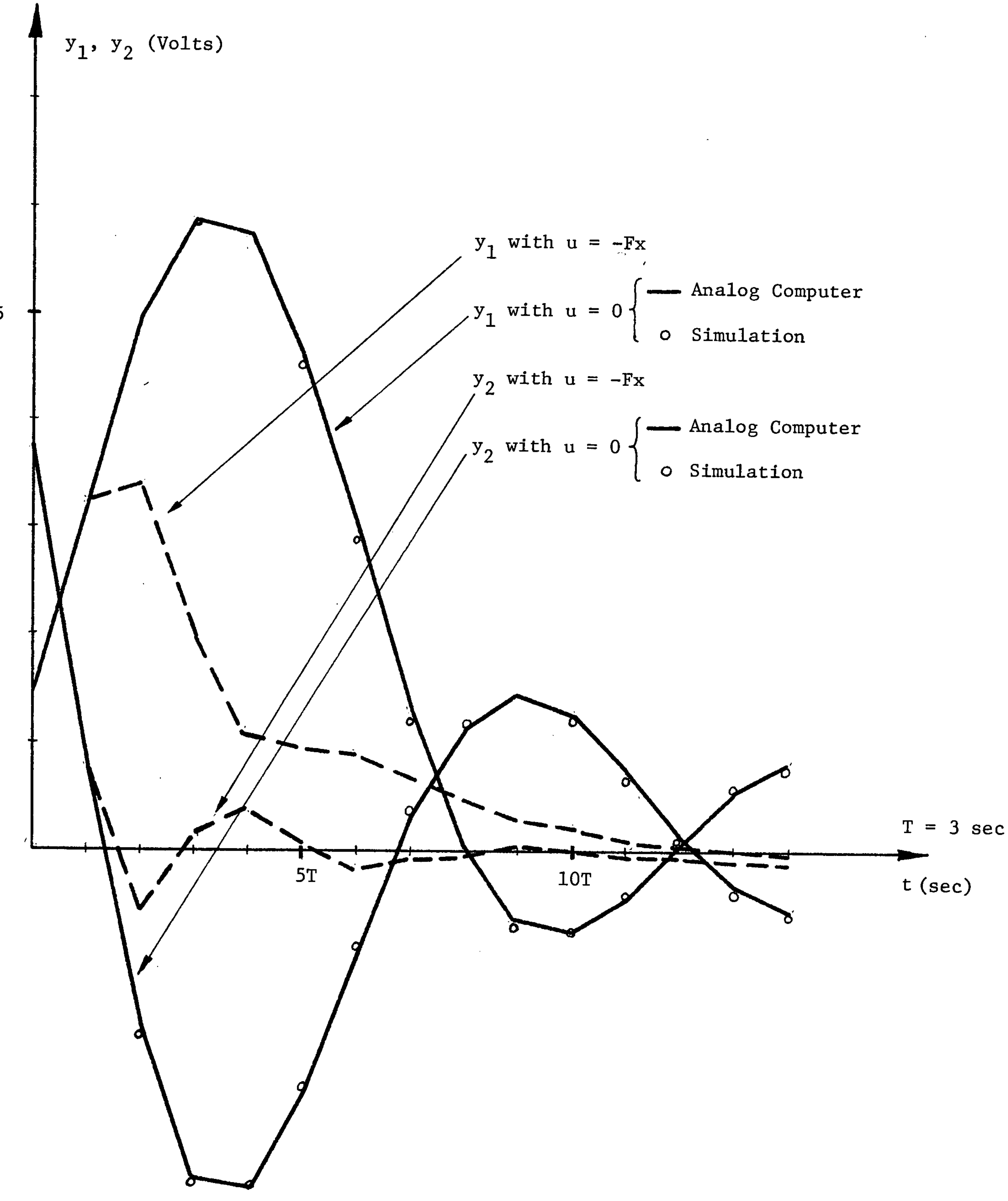
y_1, y_2 (Volts)
5
y_1 with u = -Fx
y_1 with u = 0
— Analog Computer
o Simulation
y_2 with u = -Fx
y_2 with u = 0
— Analog Computer
o Simulation
T = 3 sec
5T
10T
t (sec)

Figure 5.2.

