Adaptive Model Selection and Estimation for Nonlinear Systems Using a Sliding Data Window

W. Luo and S. A. Billings

Department of Automatic Control and Systems Engineering

University of Sheffield

Sheffield, S1 4DU

U.K.

March 1994

Research Report No. 510

Adaptive Model Selection and Estimation for Nonlinear Systems Using a Sliding Data Window

W. Luo * *and S. A. Billings* *

Abstract

A new algorithm which provides adaptive model selection and estimation on-line is derived based on the polynomial nonlinear ARMAX model (NARMAX). The algorithm uses rectangular windowing regression procedures where the forgetting factor is unity within a sliding data window. Variations in the model structure and the parameter estimates are tracked by using a sliding rectangular window based on Givens rotations. The algorithm which minimizes the loss function at every step by selecting significant regression variables and computing the corresponding parameter estimates provides an efficient adaptive procedure which can be applied in nonlinear signal processing applications. Simulated examples are included to demonstrate the performance of the new algorithm.

1. Introduction

If the structure of a system model is known a priori on-line estimation is reduced to determining estimates of unknown parameters. In this situation, either the standard recursive least squares algorithm (RLS) or variations of this which exhibit improved numerical properties can be applied. Orthogonal least squares parameter estimation algorithms based on orthogonal-triangular decomposition (*QR* algorithms) are known to be superior to all other RLS algorithms and are therefore often applied in real-time signal processing applications. When the structure of a system model is unknown detecting the system structure becomes a critical part of the identification procedure. Techniques used for the on-line modification of the order of linear models are not easily extended to the nonlinear case because the structure of the nonlinear models do not satisfy the constraints required by these techniques. New algorithms are therefore required for the on-line structure detection of nonlinear systems.

An on-line orthogonal algorithm based on an exponential windowing approach was presented in Luo, Billings and Tsang (1994). The algorithm minimizes the loss

* Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, S1 4DU, U.K.

function at each step by selecting significant regression variables while maintaining the orthogonality of the vector space as each new observation was processed. Because this algorithm uses orthogonal *QR* decomposition with column permutations and the effects of old data were exponentially weighted out, stability was ensured in continuous computation. A better alternative in some applications would be to use a rectangular window with a forgetting factor of 1 but to slide the window over the data set to provide improved adaption. If the forgetting factor λ is unity however this will cause overflow in continuous computations during on-line processing based on the exponential window approach (see Section 3 for further details) and an alternative solution must be sought.

The objective of the present study is therefore to introduce an adaptive orthogonal algorithm with on-line structure detection which is applicable for rectangular windowing regression. Because recursive *QR* decomposition algorithms provide numerically stable and accurate computations with an efficient use of memory these form an ideal basis for development of the new rectangular window method. The new algorithm can adaptively change the model structure on-line but does not require storage of all the regression matrix and associated orthogonal vectors. The new algorithm is different from the exponential windowing method because of the operation of the sliding window, that is the oldest measurement contained in the last observation window is removed before a new measurement is added. Since the oldest measurement is eliminated and the newest measurement is added the new algorithm constructs a sliding window and can produce a series of sub-models based on the data which are updated successively in each window. The new algorithm can perform adaptive on-line model structure detection using a forward selection method and will be referred to as the GFSL algorithm (Givens rotation with Forward selection and SLiding windowing algorithm).

The paper is organized as follows. First the NARMAX model is defined and the recursive orthogonal transformation with a sliding window is derived. The principle of structure detection is described in Section 4 and the procedures and initialization of the algorithm are summarized in Section 5. A discussion in Section 6 considers the computational costs and alternative algorithms. The final two sections contain numerical results and conclusions.

