



Deposited via The University of Sheffield.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/id/eprint/78100/>

---

**Monograph:**

Chen, S., Billings, S.A. and Luo, W. (1988) Orthogonal Least Squares Methods and Their Application to Nonlinear System Identification. Research Report. Acse Report 343 . Dept of Automatic Control and System Engineering. University of Sheffield

---

**Reuse**

Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

**Takedown**

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing [eprints@whiterose.ac.uk](mailto:eprints@whiterose.ac.uk) including the URL of the record and the reason for the withdrawal request.

629.8 (S)  
PAM BOX

**ORTHOGONAL LEAST SQUARES METHODS AND THEIR  
APPLICATION TO NONLINEAR SYSTEM IDENTIFICATION**

S. Chen, S.A. Billings and W. Luo

Department of Control Engineering

University of Sheffield

Mappin Street

Sheffield S1 3JD

U.K.

Research Report No.343

August 1988

**ORTHOGONAL LEAST SQUARES METHODS AND THEIR  
 APPLICATION TO NONLINEAR SYSTEM IDENTIFICATION**

**Table of Contents**

Abstract .....	2
1. Introduction .....	2
2. Nonlinear system identification and linear least squares problems .....	3
3. Review of methods for solving least squares problems .....	5
3.1. Methods based on the normal equation .....	6
3.2. Methods based on orthogonal decomposition of $P$ .....	8
3.3. Singular value decomposition of $P$ .....	13
4. Selection of subset models using orthogonal algorithms .....	14
4.1. Using the CGS procedure .....	15
4.2. Using the MGS procedure .....	17
4.3. Using the Householder transformation .....	18
5. Iterative schemes for polynomial NARMAX models .....	20
6. Simulation study .....	22
7. Conclusions .....	24
Acknowledgments .....	24
References .....	25
Tables .....	27
Figures .....	30

## Abstract

Identification algorithms which combine structure determination and parameter estimation are developed based on well-known orthogonal least squares methods. These algorithms can provide information about the structure, or which terms to include in the model, and final parameter estimates in a very simple and efficient manner for a class of multivariable discrete-time nonlinear stochastic systems which are linear in the parameters.

## 1. Introduction

Most systems encountered in the real world are nonlinear and since linear models cannot capture the rich dynamic behaviour of limit cycles, bifurcations etc associated with nonlinear systems it is important to investigate the development of identification procedures for nonlinear models. The NARMAX (Nonlinear AutoRegressive Moving Average with eXogenous inputs) model which was introduced by Leontaritis and Billings (1985) provides a basis for such a development. When a polynomial expansion of the NARMAX model is selected the model becomes linear-in-the-parameters. Providing the model structure, or which terms to include in the model has been determined, only the values of the parameters are unknown and the identification can thus be formulated as a standard least squares problem which can be solved using various well-developed numerical techniques. Unfortunately the model structure of real systems is rarely known a priori and methods of model structure determination must therefore be developed and included as a vital part of the identification procedure. Because the number of all possible candidate terms can easily run into several thousands even for 'moderately' nonlinear systems optimal multiple selection methods (Leontaritis and Billings, 1987) are difficult to use and suboptimal methods of structure determination such as stepwise regression (Draper and Smith, 1981) become very complicated and time consuming.

An orthogonal algorithm which efficiently combines structure selection and parameter estimation has been derived (Korenberg, 1985) and later extended to the identification of single-input single-output nonlinear stochastic systems (Korenberg et al, 1988). A more reliable version of the algorithm was developed in (Billings, Korenberg and Chen, 1988) and applied to multi-input multi-output nonlinear stochastic systems (Billings, Chen and Korenberg, 1988). Various simulation studies and practical applications have shown that this algorithm provides a simple and powerful means of fitting parsimonious models to real systems. A similar structure

selection algorithm incorporating some statistical tests for nonlinear models without noise modelling was reported in Kortmann and Unbehauen (1988). A slightly different structure determination algorithm using projection matrices (symmetric and orthogonal matrices) was given by Desrochers and Mohseni (1984).

Starting with a review of methods for solving least squares problems, the present study develops structure selection algorithms for the polynomial NARMAX model by modifying and augmenting some well-known techniques of orthogonal decomposition of the regression matrix. It is shown that the orthogonal algorithms developed by the authors (Korenberg et al, 1988; Billings, Chen and Korenberg, 1988) belong to this type. Advantages and disadvantages of using the different orthogonal decomposition techniques are discussed and a comparison of the resulting structure selection algorithms is given.

## 2. Nonlinear system identification and linear least squares problems

Under some mild assumptions a discrete-time multivariable nonlinear stochastic control system with  $m$  outputs and  $r$  inputs can be described by the NARMAX model (Leontaritis and Billings, 1985)

$$y(t) = f(y(t-1), \dots, y(t-n_y), u(t-1), \dots, u(t-n_u), e(t-1), \dots, e(t-n_e)) + e(t) \quad (1)$$

where

$$y(t) = \begin{bmatrix} y_1(t) \\ \vdots \\ y_m(t) \end{bmatrix}, \quad u(t) = \begin{bmatrix} u_1(t) \\ \vdots \\ u_r(t) \end{bmatrix}, \quad e(t) = \begin{bmatrix} e_1(t) \\ \vdots \\ e_m(t) \end{bmatrix} \quad (2)$$

are the system output, input and noise respectively;  $n_y$ ,  $n_u$  and  $n_e$  are the maximum lags in the output, input and noise;  $\{e(t)\}$  is a zero mean independent sequence; and  $f(\cdot)$  is some vector-valued nonlinear function. Equation (1) can be decomposed into  $m$  scalar equations as follows

$$y_i(t) = f_i(y_1(t-1), \dots, y_1(t-n_y), \dots, y_m(t-1), \dots, y_m(t-n_y), u_1(t-1), \dots, u_1(t-n_u), \dots, u_r(t-1), \dots, u_r(t-n_u), e_1(t-1), \dots, e_1(t-n_e), \dots, e_m(t-1), \dots, e_m(t-n_e)) + e_i(t) \quad i=1, \dots, m \quad (3)$$

A special case of the general NARMAX model eqn.(1) is the NARX (Nonlinear AutoRegressive with eXogenous inputs) model

$$y(t) = f(y(t-1), \dots, y(t-n_y), u(t-1), \dots, u(t-n_u)) + e(t) \quad (4)$$

or

$$y_i(t) = f_i(y_1(t-1), \dots, y_1(t-n_y), \dots, y_m(t-1), \dots, y_m(t-n_y), u_1(t-1), \dots, u_1(t-n_u), \dots, u_r(t-1), \dots, u_r(t-n_u)) + e_i(t) \quad i=1, \dots, m \quad (5)$$

In reality the nonlinear form of  $f_i(\cdot)$  in eqn.(5) is generally unknown. Any continuous  $f_i(\cdot)$  however can be arbitrarily well approximated by polynomial models (Chen and Billings, 1988b). Expanding  $f_i(\cdot)$  as a polynomial of degree  $l$  gives the representation

$$y_i(t) = \theta_0^{(i)} + \sum_{i_1=1}^n \theta_{i_1}^{(i)} x_{i_1}(t) + \sum_{i_1=1}^n \sum_{i_2=i_1}^n \theta_{i_1 i_2}^{(i)} x_{i_1}(t) x_{i_2}(t) + \dots + \sum_{i_1=1}^n \dots \sum_{i_l=i_{l-1}}^n \theta_{i_1 \dots i_l}^{(i)} x_{i_1}(t) \dots x_{i_l}(t) + e_i(t) \quad i=1, \dots, m \quad (6)$$

where

$$n = mn_y + rn_u \quad (7)$$

and

$$\left. \begin{aligned} x_1(t) &= y_1(t-1), x_2(t) = y_1(t-2), \dots, x_{mn_y}(t) = y_m(t-n_y) \\ x_{mn_y+1}(t) &= u_1(t-1), \dots, x_n(t) = u_r(t-n_u) \end{aligned} \right\} \quad (8)$$

It is clear that each subsystem model in eqn.(6) belongs to the linear regression model

$$z(t) = \sum_{i=1}^M p_i(t) \theta_i + \xi(t) \quad t=1, \dots, N \quad (9)$$

where  $N$  is the data length, the  $p_i(t)$ 's are monomials of  $x_1(t)$  to  $x_n(t)$  up to degree  $l$  ( $p_i(t)=1$  corresponding to a constant term),  $\xi(t)$  is some modelling error, and the  $\theta_i$ 's are unknown parameters to be estimated. In linear regression analysis  $z(t)$  is known as the dependent variable and the  $p_i(t)$ 's are often referred to as regressors or predictors. Eqn.(9) can be written in the matrix form

$$\mathbf{z} = \mathbf{P}\boldsymbol{\Theta} + \boldsymbol{\Xi} \quad (10)$$

with

$$\mathbf{z} = \begin{bmatrix} z(1) \\ \vdots \\ z(N) \end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix} \mathbf{p}_1 & \dots & \mathbf{p}_M \end{bmatrix}, \quad \boldsymbol{\Theta} = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_M \end{bmatrix}, \quad \boldsymbol{\Xi} = \begin{bmatrix} \xi(1) \\ \vdots \\ \xi(N) \end{bmatrix} \quad (11)$$

and

$$\mathbf{p}_i = \begin{bmatrix} p_i(1) \\ \vdots \\ p_i(N) \end{bmatrix} \quad i=1, \dots, M \quad (12)$$

In reality each subsystem in eqn.(6) may involve only a few significant terms which adequately characterize the system dynamics. If the significant terms are known a priori and only they are used to form the regression matrix  $\mathbf{P}$  a linear least squares problem can be defined:

$$\text{find parameter estimate } \hat{\boldsymbol{\Theta}} \text{ which minimizes } \|\mathbf{z} - \mathbf{P}\hat{\boldsymbol{\Theta}}\| \quad (13)$$

where  $\|\cdot\|$  is the Euclidean norm. It is well-known that the solution to this problem satisfies the normal equation

$$\mathbf{P}^T \mathbf{P} \boldsymbol{\theta} = \mathbf{P}^T \mathbf{z} \quad (14)$$

where  $\mathbf{P}^T \mathbf{P}$  is called the information matrix. Several numerical methods of solving the least squares problem are summarized in Section 3.