The formulas for the computation of $\hat{F}$ and the state observation are given in Appendix 3. The experimental scheme is shown in Figure 4.1., where the computer controlled stepping motors coupled with precision potentionmeters set the inputs according to (8) after each sampling pulse. $\hat{F}$ was computed on the NOVA in 30 seconds before the start of the control sequence.

Figure 5.2. compares for an initial state

$$x'(0) = [3.75\ \ -1.0\ \ \ 6.25] \qquad y'(0) = [1.5\ \ \ 3.75]$$

the output obtained with control (8) with the open-loop case with control u(k) = 0 for all stages. The potentiometer control voltages are shown in Figure 5.4. Note the zero-input at stage 0 resulting from the fact that the first state observation was only possible at stage 1.

Control (8) was also simulated on the IBM 370 and the obtained output data are compared with the real output in Figure 5.3. The oscillation of the real response around the simulated undisturbed output has two main reasons: The input voltage can only be adjusted stepwise, 0.025 V per motorstep, and the setting of the second potentiometer is always delayed until the first input is set. The delay in case of a 0.5 volt step for input 1 is about 0.3 seconds. Further error sources are the inaccuracy of model (6), the ramp- instead of step-function produced by the stepping motor and measurement noise.

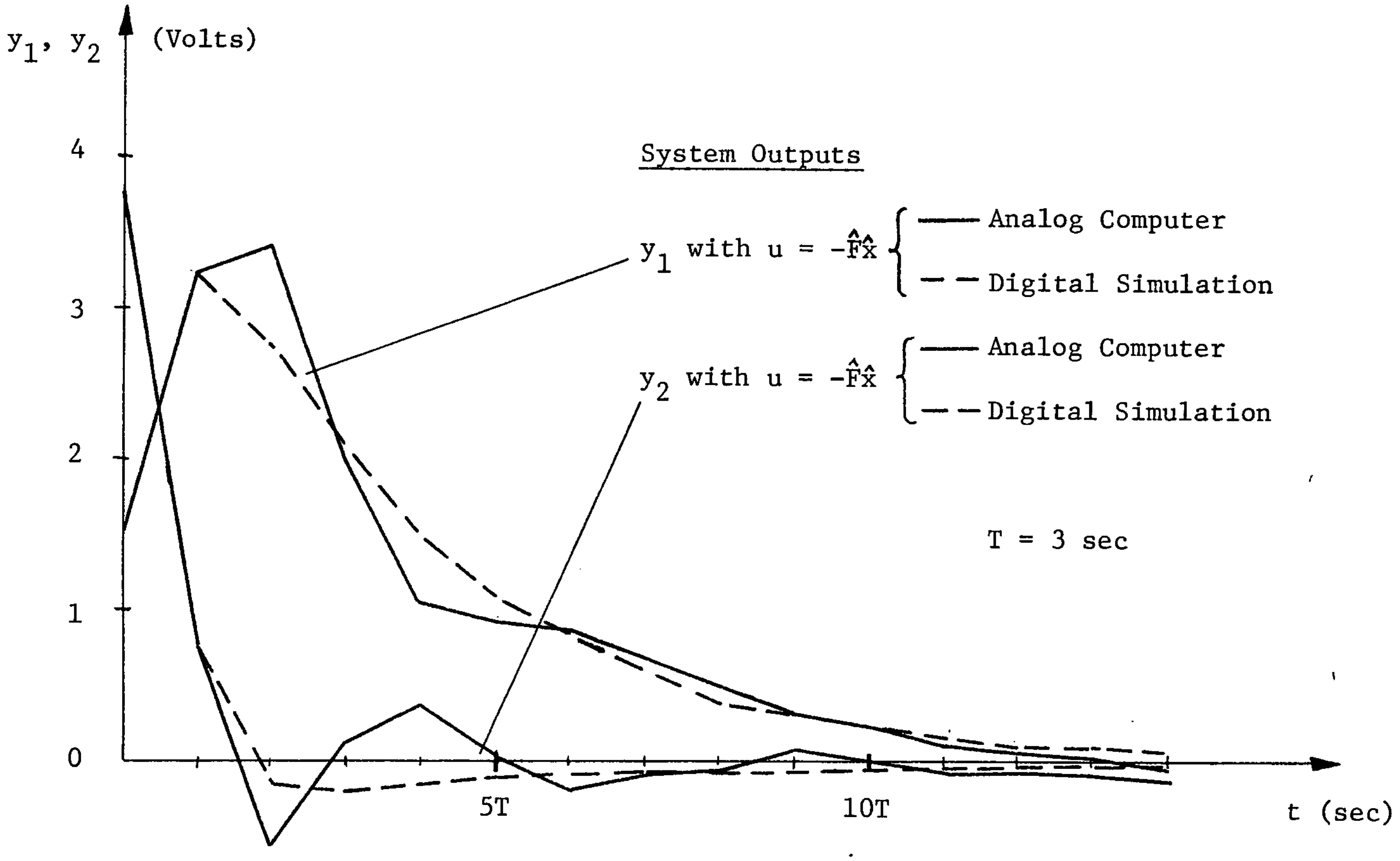

Figure 5.3.

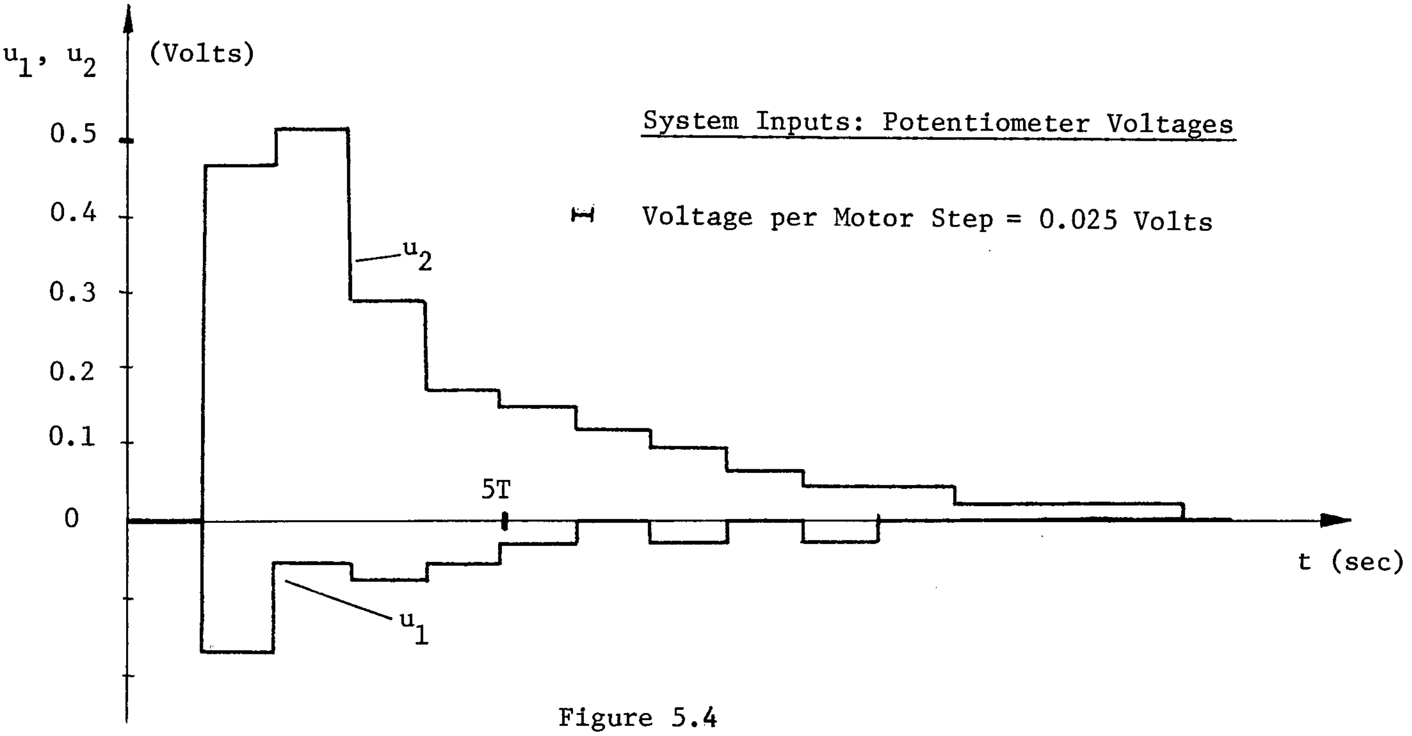

Figure 5.4

VI. CONCLUSIONS

The purpose of this thesis is to identify multi-input multi-output linear systems on-line on a minicomputer. The selector matrix method of Gopinath is employed for the derivation of two new identification methods. Both methods are successfully tested with different input sequences for various systems on a large computer. The implementation of method B on a NOVA minicomputer for the identification of systems simulated on an analog computer gives an estimated 50% storage saving in comparison with Gopinath's method and very accurate results. The main features of the two methods are:

1) Less computational requirements (less storage, more accuracy) than Gopinath's method.

2) Simple input sequences can be used. The few existing restrictions prove to be insignificant for practical applications.

3) Each column of input matrix $\hat{B}$ is identified from different data. In method B the identification of the state transition matrix $\hat{A}$ and the input matrix $\hat{B}$ are completely decoupled.

4) Some restrictions for the initial system state exist but they can be satisfied quite easily.

5) The selector matrix has to be known in advance. Two classes of selector matrixes that lead to canonical models are proposed and tested.

The main problem that remains to be solved is the extension of the proposed methods to the case of noise contaminated data. Both methods as well as Gopinath's procedure lead to a system of linear equations as shown below

$$D_1 = PD_2 + V \tag{1}$$

where D_1 and D_2 contain measured data, P is the array of unknown fixed parameters and V is an array of measurement noise terms. The classical method of least squares computes the estimate $\hat{P}$ such that the cost function

$$J = \text{trace } (D_1 - \hat{P}D_2)\ (D_1 - \hat{P}D_2)' \tag{2}$$

is minimized. However, even independent measurement errors (additive noise), i.e.

$$y(k) = Hx(k) + n(k)$$