Because the system structure is generally unknown at the beginning of the identification, the experimenter is often forced to consider the full model set, that is all possible terms in eqn.(6). The determination of the structure or which terms to include in the final model is essential since a full model set can easily involve an excessive number of terms, most of which may be redundant and should be removed. A parsimonious model is highly desirable if the model is to be employed in controller design, prediction and other applications. Let  $\mathbf{P}$  represent the full model set. The combined problem of structure selection and parameter estimation can then be stated as follows:

$$\left. \begin{array}{l} \text{Select a subset } \mathbf{P}_s \text{ of } \mathbf{P} \text{ and find the corresponding parameter} \\ \text{estimate } \hat{\boldsymbol{\theta}}_s \text{ which adequately fits the data} \end{array} \right\} \quad (15)$$

One possible approach to the above problem is to use some optimal multiple selection methods based on the theory of hypothesis testing (Leontaritis and Billings, 1987). Because the number of all the possible terms can easily become excessively large it is very difficult to attain the optimal solution since this would involve examining all the possible subset models. Some suboptimal methods have to be employed and Section 4 considers a class of suboptimal algorithms based on the orthogonal decomposition of  $\mathbf{P}$ .

So far only a polynomial expansion of the NARX model eqn.(4) has been discussed. If the same expansion is applied to the NARMAX model eqn.(1) a similar linear-in-the-parameters expression for eqn.(10) is obtained. Unlike the polynomial NARX case, however, now not all the columns of  $\mathbf{P}$  can be measured or formed from the measurements directly, and eqn.(10) becomes a pseudolinear regression model. In Section 5, the results of Section 4 are extended to the polynomial NARMAX model.

### 3. Review of methods for solving least squares problems

This section reviews numerical methods of solving the least squares problem of eqn.(13). There are three approaches which may be considered competitive for computing  $\hat{\boldsymbol{\theta}}$ :

- (a) Solve the normal equation by Gaussian elimination or by forming the Cholesky decomposition of  $\mathbf{P}^T\mathbf{P}$ .
- (b) Form an orthogonal decomposition of  $\mathbf{P}$ .
- (c) Form a singular value decomposition of  $\mathbf{P}$ .

Each of these approaches has advantages. If  $\mathbf{P}^T\mathbf{P}$  can be formed accurately (a) offers the most economical way of computing  $\hat{\theta}$  at about half the cost of the second approach (b) and one-quarter to one-eighth of the cost of the third (c). The second approach is generally the most accurate. It avoids possible ill-conditioning from the formation of  $\mathbf{P}^T\mathbf{P}$ . The orthogonal decomposition may be carried out via (modified) Gram-Schmidt orthogonalization, Householder transformation or Givens method. Method (c) is particularly useful when the rank of  $\mathbf{P}$  is unknown or when  $\mathbf{P}$  is of full rank but is ill-conditioned in an unpredictable way. This method is computationally more expensive. Throughout the discussion in this section it is assumed that  $\mathbf{P}$  has the dimension  $N \times M$  with  $M \leq N$ .

### 3.1. Methods based on the normal equation

Assume that  $\mathbf{P}$  is of full rank, then

$$\mathbf{B} = \mathbf{P}^T\mathbf{P} \tag{16}$$

is positive definite. Gaussian elimination reduces  $\mathbf{B}$  to an upper triangular matrix with positive diagonal elements. The reduction is achieved by a series of nonsingular elementary row transformations in which multiples of each row of  $\mathbf{B}$  are successively subtracted from the rows below to give zeros below the diagonal. Performing these transformations on the augmented matrix  $[\mathbf{B}:\mathbf{P}^T\mathbf{z}]$  gives rise to  $[\bar{\mathbf{V}}:\mathbf{d}]$  where

$$\bar{\mathbf{V}} = \begin{bmatrix} \bar{v}_{11} & \bar{v}_{12} & \bar{v}_{13} & \cdots & \bar{v}_{1M} \\ 0 & \bar{v}_{22} & \bar{v}_{23} & \cdots & \bar{v}_{2M} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & \bar{v}_{MM} \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} d_1 \\ d_2 \\ \cdot \\ \cdot \\ d_M \end{bmatrix} \tag{17}$$

The elements of  $\hat{\theta}$  can readily be found by backward substitution

$$\left. \begin{aligned} \hat{\theta}_M &= \frac{d_M}{\bar{v}_{MM}} \\ \hat{\theta}_i &= \frac{d_i - \sum_{k=i+1}^M \bar{v}_{ik} \hat{\theta}_k}{\bar{v}_{ii}} \quad i=M-1, \dots, 1 \end{aligned} \right\} \tag{18}$$

If elementary row transformations are processed further  $B$  can be reduced to the identity matrix  $I$  and the same transformations performed on  $[I:P^Tz]$  will lead to  $[B^{-1}:\hat{\Theta}]$ . This is known as Jordan elimination.

The Cholesky decomposition method expresses  $B$  uniquely in the form

$$B = U^T U \quad (19)$$

where  $U$  is an upper triangular matrix with positive diagonal elements.  $\hat{\Theta}$  can then be found by solving two triangular systems

$$\left. \begin{aligned} U^T q &= P^T z \\ U \Theta &= q \end{aligned} \right\} \quad (20)$$

using the forward and backward substitution algorithms. To avoid square root calculations in forming  $U$ , the information matrix  $B$  can alternatively be decomposed into

$$B = A^T D A \quad (21)$$

where  $A$  is an upper triangular matrix with unity diagonal elements and  $D$  is a positive diagonal matrix (Seber, 1977). Denote  $L = A^T$  and  $V = D A$  then

$$B = L V \quad (22)$$

The elements of  $L$  and  $V$  may be determined in  $M$  successive steps, and in each step a row of  $V$  and a column of  $L$  are computed

$$\left. \begin{aligned} v_{1j} &= b_{1j}, \quad j=1, \dots, M \\ l_{j1} &= \frac{b_{j1}}{v_{11}}, \quad j=2, \dots, M \\ v_{ij} &= b_{ij} - \sum_{k=1}^{i-1} l_{ik} v_{kj}, \quad j=i, i+1, \dots, M \\ l_{ji} &= \frac{b_{ji} - \sum_{k=1}^{i-1} l_{jk} v_{ki}}{v_{ii}}, \quad j=i+1, \dots, M \end{aligned} \right\} \quad \left. \begin{aligned} & \\ & \\ & \\ & \end{aligned} \right\} \quad \left. \begin{aligned} & \\ & \\ & \\ & \end{aligned} \right\} \quad i=2, \dots, M \quad (23)$$

If  $P^T P$  can be formed accurately, the methods based on  $P^T P$  are computationally the cheapest to implement. Forming  $P^T P$  however introduces roundoff errors; and if  $P$  is ill-conditioned, that is a small change in the elements of  $P$  can cause large changes in  $(P^T P)^{-1}$  and hence  $\hat{\Theta} = (P^T P)^{-1} P^T z$ , any errors in the formation of  $P^T P$  may have a serious effect on the stability of the least squares solution. Furthermore roundoff errors accumulate in the process of solving  $\hat{\Theta}$  and this makes the situation even worse. The problem of ill-conditioning frequently occurs in polynomial nonlinear models where the columns of  $P$  can often be highly correlated. As an illustration consider the example of a polynomial model with a single variable up to  $k$ 'th-degree (Seber, 1977,

p214)

$$z(t) = \sum_{i=0}^k \theta_i x^i(t) + e(t) \quad t=1, \dots, N \quad (24)$$

Assume that  $x(t)$  is distributed approximately uniformly on  $[0,1]$ , then for large  $N$  it can be shown that  $P^T P$  is approximately equal to the  $(k+1) \times (k+1)$  principal minor of the so-called Hilbert matrix

$$\tilde{H} = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \dots \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \dots \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix} \quad (25)$$

multiplied by  $N$ . It is well-known that  $\tilde{H}$  is very ill-conditioned. For example, let  $k=9$  and  $\tilde{H}_{10}$  be the  $10 \times 10$  principal minor of  $\tilde{H}$ , then the inverse of  $\tilde{H}_{10}$  has elements of magnitude  $3 \times 10^{10}$ . Thus a small error of  $10^{-10}$  in one element of  $P^T z$  will lead to an error of about 3 in an element of  $\hat{\theta}$ .

A measure of the ill-conditioning of  $P$  is its condition number  $\kappa[P]$  which is defined as

$$\kappa[P] = \text{ratio of the largest to smallest nonzero singular value of } P \quad (26)$$

The singular values of  $P$  are the nonnegative square roots of the eigenvalues of  $P^T P$ . Other definitions can also be used. Because

$$\kappa[P^T P] = \kappa^2[P] \quad (27)$$

and  $\kappa > 1$ ,  $P^T P$  is more ill-conditioned than  $P$ . Eqn.(27) indicates that, using  $k$ -digit binary arithmetic, we will not be able to obtain even an approximate solution to the normal equation (14) unless  $\kappa[P] \leq 2^{k/2}$  (Björck, 1967). This is clearly an unsatisfactory feature of the methods based on the normal equation. Therefore unless  $\kappa[P]$  is of moderate magnitude and  $P^T P$  can be formed accurately it is better not to form  $P^T P$  at all, and methods based on orthogonal decomposition of  $P$  are better alternatives. Although some authors (Golub, 1969; Wilkinson, 1974) have pointed out that the effect of  $\kappa^2[P]$  cannot be avoided entirely, solving least squares problems by forming an orthogonal decomposition of  $P$  is generally safer and more accurate than utilizing  $P^T P$ .