result in correlated residual vectors in (1). The reason for this is that any component of the output vector y(k) may appear in more than one column of D_2. As a consequence of correlated residual vectors the least squares estimate will be biased. This bias can be eliminated by the use of one of the many sophisticated estimation methods described in [9], [13] and [14].

Another remaining problem is the finding of the selector matrix. An efficient algorithm to determine selector matrices from noisy data would certainly make the proposed identification methods more useful. The question of "optimal" input sequences with respect to "good" parameter estimates is another problem that requires further investigation.

REFERENCES

[1] Gopinath, B., "On the Identification of Linear Time-Invariant Systems from Input-Output Data," Bell. Syst. Tech. J., Vol. 48, pp. 1101-1113, May 1969.

[2] Budin, M.A., "Minimal Realization of Discrete Linear Systems from Input-Output Observations", IEEE Trans. Automat. Contr., Vol. AC-16, pp. 395-401, Oct. 1971.

[3] Bonivento, C., Guirdozi R. and Marro G., "Irreducible Canonical Realizations from External Data Sequences", Int. J. Control, Vol. 17, No. 3, pp. 553-563, 1973.

[4] Ackermann, J´.E., "Die Minimale Ein-Ausgangs Beschreibung von Mehrgrössensystemen und ihre Bestimmung aus Ein-Ausgangs Messungen", Regelungstechnik, Heft 5, pp. 203-206, 1971.

[5] Luenberger, D.G., "Observers for Multivariable Systems", IEEE Trans. Automat. Contr., Vol. AC-11, p.p. 190-197, April 1966.

[6] Bucy, R.S., "Canonical Forms for Multivariable Systems", IEEE Trans. Automat. Contr., Vol. AC-13, p.p. 567-569, Oct. 1968.

[7] Tapp, R.J., "A Parameter Estimation Algorithm for Small Digital Computers", Master's Thesis Dept. of Electrical Engineering, UBC, July 1972.

[8] Bingulac, S.P. and Djorović, M., "Estimation of State Variable Models Using Input-Output Data", IFAC Symp. on Estimation, TS-14, pp. 995-1004, June 1973.

[9] Åström, K.J. and Eykhoff, P., "System Identification - A Survey". Automatica, Vol. 7, pp. 123-162, 1971.