### 3.2. Methods based on orthogonal decomposition of $P$

As shown in Section 3.1 actually forming and solving the normal equation (14) numerically has serious drawbacks because  $P^T P$  is often so ill-conditioned that the

answer obtained is quite inaccurate. Now let  $F = PS$  where  $S$  is an  $M \times M$  nonsingular matrix. From eqn.(14) it follows

$$F^T P \hat{\theta} = F^T z \quad (28)$$

This equation can be used instead of the normal equation for solving for  $\hat{\theta}$  and  $F$  can be chosen in such a way that

$$\kappa[F^T P] = \kappa[P] \quad (29)$$

Indeed, since  $P$  is of full rank, it can be factorized as

$$P = QR \quad (30)$$

where  $Q$  is an  $N \times M$  orthogonal matrix ( $Q^T Q = I$ ), that is the columns of  $Q$  are orthonormal, and  $R$  is an  $M \times M$  upper triangular matrix. Choosing  $F = PR^{-1} = Q$ , the matrix  $F^T P = Q^T QR = R$  is triangular and eqn.(28) can easily be solved by backward substitution. Moreover the condition (29) is satisfied since

$$\kappa[F^T P] = \kappa[R] = \kappa[QR] = \kappa[P] \quad (31)$$

$R^T R$  is in fact the Cholesky decomposition of  $P^T P$ , eqn.(19). The factorization of eqn.(30) can be obtained in several ways and these are summarized in the following.

#### *Classical Gram-Schmidt*

In the actual computation it is preferable to use a different factorization of  $P$  rather than eqn.(30) in order to avoid computing square roots. The factorization that corresponds to the Cholesky decomposition of eqn.(21) is

$$P = WA \quad (32)$$

where

$$A = \begin{bmatrix} 1 & \alpha_{12} & \alpha_{13} & \cdots & \alpha_{1M} \\ & 1 & \alpha_{23} & \cdots & \alpha_{2M} \\ & & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot \\ & & & & \cdot \\ & & & & 1 & \alpha_{M-1M} \\ & & & & & 1 \end{bmatrix} \quad (33)$$

is an  $M \times M$  unit upper triangular matrix and

$$W = [w_1 \cdots w_M] \quad (34)$$

is an  $N \times M$  matrix with orthogonal columns that satisfy

$$W^T W = D \quad (35)$$

and  $D$  is the positive diagonal matrix in eqn.(21).

The classical Gram-Schmidt (CGS) procedure computes  $A$  one column at a time and orthogonalizes  $P$  as follows: at the  $k$ 'th stage make the  $k$ 'th column orthogonal to

each of the  $k-1$  previously orthogonalized columns and repeat the operations for  $k=2, \dots, M$ . The computational procedure is represented as

$$\left. \begin{aligned} \mathbf{w}_1 &= \mathbf{p}_1 \\ \alpha_{ik} &= \frac{\langle \mathbf{w}_i, \mathbf{p}_k \rangle}{\langle \mathbf{w}_i, \mathbf{w}_i \rangle} \quad 1 \leq i < k \\ \mathbf{w}_k &= \mathbf{p}_k - \sum_{i=1}^{k-1} \alpha_{ik} \mathbf{w}_i \end{aligned} \right\} k=2, \dots, M \quad (36)$$

where  $\langle \cdot, \cdot \rangle$  denotes the inner product, that is,

$$\langle \mathbf{w}_i, \mathbf{p}_k \rangle = \mathbf{w}_i^T \mathbf{p}_k = \sum_{l=1}^N w_i(l) p_k(l) \quad (37)$$

Define

$$\mathbf{g} = \mathbf{D}^{-1} \mathbf{W}^T \mathbf{z} \quad (38)$$

or

$$g_i = \frac{\langle \mathbf{w}_i, \mathbf{z} \rangle}{\langle \mathbf{w}_i, \mathbf{w}_i \rangle} \quad i=1, \dots, M \quad (39)$$

Then the parameter estimate  $\hat{\Theta}$  is readily computed from

$$\mathbf{A} \hat{\Theta} = \mathbf{g} \quad (40)$$

using backward substitution.

#### Modified Gram-Schmidt

It is well-known that the CGS procedure is very sensitive to roundoff errors. The experimental results (Rice, 1966) and the theoretical analysis (Björck, 1967) indicate that if  $\mathbf{P}$  is ill-conditioned, using CGS procedure, the computed columns of  $\mathbf{W}$  will soon lose their orthogonality and reorthogonalization will be needed. On the other hand, the modified Gram-Schmidt (MGS) procedure is numerically superior.

The MGS procedure calculates  $\mathbf{A}$  one row at a time and orthogonalizes  $\mathbf{P}$  as follows: at the  $k$ 'th stage make the columns subscripted  $k+1, \dots, M$  orthogonal to the  $k$ 'th column and repeat the operations for  $k=1, \dots, M-1$  (e.g. Björck, 1967). Specifically, denoting  $\mathbf{p}_i^{(0)} = \mathbf{p}_i$   $i=1, \dots, M$ , then

$$\left. \begin{aligned} \mathbf{w}_k &= \mathbf{p}_k^{(k-1)} \\ \alpha_{ki} &= \frac{\langle \mathbf{w}_k, \mathbf{p}_i^{(k-1)} \rangle}{\langle \mathbf{w}_k, \mathbf{w}_k \rangle} \quad i=k+1, \dots, M \\ \mathbf{p}_i^{(k)} &= \mathbf{p}_i^{(k-1)} - \alpha_{ki} \mathbf{w}_k \quad i=k+1, \dots, M \\ \mathbf{w}_M &= \mathbf{p}_M^{(M-1)} \end{aligned} \right\} k=1, 2, \dots, M-1 \quad (41)$$

The elements of  $\mathbf{g}$  are computed by transforming  $\mathbf{z}^{(0)} = \mathbf{z}$  in a similar way

$$\left. \begin{aligned} g_k &= \frac{\langle \mathbf{w}_k, \mathbf{z}^{(k-1)} \rangle}{\langle \mathbf{w}_k, \mathbf{w}_k \rangle} \\ \mathbf{z}^{(k)} &= \mathbf{z}^{(k-1)} - g_k \mathbf{w}_k \end{aligned} \right\} k=1,2,\dots,M \quad (42)$$

The CGS and MGS algorithms have distinct differences in computational behaviour. The MGS procedure is more accurate and more stable than the CGS procedure. This is particularly remarkable since both methods perform basically the same operations, only in a different sequence. Indeed, if there were no computer roundoff errors they would produce the same set of  $\mathbf{w}_i$ 's with the same number of operations.

### Householder transformation

An equivalent decomposition to eqn.(30) can be obtained by augmenting  $\mathbf{Q}$  with  $N-M$  further orthonormal columns to make up a full set of  $N$  orthonormal vectors for  $N$ -dimensional Euclidean space:

$$\tilde{\mathbf{Q}} = [\mathbf{Q} : \tilde{\mathbf{q}}_{M+1} \dots \tilde{\mathbf{q}}_N] = [\mathbf{Q}_M : \tilde{\mathbf{Q}}_{N-M}] \quad (43)$$

Then

$$\mathbf{P} = \tilde{\mathbf{Q}} \tilde{\mathbf{R}} = \tilde{\mathbf{Q}} \begin{bmatrix} \mathbf{R} \\ 0 \end{bmatrix} \quad (44)$$

where  $\mathbf{R}$  is the  $M \times M$  upper triangular matrix in eqn.(30), and  $\tilde{\mathbf{Q}}^T$  can be used to triangularize  $\mathbf{P}$ . If  $\tilde{\mathbf{Q}}^T \mathbf{z}$  is partitioned into

$$\tilde{\mathbf{Q}}^T \mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \left. \begin{array}{l} \} M \\ \} N-M \end{array} \right\} \quad (45)$$

we have

$$\|\mathbf{z} - \mathbf{P}\Theta\| = \|\tilde{\mathbf{Q}}^T(\mathbf{z} - \mathbf{P}\Theta)\| = \|z_1 - \mathbf{R}\Theta\| + \|z_2\| \quad (46)$$

The least squares estimates can therefore be obtained by solving the triangular system

$$\mathbf{R}\Theta = z_1 \quad (47)$$

and the sum of the squares of the residuals is given by  $\|z_2\|^2$ .

The Householder method triangularizes the  $N \times (M+1)$  augmented matrix

$$\tilde{\mathbf{P}} = [\mathbf{P} : \mathbf{z}] = \begin{bmatrix} \tilde{p}_{11} & \dots & \tilde{p}_{1M} & \tilde{p}_{1M+1} \\ \tilde{p}_{21} & \dots & \tilde{p}_{2M} & \tilde{p}_{2M+1} \\ \dots & \dots & \dots & \dots \\ \tilde{p}_{N1} & \dots & \tilde{p}_{NM} & \tilde{p}_{NM+1} \end{bmatrix} \quad (48)$$

to give

$$\tilde{\mathbf{P}}^{(M)} = \begin{bmatrix} \mathbf{R} & z_1 \\ 0 & z_2 \end{bmatrix} \quad (49)$$

using a series of Householder transformations

$$\mathbf{H}^{(k)} = \mathbf{I} - \beta^{(k)} \mathbf{v}^{(k)} (\mathbf{v}^{(k)})^T \quad k=1,\dots,M \quad (50)$$

Here the  $\mathbf{v}^{(k)}$ 's are  $N$ -vectors with certain properties, the  $\mathbf{H}^{(k)}$ 's are  $N \times N$  symmetric and orthogonal matrices, that is,  $(\mathbf{H}^{(k)})^T = \mathbf{H}^{(k)}$  and  $(\mathbf{H}^{(k)})^T \mathbf{H}^{(k)} = \mathbf{I}$ . Furthermore

$$\bar{\mathbf{Q}}^T = \mathbf{H}^{(M)} \mathbf{H}^{(M-1)} \dots \mathbf{H}^{(1)} \quad (51)$$

Denoting

$$\bar{\mathbf{P}}^{(k)} = \mathbf{H}^{(k)} \bar{\mathbf{P}}^{(k-1)}, \quad k=1, \dots, M \text{ and } \bar{\mathbf{P}}^{(0)} = \bar{\mathbf{P}} \quad (52)$$

the  $k$ 'th Householder transformation is explicitly defined as (e.g. Golub, 1965; Bierman, 1977, Chapter IV)

$$\left. \begin{aligned} \sigma^{(k)} &= \sqrt{\sum_{i=k}^N (\bar{p}_{ik}^{(k-1)})^2} \\ \beta^{(k)} &= \frac{1}{\sigma^{(k)} (\sigma^{(k)} + |\bar{p}_{kk}^{(k-1)}|)} \\ \mathbf{v}_i^{(k)} &= \begin{cases} 0 & \text{for } i < k \\ \bar{p}_{kk}^{(k-1)} + \text{sgn}[\bar{p}_{kk}^{(k-1)}] \sigma^{(k)} & \text{for } i = k \\ \bar{p}_{ik}^{(k-1)} & \text{for } i > k \end{cases} \end{aligned} \right\} \quad (53)$$

and

$$\bar{\mathbf{P}}^{(k)} = \bar{\mathbf{P}}^{(k-1)} - \mathbf{v}^{(k)} \left[ \beta^{(k)} (\mathbf{v}^{(k)})^T \bar{\mathbf{P}}^{(k-1)} \right] \quad (54)$$

### Givens method

A Givens transformation rotates two row vectors

$$\left. \begin{aligned} 0, \dots, 0, r_i, r_{i+1}, \dots, r_k, \dots \\ 0, \dots, 0, \beta_i, \beta_{i+1}, \dots, \beta_k, \dots \end{aligned} \right\} \quad (55)$$

resulting in two new row vectors

$$\left. \begin{aligned} 0, \dots, 0, \bar{r}_i, \bar{r}_{i+1}, \dots, \bar{r}_k, \dots \\ 0, \dots, 0, 0, \bar{\beta}_{i+1}, \dots, \bar{\beta}_k, \dots \end{aligned} \right\} \quad (56)$$

where

$$\left. \begin{aligned} \bar{r}_k &= cr_k + s\beta_k \\ \bar{\beta}_k &= -sr_k + c\beta_k \\ \bar{r}_i &= \sqrt{r_i^2 + \beta_i^2} \\ c &= r_i / \bar{r}_i \\ s &= \beta_i / \bar{r}_i \end{aligned} \right\} \quad (57)$$

There are two ways of applying a sequence of Givens transformations to triangularize  $\bar{\mathbf{P}}$  of eqn.(48).

For  $j=1, \dots, M$ , the first method rotates the  $j$ 'th row successively with the  $(j+1)$ 'th, ...,  $N$ 'th row so that the last  $N-j$  elements in the  $j$ 'th column are reduced to zeros.

This leads to the triangular system, of eqn.(49).  $z_2$  can be transformed further to  $(\delta, 0, \dots, 0)^T$ , then the sum of the squares of the residuals is simply  $\delta^2$ .

The second method processes one row of  $\tilde{P}$  at a time as follows:

$$\begin{bmatrix} \mathbf{R}^{(t-1)} & \mathbf{z}_1^{(t-1)} \\ & \delta^{(t-1)} \\ \tilde{p}_{t1}, \dots, \tilde{p}_{tM}, & \tilde{p}_{tM+1} \end{bmatrix} \rightarrow \begin{bmatrix} \mathbf{R}^{(t)} & \mathbf{z}_1^{(t)} \\ & \delta^{(t)} \\ 0, \dots, 0, & 0 \end{bmatrix} \quad (58)$$

for  $t=1, \dots, N$ . The initial  $\mathbf{R}^{(0)}$ ,  $\mathbf{z}_1^{(0)}$  and  $\delta^{(0)}$  can be taken as zeros. It is readily seen that this way of implementing Givens method leads to on-line identification algorithms (e.g. Result 7.2.2 in Goodwin and Payne, 1977). Similar to the Gram-Schmidt procedure, the computation of square roots in eqn.(57) can be avoided (Gentleman, 1973).

### Comparisons

MGS and Householder algorithms are highly competitive. The MGS algorithm is easier to program but requires slightly more computation. It seems that the MGS algorithm is slightly more accurate than the Householder algorithm (Jordan, 1968; Wampler, 1970).

The reduction of  $\mathbf{P}$  to upper triangular form requires approximately  $NM^2 - M^3/2$  multiplications and  $M$  square roots for the Householder method; and about  $2NM^2$  multiplications and  $NM$  square roots for the Givens method if transformations are carried out in the form of eqn.(57) (Seber, 1977). However the modified Givens method (Gentleman, 1973) requires no square roots and reduces multiplications to only three quarters of  $2NM^2$ . Furthermore the Givens method processes one row at a time and has important on-line implementations.

### 3.3. Singular value decomposition of $\mathbf{P}$

If the rank of  $\mathbf{P}$  is less than  $M$  the least squares problem no longer has a unique solution. Many of the methods discussed in Subsections 3.1 and 3.2 can be adapted to handle this. The singular value decomposition however offers a general solution to the least squares problem.

Assume that the rank of  $\mathbf{P}$  is  $\eta$  ( $\leq M$ ). According to the singular value decomposition theorem (e.g. Golub and Reinsch, 1970)  $\mathbf{P}$  can be factorized as

$$\mathbf{P} = \tilde{\mathbf{U}}\tilde{\mathbf{S}}\tilde{\mathbf{V}}^T \quad (59)$$

where  $\tilde{\mathbf{U}}$  is an  $N \times M$  orthogonal matrix consisting of  $M$  orthonormalized eigenvectors associated with the  $M$  largest eigenvalues of  $\mathbf{P}\mathbf{P}^T$ ,  $\tilde{\mathbf{V}}$  is an  $M \times M$  orthogonal matrix consisting of the orthonormalized eigenvectors of  $\mathbf{P}^T\mathbf{P}$ , and

$$\mathbf{S} = \text{diag}[s_1, \dots, s_M] \quad (60)$$

has the singular values of  $\mathbf{P}$  as its diagonal elements. The diagonal elements of  $\mathbf{S}$  are in the order

$$s_1 \geq s_2 \geq \dots \geq s_M \geq 0 \quad (61)$$

If  $\eta < M$ ,  $s_{\eta+1} = \dots = s_M = 0$ . There are alternative representation to eqn.(59) (e.g. Hanson and Lawson, 1969), however, eqn.(59) is convenient for computational purposes.

The pseudoinverse of  $\mathbf{P}$  is defined as

$$\mathbf{P}^+ = \tilde{\mathbf{V}}\mathbf{S}^+\tilde{\mathbf{U}}^T \quad (62)$$

where

$$\mathbf{S}^+ = \text{diag}[s_1^+, \dots, s_M^+] \quad (63)$$

and

$$s_i^+ = \begin{cases} 1/s_i & \text{for } s_i > 0 \\ 0 & \text{for } s_i = 0 \end{cases} \quad (64)$$

Then

$$\hat{\Theta} = \mathbf{P}^+\mathbf{z} = \tilde{\mathbf{V}}\mathbf{S}^+(\tilde{\mathbf{U}}^T\mathbf{z}) \quad (65)$$

is a solution of the least squares problem. Details of an algorithm for computing  $\hat{\Theta}$  using singular value decomposition can be found in (Golub, 1969; Golub and Reinsch, 1970).

#### 4. Selection of subset models using orthogonal algorithms

When identifying nonlinear systems with unknown structure, it is important to avoid losing significant terms which must be included in the final model, and consequently the experimenter is forced to start with large values of  $n_y$  and  $n_u$  and a high polynomial degree  $l$  in eqn.(4). The number of columns in  $\mathbf{P}$  is therefore often very large, even thousands perhaps. For example, even in the single-input single-output (SISO) case, if  $n_y=n_u=10$  and  $l=3$ ,  $\mathbf{P}$  has 1771 columns. Previous experience has shown that provided the significant terms in the model can be detected models with about ten terms are usually sufficient to capture the dynamics of highly nonlinear SISO processes (Billings, 1986; Billings and Fadzil, 1985; Billings et al, 1988; Billings, Chen and Backhouse, 1988). Throughout the rest of the discussion  $\mathbf{P}$  will be used to represent the full model set of a (sub)system so that  $M \leq N$  becomes unnecessary. This section considers the combined problem of structure selection and parameter estimation, defined by eqn.(15). It is shown that by augmenting the orthogonal decomposition techniques of Subsection 3.2, simple and efficient algorithms can be derived that determine  $\mathbf{P}_r$ , a subset of  $\mathbf{P}$ , in a forward-regression manner by choosing one column of

$P$  for which the sum of squares of residuals is maximally reduced at a time.

#### 4.1. Using the CGS procedure

Assume that  $P_s$  has  $M_s$  ( $M_s < M$  and  $M_s \leq N$ ) columns. Factorize  $P_s$  into  $W_s A_s$  as in eqn.(32) where  $W_s$  is an  $N \times M_s$  matrix consisting of  $M_s$  orthogonal columns and  $A_s$  is an  $M_s \times M_s$  unit upper triangular matrix. The residuals are defined by

$$\hat{\xi} = \begin{bmatrix} \xi(1) \\ \cdot \\ \cdot \\ \cdot \\ \xi(N) \end{bmatrix} = z - P_s \hat{\theta}_s = z - (P_s A_s^{-1})(A_s \hat{\theta}_s) = z - W_s g_s \quad (66)$$

Eqn.(66) can be rewritten as

$$z = W_s g_s + \hat{\xi} \quad (67)$$

The sum of squares of the dependent variable  $z$  is therefore

$$\langle z, z \rangle = \sum_{i=1}^{M_s} g_i^2 \langle w_i, w_i \rangle + \langle \hat{\xi}, \hat{\xi} \rangle \quad (68)$$

The error reduction ratio due to  $w_i$  is thus defined as the proportion of the dependent variable variance explained by  $w_i$

$$[err]_i = \frac{g_i^2 \langle w_i, w_i \rangle}{\langle z, z \rangle} \quad (69)$$

Eqn.(69) suggests a way of computing  $W_s$  (and hence  $P_s$ ) from  $P$  by the CGS procedure. At the  $i$ 'th stage, by interchanging the  $i$  to  $M$  columns of  $P$  we can select a  $p_i$  which gives the largest  $[err]_i$  when orthogonalized into  $w_i$ . The detailed procedure is as follows:

At the first stage, for  $i=1, \dots, M$ , denote  $w_1^i = p_i$  and compute

$$g_1^i = \frac{\langle w_1^i, z \rangle}{\langle w_1^i, w_1^i \rangle}; \quad [err]_1^i = \frac{(g_1^i)^2 \langle w_1^i, w_1^i \rangle}{\langle z, z \rangle}$$

Assume that  $[err]_1^j = \max\{[err]_1^i, 1 \leq i \leq M\}$ . Then  $w_1 = w_1^j (=p_j)$  is selected as the first column of  $W_s$ , together with the first element of  $g_s$ ,  $g_1 = g_1^j$  and  $[err]_1 = [err]_1^j$ .