[10] Ackermann, J.E., "Abtastregelung", Springer Verlag, New York 1972.

[11] Kalman, R.E., "Mathematical Description of Linear Dynamical Systems", SIAM J. on Control, Vol. 1, No. 2, pp. 152-192, 1963.

[12] Kwakernaak, H. and Sivan, R., "Linear Optimal Control Systems", John Wiley and Sons, New York, 1972.

[13] Isermann, R., "Comparison of Six On-Line Identification and Parameter Estimation Methods", Automatica, Vol. 10, pp. 81-103, 1974.

[14] Saridis, G.N., "Comparison of On-Line Identification Algorithms", Automatica, Vol. 10, pp. 69-79, 1974.

[15] Gopinath, B., "On the Control of Linear Multiple Input-Output Systems", Bell. Syst. Tech. J., Vol. 50, pp. 1063-1081, March 1971.

APPENDIX 1

Proof of Theorem 1:

Assuming that

$$H = \begin{bmatrix} \text{---}h_1\text{---} \\ \text{---}h_2\text{---} \\ \vdots \\ \text{---}h_\ell\text{---} \end{bmatrix} \tag{1}$$

we get for (15)

$$S \begin{bmatrix} H \\ HA \\ \vdots \\ HA^{n^*-1} \end{bmatrix} = \left[\begin{array}{c} \text{---}h_1\text{---} \\ \vdots \\ h_1A^{n_1-1} \\ \hline \text{---}h_2\text{---} \\ \vdots \\ \hline \text{---}h_\ell\text{---} \\ \vdots \\ h_\ell A^{n_\ell-1} \end{array}\right] \tag{2}$$

where

$$\sum_{i=1}^{\ell} n_i = n \text{ and } n^* \geq n_i \geq 1 \text{ for } i=1,\ldots\ell$$

Luenberger [5] has shown that for a completely observable system (A,B,H) with $\rho[H] = \ell$ and observability coefficient n* there exists a sequence $n_1, n_2, \ldots n_\ell$ such that T given by (15) is nonsingular q.e.d.

Proof of Theorem 2:

Using the known relation

$$S\overline{y}(k) = S \begin{bmatrix} H \\ HA \\ \vdots \\ HA^{n^*-1} \end{bmatrix} x(k) + S \begin{bmatrix} 0\ldots & & \\ HB & 0.. & \\ & & \\ HA^{n^*-2}\ldots & & 0 \end{bmatrix} \overline{u}(k) \tag{3}$$

$$= Tx(k) + SR_1\overline{u}(k)$$

we get

$$U_{n*}(k) = \left[\begin{array}{c|c} T & SR_1 \\ \hline 0 & I_{mn*} \end{array}\right] \left[\begin{array}{c} x(k)\ldots\ x(k+n+mn*-1) \\ \hline \bar{u}(k)\ldots\ \bar{u}(k+n+mn*-1) \end{array}\right] = QP \qquad (4)$$

The (n+mn*) x (n+mn*) matrix Q is nonsingular because T is nonsingular (15) and therefore

$$\rho[U_{n*}(k)] = \rho[P]$$

The rest of the proof follows from Lemma 2 in Gopinath's paper [1].

<u>Definition of a Cyclic Matrix</u>:

C is an m x m matrix and u(k) is m x 1. C is cyclic if there exists u(k) such that the matrix

[illegible] r $[u(k)\ Cu(k)....C^{m-1}u(k)]$

is nonsingular.

APPENDIX 2

The Recursive Least Squares Algorithm of Tapp [7].

The relation between a collection of input-output data and model parameters was shown in (72) to be given by

$$Z = \underline{\hat{A}}D + V \qquad (1)$$

where

$Z = z_1 \; z_2 \ldots z_p$ and $D = d_1 \; d_2 \ldots d_p$ are s x p

$\underline{\hat{A}}$ is an s x s matrix of unknown fixed parameters and V is an array of noise terms. The least squares estimate $\underline{\hat{A}}_e$ of the parameters $\underline{\hat{A}}$ minimizes the cost

$$J = \text{trace } (Z - \underline{\hat{A}}_e D)\ (Z - \underline{\hat{A}}_e D)' \qquad (2)$$