At the second stage, for  $i=1, \dots, M$  and  $i \neq j$ , compute

$$\alpha_{12}^i = \frac{\langle w_1, p_i \rangle}{\langle w_1, w_1 \rangle}; \quad w_2^i = p_i - \alpha_{12}^i w_1$$

$$g_2^i = \frac{\langle w_2^i, z \rangle}{\langle w_2^i, w_2^i \rangle}; \quad [err]_2^i = \frac{(g_2^i)^2 \langle w_2^i, w_2^i \rangle}{\langle z, z \rangle}$$

Assume that  $[err]_2^k = \max\{[err]_2^i, 1 \leq i \leq M \text{ and } i \neq j\}$ . Then  $w_2 = w_2^k (=p_k - \alpha_{12}^k w_1)$  is selected as the second column of  $W_s$ , together with the second column of  $A_s$ ,  $\alpha_{12} = \alpha_{12}^k$ , the second element of  $g_s$ ,  $g_2 = g_2^k$  and  $[err]_2 = [err]_2^k$ .

The selection procedure is continued until the  $M_s$ 'th stage when

$$1 - \sum_{i=1}^{M_s} [err]_i < \rho \quad (70)$$

where  $\rho$  ( $0 < \rho \leq 1$ ) is a desired tolerance. Other criteria can also be used to stop the selection procedure, for example,

$$[err]_{M_s+1} \times 100 < a \text{ tolerance} \quad \text{or} \quad \frac{g_{M_s+1}^2 \langle \mathbf{w}_{M_s+1}, \mathbf{w}_{M_s+1} \rangle}{\langle \mathbf{L}, \mathbf{L} \rangle - \sum_{i=1}^{M_s} g_i^2 \langle \mathbf{w}_i, \mathbf{w}_i \rangle} \times 100 < a \text{ tolerance}$$

The subset model parameter estimate  $\hat{\Theta}_s$  can easily be computed from  $A_s \Theta_s = g_s$  by backward substitution.

It can easily be seen that the orthogonalization procedure used in (Korenberg, 1985; Korenberg et al, 1988; Billings, Korenberg and Chen, 1988; Billings, Chen and Korenberg, 1988) is the CGS procedure by simply comparing it with eqn.(36), and the algorithm of this subsection is in fact identical to the forward-regression orthogonal algorithm of Billings, Korenberg and Chen (1988), Billings, Chen and Korenberg (1988). Application of this orthogonal algorithm to the identification of the polynomial NARX model is straightforward since the identification of any subsystem is decoupled from the others. From eqn.s(68) and (69) it is seen that  $1 - \sum_{i=1}^{M_s} [err]_i$  is the proportion of the unexplained dependent variable variance. The value of  $\rho$  determines how many terms will be included in the final (sub)model and hence the complexity of the model. Let  $\rho_i$  be the desired tolerance for the  $i$ 'th subsystem. Ideal  $\rho_i$  should be closely related to  $E[e_i^2(t)]/E[y_i^2(t)]$ . Since the latter is not known a priori the appropriate  $\rho_i$  may have to be found by trial and error.

The criterion of eqn.(70) concerns only the performance of the model (variance of residuals) and does not take into account the model complexity. A criterion that compromises between the performance and complexity of the model is Akaike's information criterion  $AIC(\phi)$

$$AIC(\phi) = N \log C(\hat{\Theta}_s) + M_s \phi \quad (71)$$

where

$$C(\hat{\Theta}_s) = \frac{1}{N} \langle \hat{\mathbf{E}}, \hat{\mathbf{E}} \rangle \quad (72)$$

is the variance of the residuals, and  $\phi$  is the critical value of the chi-square distribution with one degree of freedom for a given significance level. To use this criterion the user is first required to specify a significance level. Leontaritis and Billings (1987) pointed out that  $\phi=4$  is a convenient choice and it corresponds to the significance level

of 0.0456. AIC(4) provides an alternative criterion to stop the above forward-regression model selection procedure. When the minimum of AIC(4) is reached the selection procedure is terminated. Other statistical criteria can also be employed to stop the selection, and the relationship between these criteria and AIC( $\phi$ ) has been investigated in (Söderström, 1977; Leontaritis and Billings, 1987).

As mentioned before the CGS orthogonalization is sensitive to computer roundoff errors. Notice however that the model selection algorithm in this subsection is not designed to orthogonalize the whole  $P$  which is often ill-conditioned. It selects  $P_s$ , usually a small subset of  $P$  and typically about ten columns for SISO systems. Since  $P_s$  contains only significant terms of the system it is usually well-conditioned and the problem of columns of  $W_s$  losing their orthogonality rarely occurs. Indeed this model selection algorithm performed well in many previous applications. Nevertheless it may be desirable to employ the MGS orthogonalization procedure in a similar model selection algorithm because of its numerical superiority.

Using the CGS procedure to select terms does have two important advantages which are worth emphasising. Storing the matrix  $P$  in the memory of a microcomputer could be a problem because its size is often huge. Notice that each column of  $P$  is a monomial function of the input-output data and the CGS algorithm computes one column of  $A$  and orthogonalizes a column of  $P$  at a time. Every time a column is to be orthogonalized it can be generated quickly from the input-output data. In this way storing  $P$  can be avoided and only  $W_s$ , which is often of modest dimensions, needs to be kept. Implementing the CGS algorithm in a microcomputer should therefore be straightforward. This will not be the case for MGS because storage for the whole  $P$  is required. The way that the CGS orthogonalization operates also makes the algorithm easier to extend to the polynomial NARMAX model where noise modelling is a part of the identification procedure. This is further discussed in Section 5.

#### 4.2. Using the MGS procedure

The development of a forward-regression orthogonal algorithm using MGS orthogonalization is straightforward. Because the MGS procedure makes the  $(k+1)$ 'th, ...,  $M$ 'th columns orthogonal to the  $k$ 'th column at the  $k$ 'th stage, the matrix  $P$  must be kept in computer memory. The memory space for  $A$ , up to  $M_s$  rows, must also be provided and this is of course much larger than the space for  $A_s$  required in the algorithm using CGS orthogonalization. Employing the same notations as eqn.s(41) and (42) and setting

$$\mathbf{P}^{(k-1)} = [ \mathbf{w}_1 \dots \mathbf{w}_{k-1} \mathbf{p}_k^{(k-1)} \dots \mathbf{p}_M^{(k-1)} ] \quad (73)$$

If some of the columns  $\mathbf{p}_k^{(k-1)}, \dots, \mathbf{p}_M^{(k-1)}$  in  $\mathbf{P}^{(k-1)}$  have been interchanged this will still be referred to as  $\mathbf{P}^{(k-1)}$  for notational convenience. The forward-regression orthogonal algorithm using the MGS orthogonalization can now be summarized as follows:

At the  $k$ 'th stage, for  $i=k, \dots, M$ , compute

$$g_k^{(i)} = \frac{\langle \mathbf{p}_i^{(k-1)}, \mathbf{z}^{(k-1)} \rangle}{\langle \mathbf{p}_i^{(k-1)}, \mathbf{p}_i^{(k-1)} \rangle}; \quad [err]_k^{(i)} = \frac{(g_k^{(i)})^2 \langle \mathbf{p}_i^{(k-1)}, \mathbf{p}_i^{(k-1)} \rangle}{\langle \mathbf{z}, \mathbf{z} \rangle}$$

Assume that  $[err]_k^{(j)} = \max\{[err]_k^{(i)}, k \leq i \leq M\}$ . Then the  $j$ 'th column of  $\mathbf{P}^{(k-1)}$  is interchanged with the  $k$ 'th column; the  $j$ 'th column of  $\mathbf{A}$  is interchanged up to the  $(k-1)$ 'th row with the  $k$ 'th column. The rest of the operations are as indicated in eqn.s(41) and (42). The selection procedure can be terminated in the same ways as discussed in Subsection 4.1. Notice that  $\mathbf{z}^{(M)}$  is simply the residual vector and

$$\frac{\langle \mathbf{z}^{(M)}, \mathbf{z}^{(M)} \rangle}{\langle \mathbf{z}, \mathbf{z} \rangle} = 1 - \sum_{i=1}^{M_s} [err]_i \quad (74)$$

$\mathbf{A}_s$  is the  $M_s \times M_s$  principal minor of  $\mathbf{A}$ .

### 4.3. Using the Householder transformation

The Householder transformation method can be employed to derive a forward-regression algorithm. Unfortunately as in the case of the MGS method, the whole of  $\mathbf{P}$  must be stored in computer memory. Denote

$$\tilde{\mathbf{P}}^{(0)} = [ \mathbf{P}; \mathbf{z} ] = [ \tilde{\mathbf{p}}_1^{(0)} \dots \tilde{\mathbf{p}}_M^{(0)}; \mathbf{z}^{(0)} ] \quad (75)$$

and  $\mathbf{R}_k$  the  $k \times k$  principal minor of  $\mathbf{R}$  where  $\mathbf{R}$  is defined in eqn.(44).

After  $\mathbf{H}^{(i)}$   $i=1, \dots, k-1$  have been successively applied to  $\tilde{\mathbf{P}}^{(0)}$ , it is transformed to

$$\tilde{\mathbf{P}}^{(k-1)} = \begin{bmatrix} \mathbf{R}_{k-1} & & \\ 0 & \tilde{\mathbf{p}}_k^{(k-1)} & \dots & \tilde{\mathbf{p}}_M^{(k-1)}; \mathbf{z}^{(k-1)} \end{bmatrix} \quad (76)$$

Two important properties of  $\mathbf{H}^{(k)}$  should be noticed:

- (i) It leaves the first  $k-1$  rows of  $\tilde{\mathbf{P}}^{(k-1)}$  unchanged.
- (ii) It leaves the column lengths invariant.

If the process were stopped at the  $(k-1)$ 'th stage and a subset model of  $k-1$  parameters were chosen, the sum of the squares of residuals would be

$$\sum_{i=k}^N [z_i^{(k-1)}]^2 \quad (77)$$

and this is reduced to

$$\sum_{i=k+1}^N [z_i^{(k)}]^2 = \sum_{i=k}^N [z_i^{(k-1)}]^2 - [z_k^{(k)}]^2 \quad (78)$$

after  $H^{(k)}$  has been applied to  $\bar{P}^{(k-1)}$ . The task is then to choose a column from  $\bar{p}_k^{(k-1)}, \dots, \bar{p}_M^{(k-1)}$  for which  $(z_k^{(k)})^2$  is maximized, and this can be achieved as follows:

Denote  $\bar{p}_j^{(k-1)} = (\bar{p}_{ij}^{(k-1)}, \dots, \bar{p}_{Nj}^{(k-1)})^T$   $j=k, \dots, M$ . Compute

$$a_j^{(k)} = \sqrt{\sum_{i=k}^N (\bar{p}_{ij}^{(k-1)})^2}, \quad b_j^{(k)} = \sum_{i=k}^N \bar{p}_{ij}^{(k-1)} z_i^{(k-1)}, \quad \text{for } j=k, \dots, M$$

Assume that the maximum of

$$\left[ z_k^{(k-1)} - (\bar{p}_{kj}^{(k-1)} + \text{sgn}[\bar{p}_{kj}^{(k-1)}] a_j^{(k)}) \times \frac{b_j^{(k)} + \text{sgn}[\bar{p}_{kj}^{(k-1)}] a_j^{(k)} z_k^{(k-1)}}{a_j^{(k)} (a_j^{(k)} + |\bar{p}_{kj}^{(k-1)}|)} \right]^2 = \left[ \frac{b_j^{(k)}}{a_j^{(k)}} \right]^2 \quad \text{for } j=k, \dots, M$$

is achieved at  $j = j_m$ . Then interchange the  $j_m$ 'th column of  $\bar{P}^{(k-1)}$  with the  $k$ 'th column. The rest of the operations are as indicated in eqn.s(53) and (54). The procedure is terminated at the  $M_s$ 'th stage when

$$1 - \sum_{i=1}^{M_s} \frac{[z_i^{(M_s)}]^2}{\langle z, z \rangle} = 1 - \sum_{i=1}^{M_s} \frac{[z_i^{(i)}]^2}{\langle z, z \rangle} < \rho \quad (79)$$

or when AIC(4) is minimized. It can be seen that  $(z_i^{(i)})^2 / \langle z, z \rangle$  corresponds to  $[err]_i$  in the CGS and MGS algorithms and if the notation  $[err]_i = (z_i^{(i)})^2 / \langle z, z \rangle$  is used eqn.(79) becomes identical to eqn.(70). The subset model parameter estimate  $\hat{\Theta}_s$  is computed from

$$R_{M_s} \Theta_s = \begin{bmatrix} z_1^{(M_s)} \\ \cdot \\ \cdot \\ \cdot \\ z_{M_s}^{(M_s)} \end{bmatrix}, \quad \text{that is } R_{M_s} \Theta_s = \begin{bmatrix} z_1^{(1)} \\ \cdot \\ \cdot \\ \cdot \\ z_{M_s}^{(M_s)} \end{bmatrix} \quad (80)$$

using backward substitution.

This algorithm seems to require less computer memory space than the algorithm based on the MGS method because  $R$  is stored by overwriting part of  $P$ . Using the Householder transformation method to select predictors in such a forward-regression manner was mentioned in (Golub, 1965).  $a_j^{(k)}$  and  $b_j^{(k)}$  for  $j=k, \dots, M$  can be calculated quickly in the following way. Given

$$(a_j^{(1)})^2 = \sum_{i=1}^N (\bar{p}_{ij}^{(0)})^2, \quad b_j^{(1)} = \sum_{i=1}^N \bar{p}_{ij}^{(0)} z_i^{(0)} \quad j=1, \dots, M \quad (81)$$

After  $\bar{P}^{(k)}$  has been computed,  $(a_j^{(k+1)})^2$  and  $b_j^{(k+1)}$  for  $j=k+1, \dots, M$  are updated according to

$$(a_j^{(k+1)})^2 = (a_j^{(k)})^2 - (\bar{p}_{kj}^{(k)})^2, \quad b_j^{(k+1)} = b_j^{(k)} - \bar{p}_{kj}^{(k)} z_k^{(k)} \quad j=k+1, \dots, M \quad (82)$$

Naturally, if the columns of  $\bar{P}^{(k-1)}$  are interchanged, the  $a_j^{(k)}$ 's must be interchanged accordingly and so must the  $b_j^{(k)}$ 's.

### 5. Iterative schemes for polynomial NARMAX models

For the polynomial NARMAX model delayed noise terms are included in each subsystem model and these are generally unmeasured. The solution to this problem is to replace  $e(t)$  by the prediction errors or residuals  $\varepsilon(t)$  in the identification process. Let  $P_i$  represent the full submodel set of subsystem  $i$  which is partitioned into

$$P_i = [ P_{p_i}; P_{n_i} ] \quad (83)$$

where each column in  $P_{p_i}$  is a monomial function of the input-output data only and each column in  $P_{n_i}$  is a monomial function of the prediction errors and the input-output data.  $P_{p_i}$  may therefore be referred to as the full  $i$ 'th sub-process model and  $P_{n_i}$  the full  $i$ 'th sub-noise model. A subset  $P_{si}$  of  $P_i$  is similarly represented as

$$P_{si} = [ P_{p_{si}}; P_{n_{si}} ] \quad (84)$$

Referring to Fig.1, a general iterative scheme using the orthogonal algorithms of Section 4 can be described as follows.

Initially the  $i$ 'th estimator selects columns of  $P_{p_{si}}$  from  $P_{p_i}$ . The selection is terminated after  $M_{p_{si}}$  columns have been selected and the condition

$$1 - \sum_{j=1}^{M_{p_{si}}} [err]_j < \rho_{p_i} \quad (85)$$

is satisfied where  $\rho_{p_i}$  is the tolerance for the  $i$ 'th sub-process model. The initial submodel parameter estimate  $\hat{\Theta}_{si}^0$  (containing only  $M_{p_{si}}$  elements) can be computed and then sent to the coordinator. Based on  $\hat{\Theta}_{si}^0$   $i=1, \dots, m$ , the coordinator can generate the initial prediction error sequence  $\{\varepsilon^1(t)\}$ .

At the  $k$ 'th iteration ( $k \geq 1$ ) each estimator receives  $\{\varepsilon^k(t)\}$  from the coordinator. This allows the  $i$ 'th estimator to form  $P_{n_i}^k$  and to select columns from  $P_{n_i}^k$ . Assume that after  $M_{n_i}^k$  more columns have been added to  $P_{p_{si}}$  the condition

$$1 - \sum_{j=1}^{M_{p_{si}} + M_{n_i}^k} [err]_j < \rho_{n_i} \quad (86)$$

is satisfied, where  $\rho_{n_i}$  ( $< \rho_{p_i}$ ) is the tolerance for the  $i$ 'th sub-noise model, the selection procedure is stopped,  $\hat{\Theta}_{si}^k$  (containing  $M_{p_{si}} + M_{n_i}^k$  elements) is computed and sent to the coordinator.

Previous experience involving the CGS orthogonal algorithm has shown that typically 4 to 6 iterations are usually adequate. Since the MGS and Householder transformation algorithms are numerically more accurate than the CGS algorithm, 4 to 6 iterations should also be adequate for the iterative schemes using these two methods.

Notice that the selection of the process and noise model parameters is decoupled. However there is no particular reason why the process model terms should be selected first and the noise model terms selected later other than that this way is convenient for updating  $\{\epsilon(t)\}$ . For SISO systems, an additional stage may be added to the above scheme: After a reasonable  $\{\epsilon(t)\}$  has been obtained, we may reselect process and noise model terms together from  $\mathbf{P}$  and stop the procedure when  $AIC(4)$  is minimized. It is also possible to terminate the iterative scheme using  $AIC(\phi)$  instead of the error-reduction-ratio criterion for SISO systems. For example, the process model regression is stopped when  $AIC(4)$  is minimized and, at each iteration, the noise model regression is stopped when  $AIC(2)$  is minimized. Theoretical analysis (Leontaritis and Billings, 1987) and simulation studies (Leontaritis and Billings, 1987; Chen and Billings, 1988a) indicate that  $AIC(2)$  often overestimates the number of necessary parameters. Ideally we should treat the process and noise terms equally. When  $AIC(4)$  is used to terminate the process model regression however we are in fact trying to fit a NARX model to the system which may be better modelled by a NARMAX model and some unnecessary process terms may be included in the initial stage.  $AIC(2)$  is therefore deliberately used for the noise model regression at each iteration in order to avoid the possibility of losing significant noise model terms. For multivariable systems  $AIC(\phi)$  becomes

$$AIC(\phi) = N \log \det C(\hat{\Theta}_s) + M_s \phi \quad (87)$$

where

$$C(\hat{\Theta}_s) = \frac{1}{N} \sum_{t=1}^N \epsilon(t) \epsilon^T(t) \quad (88)$$

and  $M_s$  is the number of all the subsystem parameters. It is difficult to apply this criterion to terminate the above iterative scheme for multivariable systems because the model structure determination is done in a decentralized way.