and can be computed recursively as follows with k = 1,2....p

$$c_k = (I - Q_{k-1})\ d_k \qquad (3)$$

$$e_k = c_k\ (c_k' c_k)^{-1} \qquad (4)$$

$$Q_k = Q_{k-1} + e_k c_k' \ , \ Q_k = 0 \text{ at } k = 0 \qquad (5)$$

$k \leq s$

$$R_k = R_{k-1} - R_{k-1} d_k e_k' - e_k d_k' R_{k-1} + e_k d_k' R_{k-1} d_k e_k' + e_k e_k' \ ,$$
$$R_k = 0 \text{ at } k = 0 \qquad (6)$$

$$\underline{\hat{A}}_{e,k} = \underline{\hat{A}}_{e,k-1} + e_k\ (z_k' - d_k' \underline{\hat{A}}_{e,k-1})\ ,\ \underline{\hat{A}}_{e,k} = 0 \text{ at } k = 0 \qquad (7)$$

$$b_k = P_{k-1} d_k\ (1 + d_k' P_{k-1} d_k)^{-1} \qquad (8)$$

$p \geq k > s$

$$P_k = P_{k-1} - b_k d_k' P_{k-1} \ , \ P_k = R_k \text{ at } k = s \qquad (9)$$

$$\underline{\hat{A}}_{e,k} = \underline{\hat{A}}_{e,k-1} + b_k\ (z_k' - d_k' \underline{\hat{A}}_{e,k-1})\ ,\ \underline{\hat{A}}_{e,k} = \underline{\hat{A}}_{e,k}$$
$$\text{from (7) at } k = s \qquad (10)$$

Whenever the first s data vectors $d_1, d_2 \ldots d_s$ are linearly dependent, the equation (3) will give

$$c_k = (I - Q_{k-1})\ d_k = 0$$

Since the recursive least squares procedure and condition (56) require that $d_1, d_2 \ldots d_s$ are linearly independent, the value of c_k for a new data vector is an extremly useful indicator of linear independency.

APPENDIX 3

Suboptimal Control:

The solution of the optimal control problem for the deterministic system

$$x(k + 1) = A\ x(k) + B\ u(k) \qquad (1)$$

with the criterion

$$J = x'(N)Q_0x(N) + \mathop{}\limits_{k=0}^{N-1} x'(k)Q_1x(k) + u'(k)Q_2u(k) \qquad (2)$$

is a time dependent feedback

$$u(k) = -F(k)\ x(k) \qquad (3)$$

where

$$\begin{aligned} F(k) &= [Q_2 + B'S(k+1)B]^{-1}B'S(k+1)A \\ S(k) &= [A - BF(k)]'S(k+1)A + Q_1 \qquad (4) \\ S(N) &= Q_0 \end{aligned}$$

The matrices Q_0 and Q_1 be symmetric and nonnegative, the matrix Q_2 be positive definite.

It is known that F(k) and S(k) asymptotically approach a "steady state" value for decreasing k. The suboptimal control

$$u(k) = -F(0)\ x(k) = -F\ x(k) \qquad (5)$$

is a good approximation of (3) because F(k) departs significantly from its asymptotic value only over the terminal stages of the process when x(k) is already small. The larger the N the better the approximation will be. The relative increase of the cost J for suboptimal control of the system in experiment 3 was computed. It was less than 3% for N = 15.

Observer:

The particular structure of $(\hat{A},\hat{B},\hat{H})$ described in (18) gives a very simple state observer. At stage $k + n^* - 1$ the past model state $\hat{x}(k)$ is found from (22)

$$\hat{x}(k) = S\ \bar{y}(k) - SR_1\ \bar{u}(k)$$

where $\bar{u}(k)$ and $\bar{y}(k)$ are defined in (20) and SR_1 has the form in (23). By repeated use of

$$\hat{x}(k + 1) = \hat{A}\ \hat{x}(k) + \hat{B}\ u(k) \quad \text{and} \quad u(k), u(k+1) \ldots u(k+n^*-2)$$

we get the present model state $x(k+n^*-1)$. Note that the first observation is possible at stage $n^* - 1$ of the process, therefore an n^* as small as possible should be chosen for the identification (see experiment 1).