Computational aspects of different schemes obtained using different orthogonal methods applied to the NARMAX model are now discussed.

### *CGS scheme*

The CGS algorithm orthogonalizes one column at a time. This makes the iterative CGS scheme a simple and natural extension of the forward-regression orthogonal procedure of Subsection 4.1. After the initial stage,  $\mathbf{W}_{si}$  contains  $M_{p_{si}}$  orthogonal columns and the first  $M_{p_{si}}$  columns of  $\mathbf{A}_{si}$  have been computed. At the  $k$ 'th iteration, the  $i$ 'th estimator simply computes  $M_{p_{si}+1, \dots, M_{p_{si}}+M_{n_{si}}^{(k)}}$  columns of  $\mathbf{A}_{si}$  and selects



single precision.

*Example 1.* The data was collected from a large pilot scale liquid level system where the input was a zero mean Gaussian signal. A description of this process is given in (Billings and Voon, 1986). The inputs and outputs of the system are illustrated in Fig.2.

A polynomial NARX model ( $l=n_y=n_x=3$  and  $n_e=0$ ) was used to fit the data. The full model set consisted of 84 terms. The results obtained by applying the forward-regression variants of CGS, MGS and Householder transformation (HT) algorithms are given in TABLEs 1 to 3 respectively. As expected these three algorithms all produced the same model.

*Example 2.* This is a simulated system. The data was generated by

$$y(t) = 0.5y(t-1) + u(t-2) + 0.1u^2(t-1) + 0.5e(t-1) + 0.2u(t-1)e(t-2) + e(t)$$

where the system noise  $e(t)$  was a Gaussian white sequence with mean zero and variance 0.04 and the system input  $u(t)$  was an independent sequence of uniform distribution with mean zero and variance 1.0. The inputs and outputs of the system are shown in Fig.3.

A polynomial NARMAX model with  $l=n_y=n_x=n_e=2$  was used to fit the data. The full model set contained 28 terms. Using the iterative scheme discussed in Section 5, that is selecting process model terms first and then entering an iterative loop to select the noise model terms, the CGS, MGS and HT algorithms all gave the correct final model, and this can be seen from TABLEs 4 to 6 where  $\rho_p=0.034$ ,  $\rho_n=0.026$  and five iterations (computing the residual sequence 5 times) were involved.

As mentioned in Section 5, other iterative strategies can also be employed. The following procedure was also tested on Example 2. First use AIC(4) to terminate the process model regression and use this to produce the initial residual sequence  $\{e^{(1)}(t)\}$  and noise model set  $P_n^{(1)}$ . Next use AIC(2) to terminate the noise model regression. Having obtained  $\{e^{(k)}(t)\}$   $k \geq 2$  then select the process and noise model terms together from the full model set or the model set obtained from the previous iteration. Each of such regressions are terminated when AIC(4) is minimized. When the model set contains the noise terms, to compute the exact AIC( $\phi$ ) value each time a term is selected would require the computation of all the selected parameter values  $\hat{\theta}_i$  and the recalculation of the residual sequence. This can be avoided by computing the approximate AIC( $\phi$ ) value using the approximate variance of the residuals. For a model of  $M_r$  terms, if the residual sequence is approximated by

$$\hat{E} = z - \sum_{i=1}^{M_r} g_i w_i \quad (92)$$

then the approximate variance of the residuals is readily given by

$$\sigma_z^2 (1 - \sum_{i=1}^{M_r} [err]_i) \quad (93)$$

where

$$\sigma_z^2 = \frac{1}{N} \langle z, z \rangle \quad (94)$$

Notice that for polynomial NARX models eqn.(92) gives the exact residual sequence and eqn.(93) the exact variance of the residuals. Using this procedure and involving only two iterations ( $k=2$ ), the CGS, MGS and HT algorithms produced the models shown in TABLEs 7 to 9 respectively.

## 7. Conclusions

Several orthogonal forward-regression estimators have been derived for the identification of polynomial NARMAX systems by modifying and augmenting some well-known orthogonal least squares methods. It has been shown that these estimators efficiently combine structure determination with parameter estimation to provide very powerful procedures for identifying parsimonious models of structure-unknown systems. The application to both simulated and real data has been demonstrated.

Whilst the iterative CGS scheme is easier to implement on a microcomputer and its coding is simpler experience has shown that the iterative MGS and HT schemes work faster. The first (off-line) version of Givens method discussed in Subsection 3.2 can also be used to develop a similar model structure selection routine but it will require more computations compared with the three routines discussed in this paper.

## Acknowledgments

Two of the authors (SC and SAB) gratefully acknowledge financial support for the work presented above from the Science and Engineering Research Council (Grant Ref. GR/D/30587).

## References

- [1] Bierman, G.J. (1977). *Factorization Methods for Discrete Sequential Estimation*. Academic Press, New York.
- [2] Billings, S.A. (1986). Introduction to nonlinear systems analysis and identification. In: Godfrey, K., and P. Jones (Eds.), *Signal Processing for Control*, pp261-294, Springer Verlag.
- [3] Billings, S.A., S. Chen, and R.J. Backhouse (1988). The identification of linear and nonlinear models of a turbocharged automotive diesel engine. *Mechanical Systems and Signal Processing*, (to appear).
- [4] Billings, S.A., S. Chen, and M.J. Korenberg (1988). Identification of MIMO nonlinear systems using a forward-regression orthogonal estimator. *Int. J. Control*, (to appear).
- [5] Billings, S.A., and M.B. Fadzil (1985). The practical identification of systems with nonlinearities. *Proc. of the 7th IFAC Symp. on Identification and System Parameter Estimation*, York, U.K., pp155-160.
- [6] Billings, S.A., M.B. Fadzil, J. Sulley, and P.M. Johnson (1988). Identification of a nonlinear difference equation model of an industrial diesel generator. *Mechanical Systems and Signal Processing*, Vol.2, pp59-76.
- [7] Billings, S.A., M.J. Korenberg, and S. Chen (1988). Identification of non-linear output-affine systems using an orthogonal least-squares algorithm. *Int. J. Systems Sci.*, Vol.19, No.8, pp1559-1568.
- [8] Billings, S.A., and W.S.F. Voon (1986). A prediction-error and stepwise-regression estimation algorithm for non-linear systems. *Int. J. Control*, Vol.44, No.3, pp803-822.
- [9] Björck, A. (1967). Solving linear least squares problems by Gram-Schmidt orthogonalization. *BIT*, Vol.7, pp1-21.
- [10] Chen, S., and S.A. Billings (1988a). Prediction-error estimation algorithm for non-linear output-affine systems. *Int. J. Control*, Vol.47, No.1, pp309-332.
- [11] Chen, S., and S.A. Billings (1988b). Representations of nonlinear systems: the NARMAX model. *Int. J. Control*, (to appear).
- [12] Desrochers, A., and S. Mohseni (1984). On determining the structure of a non-linear system. *Int. J. Control*, Vol.40, No.5, pp923-938.
- [13] Draper, N.R., and H. Smith (1981). *Applied Regression Analysis*. Wiley, New York.
- [14] Gentleman, W.M. (1973). Least squares computations by Givens transformations without square roots. *J. Inst. Maths Applics*, Vol.12, pp329-336.
- [15] Golub, G. (1965). Numerical methods for solving linear least squares problems. *Numerische Mathematik*, Vol.7, pp206-216.
- [16] Golub, G.H. (1969). Matrix decompositions and statistical calculations. In: Milton, R.C., and J.A. Nelder (Eds.), *Statistical Computation*, pp365-397, Academic Press, New York.
- [17] Golub, G.H., and C. Reinsch (1970). Singular value decomposition and least squares solutions. *Numerische Mathematik*, Vol.14, pp403-420.

- [18] Goodwin, G.C., and R.L. Payne (1977). *Dynamic System Identification: Experiment Design and Data Analysis*. Academic Press, New York.
- [19] Hanson, R.J., and C.L. Lawson (1969). Extensions and applications of the Householder algorithm for solving linear least squares problems. *Math. Comput.*, Vol.23, pp787-812.
- [20] Jordan, T.L. (1968). Experiments on error growth associated with some linear least-squares procedures. *Math. Comput.*, Vol.22, pp579-588.
- [21] Korenberg, M.J. (1985). Orthogonal identification of nonlinear difference equation models. *Mid. West Symp. on Cts and Systems*, Louisville.
- [22] Korenberg, M.J., S.A. Billings, Y.P. Liu and P.J. McIlroy (1988). Orthogonal parameter estimation algorithm for non-linear stochastic systems. *Int. J. Control*, Vol.48, No.1, pp193-210.
- [23] Kortmann, M., and H. Unbehauen (1988). A model structure selection algorithm in the identification of multivariable nonlinear systems with application to a turbogenerator set. Presented at *the 12th IMACS World Congress on Scientific Computation*, July 18-22, 1988, Paris.
- [24] Leontaritis, I.J., and S.A. Billings (1985). Input-output parametric models for non-linear systems, Part I: deterministic non-linear systems; Part II: stochastic non-linear systems. *Int. J. Control*, Vol.41, No.2, pp303-344.
- [25] Leontaritis, I.J., and S.A. Billings (1987). Model selection and validation methods for non-linear systems. *Int. J. Control*, Vol.45, No.1, pp311-341.
- [26] Rice, J.R. (1966). Experiments on Gram-Schmidt orthogonalization. *Math. Comput.*, Vol.20, pp325-328.
- [27] Seber, G.A.F. (1977). *Linear Regression Analysis*. John Wiley & Sons, New York.
- [28] Söderström, T. (1977). On model structure testing in system identification. *Int. J. Control*, Vol.26, No.1, pp1-18.
- [29] Wampler, R.H. (1970). A report on the accuracy of some widely used least squares computer programs. *J. American Statistical Association*, Vol.65, pp549-565.
- [30] Wilkinson, J.H. (1974). The classical error analysis for the solution of linear systems. *Inst. Maths Applics, Bulletin*, Vol.10 pp175-180.

**TABLE 1. Selected model of Example 1 (using CGS and ERR criterion)**

terms	estimates	[err] <sub>i</sub> 's	standard deviations
y(t-1)	0.80916E+0	0.97444E+0	0.47816E-1
u(t-1)	0.41813E+0	0.14158E-1	0.15593E-1
u(t-2)	-0.10762E+0	0.26158E-2	0.16974E-1
y(t-2)y(t-3)u(t-2)	-0.36722E+0	0.12047E-2	0.53508E-1
y(t-1)u(t-1)	-0.33901E+0	0.18040E-2	0.27629E-1
y(t-2)y(t-3)u(t-3)	0.14530E+0	0.31312E-3	0.20979E-1
u(t-1)u <sup>2</sup> (t-2)	-0.16590E+0	0.17932E-3	0.43352E-1
y(t-1)u(t-2)	0.16459E+0	0.73725E-4	0.31417E-1
y <sup>2</sup> (t-2)y(t-3)	-0.39164E-1	0.90227E-4	0.60504E-2
y(t-1)y(t-2)	-0.58358E-1	0.18795E-3	0.10442E-1
y(t-2)	0.16186E+0	0.12685E-3	0.46291E-1
y <sup>2</sup> (t-3)u(t-2)	0.13916E+0	0.64359E-4	0.54245E-1

tolerance  $\rho = 0.0048$ , variance of residuals  $\sigma_e^2 = 0.18585E-2$ ,  
residual variance and output variance ratio =  $0.47429E-2$

**TABLE 2. Selected model of Example 1 (using MGS and ERR criterion)**

terms	estimates	[err] <sub>i</sub> 's	standard deviations
y(t-1)	0.80914E+0	0.97444E+0	0.47816E-1
u(t-1)	0.41814E+0	0.14158E-1	0.15593E-1
u(t-2)	-0.10761E+0	0.26158E-2	0.16974E-1
y(t-2)y(t-3)u(t-2)	-0.36721E+0	0.12047E-2	0.53508E-1
y(t-1)u(t-1)	-0.33901E+0	0.18039E-2	0.27630E-1
y(t-2)y(t-3)u(t-3)	0.14528E+0	0.31304E-3	0.20979E-1
u(t-1)u <sup>2</sup> (t-2)	-0.16590E+0	0.17933E-3	0.43352E-1
y(t-1)u(t-2)	0.16457E+0	0.73714E-4	0.31417E-1
y <sup>2</sup> (t-2)y(t-3)	-0.39161E-1	0.90206E-4	0.60505E-2
y(t-1)y(t-2)	-0.58353E-1	0.18788E-3	0.10443E-1
y(t-2)	0.16188E+0	0.12688E-3	0.46291E-1
y <sup>2</sup> (t-3)u(t-2)	0.13916E+0	0.64358E-4	0.54245E-1

tolerance  $\rho = 0.0048$ , variance of residuals  $\sigma_e^2 = 0.18585E-2$ ,  
residual variance and output variance ratio =  $0.47429E-2$

**TABLE 3. Selected model of Example 1 (using HT and ERR criterion)**

terms	estimates	$[err]_i^2$ 's	standard deviations
$y(t-1)$	0.80914E+0	0.97444E+0	0.47816E-1
$u(t-1)$	0.41814E+0	0.14158E-1	0.15593E-1
$u(t-2)$	-0.10761E+0	0.26158E-2	0.16974E-1
$y(t-2)y(t-3)u(t-2)$	-0.36721E+0	0.12047E-2	0.53508E-1
$y(t-1)u(t-1)$	-0.33901E+0	0.18039E-2	0.27630E-1
$y(t-2)y(t-3)u(t-3)$	0.14528E+0	0.31308E-3	0.20979E-1
$u(t-1)u^2(t-2)$	-0.16590E+0	0.17933E-3	0.43352E-1
$y(t-1)u(t-2)$	0.16457E+0	0.73714E-4	0.31417E-1
$y^2(t-2)y(t-3)$	-0.39161E-1	0.90200E-4	0.60505E-2
$y(t-1)y(t-2)$	-0.58353E-1	0.18792E-3	0.10443E-1
$y(t-2)$	0.16188E+0	0.12687E-3	0.46291E-1
$y^2(t-3)u(t-2)$	0.13916E+0	0.64364E-4	0.54245E-1

tolerance  $\rho = 0.0048$ , variance of residuals  $\sigma_e^2 = 0.18585E-2$   
 residual variance and output variance ratio =  $0.47429E-2$

**TABLE 4. Selected model of Example 2 (using CGS and ERR criterion)**

terms	estimates	$[err]_i^2$ 's	standard deviations
$u(t-2)$	0.10032E+1	0.67047E+0	0.89932E-2
$y(t-1)$	0.50276E+0	0.28735E+0	0.73754E-2
$u^2(t-1)$	0.91139E-1	0.85640E-2	0.68862E-2
$e(t-1)$	0.54833E+0	0.69294E-2	0.45770E-1
$u(t-1)e(t-2)$	0.23770E+0	0.16084E-2	0.42268E-1

variance of residuals  $\sigma_e^2 = 0.41436E-1$   
 residual variance and output variance ratio =  $0.25072E-1$

**TABLE 5. Selected model of Example 2 (using MGS and ERR criterion)**

terms	estimates	$[err]_i^2$ 's	standard deviations
$u(t-2)$	0.10032E+1	0.67047E+0	0.89932E-2
$y(t-1)$	0.50276E+0	0.28735E+0	0.73754E-2
$u^2(t-1)$	0.91139E-1	0.85640E-2	0.68862E-2
$e(t-1)$	0.54833E+0	0.69294E-2	0.45770E-1
$u(t-1)e(t-2)$	0.23770E+0	0.16084E-2	0.42268E-1

variance of residuals  $\sigma_e^2 = 0.41436E-1$   
 residual variance and output variance ratio =  $0.25072E-1$

**TABLE 6. Selected model of Example 2 (using HT and ERR criterion)**

terms	estimates	$[err]_i$ 's	standard deviations
$u(t-2)$	0.10032E+1	0.67047E+0	0.89932E-2
$y(t-1)$	0.50276E+0	0.28735E+0	0.73754E-2
$u^2(t-1)$	0.91139E-1	0.85640E-2	0.68862E-2
$e(t-1)$	0.54834E+0	0.69294E-2	0.45770E-1
$u(t-1)e(t-2)$	0.23770E+0	0.16084E-2	0.42268E-1

variance of residuals  $\sigma_e^2 = 0.41436E-1$   
 residual variance and output variance ratio = 0.25072E-1

**TABLE 7. Selected model of Example 2 (using CGS and AIC criterion)**

terms	estimates	$[err]_i$ 's	standard deviations
$u(t-2)$	0.10034E+1	0.67047E+0	0.90317E-2
$y(t-1)$	0.50355E+0	0.28735E+0	0.74037E-2
$u^2(t-1)$	0.89880E-1	0.85640E-2	0.69221E-2
$e(t-1)$	0.51440E+0	0.65584E-2	0.44959E-1
$u(t-1)e(t-2)$	0.20321E+0	0.12277E-2	0.41518E-1

variance of residuals  $\sigma_e^2 = 0.41754E-1$   
 residual variance and output variance ratio = 0.25265E-1

**TABLE 8. Selected model of Example 2 (using MGS and AIC criterion)**

terms	estimates	$[err]_i$ 's	standard deviations
$u(t-2)$	0.10034E+1	0.67047E+0	0.90317E-2
$y(t-1)$	0.50355E+0	0.28735E+0	0.74037E-2
$u^2(t-1)$	0.89880E-1	0.85640E-2	0.69221E-2
$e(t-1)$	0.51440E+0	0.65584E-2	0.44959E-1
$u(t-1)e(t-2)$	0.20321E+0	0.12277E-2	0.41518E-1

variance of residuals  $\sigma_e^2 = 0.41754E-1$   
 residual variance and output variance ratio = 0.25265E-1

**TABLE 9. Selected model of Example 2 (using HT and AIC criterion)**

terms	estimates	$[err]_i$ 's	standard deviations
$u(t-2)$	0.10034E+1	0.67047E+0	0.90317E-2
$y(t-1)$	0.50355E+0	0.28735E+0	0.74037E-2
$u^2(t-1)$	0.89880E-1	0.85640E-2	0.69221E-2
$e(t-1)$	0.51440E+0	0.65584E-2	0.44959E-1
$u(t-1)e(t-2)$	0.20321E+0	0.12277E-2	0.41518E-1

variance of residuals  $\sigma_e^2 = 0.41754E-1$   
 residual variance and output variance ratio = 0.25265E-1

Fig.1. Iterative scheme for the polynomial NARMAX model

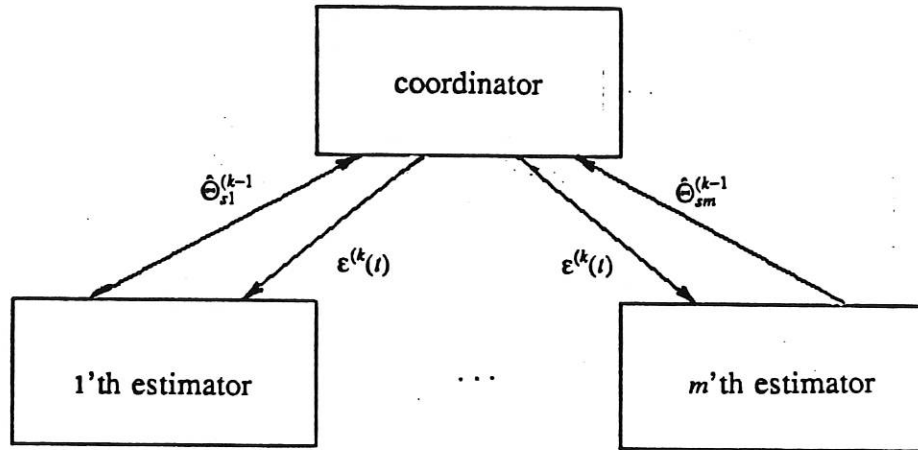


Fig.2. Inputs and outputs of Example 1