2. NARMAX Model

A single-input single-output NARMAX model (Leontaritis and Billings 1985) can be defined as

$$y(t)=F[y(t-1),....,y(t-n_y),u(t-d),....,u(t-d-n_u+1),\varepsilon(t-1),....,\varepsilon(t-n_\varepsilon)]+\varepsilon(t). \quad (2\text{-}1)$$

where t denotes discrete time, $y(t)$, $u(t)$ and $\varepsilon(t)$ represent the output, input, and prediction errors respectively, n_y, n_u and n_ε are the corresponding orders, $F[.]$ is some non-linear function and d is the minimum time delay of the input. The prediction errors are defined by

$$\varepsilon(t) = y(t) - \hat{y}(t) \quad (2\text{-}2)$$

where $\hat{}$ denotes predicted values, $\hat{y}(t)$ is the one step ahead predicted output. The degree of the power terms in $y(.)$, $u(.)$ and $\varepsilon(.)$ is defined by n_l.

Defining

$$\Delta_1=y(t-1),\ \Delta_2=y(t-2),\ ....,\ \Delta_{n_y}=y(t-n_y)\ ,$$

$$\Delta_{n_y+1}=u(t-d),\ \Delta_{n_y+2}=u(t-1-d),\ ....,\ \Delta_{n_y+n_u}=u(t-n_u-d)\ ,$$

$$\Delta_{n_y+n_u+1}=\varepsilon(t-1),\ \Delta_{n_y+n_u+2}=\varepsilon(t-2),\ ....,\ \Delta_{n_y+n_u+n_\varepsilon}=\varepsilon(t-n_\varepsilon)\ ,$$

$$S = n_y+n_u+n_\varepsilon\ ,$$

then a polynomial NARMAX model with degree n_l can be expressed as

$$y(t)= \kappa_0(t) + \sum_{i_1=1}^{S} \kappa_{i_1}(t)\Delta_{i_1} + \sum_{i_1=1}^{S}\sum_{i_2=1}^{S} \kappa_{i_1 i_2}(t)\Delta_{i_1}\Delta_{i_2} + \cdots$$

$$+ \sum_{i_1=1}^{S}\sum_{i_2=1}^{S} \cdots \sum_{i_{n_l}=1}^{S} \kappa_{i_1 i_2 \cdots i_{n_l}}(t)\Delta_{i_1}\Delta_{i_2} \cdots \Delta_{i_{n_l}}\ , \quad (2\text{-}3)$$

where κ denotes the unknown parameters. Transforming the lagged sequences in Eqn (2-3) into regressors forms a pseudo-linear regression model given by

$$y(t) = \sum_{i=1}^{m}\phi_i(t)\theta_i(t) + \varepsilon(t). \quad (2\text{-}4)$$

where $\phi_i(t)$ expresses the ith regression variable (regressor) which is a monomials of lagged $u(t)$, $y(t)$ and/or $\varepsilon(t)$, m is the number of the regressors and $\theta_i(t)$ is the unknown parameter corresponding to $\phi_i(t)$. The regression equations at the time points 1, 2, ..., t are given by

$$\mathbf{y}(t) = \Phi(t)\theta(t) + \varepsilon(t) \quad (2\text{-}5)$$

where $\Phi(t)$ is the $t \times m$ regression matrix, $\theta(t)$ is the $m \times 1$ parameter matrix, $y(t)$ and $\varepsilon(t)$ are $t \times 1$ matrices. If the functions $\phi_i(.)$ are chosen properly, the polynomial model (2-4) can arbitrarily well approximate any continuous $F[.]$ in (2-1) (Chen and Billings 1989). Various possibilities for parameterizing $F[.]$ exist including the extended model set NARMAX models which may involve functions such as absolute value, exponential, logarithmic, $sgn(.)$ etc. (Billings and Chen 1989). Since the linear-in-the-parameter property is preserved several efficient identification algorithms have been developed (e.g. Liu et al 1987, Chen and Billings 1988, Chen et al 1989) for this class of NARMAX model.

A NARMAX model with an adaptive or changing structure can be defined by extending the definition of m in Eqn (2-4) to be $m(t)$. Such a model will be time-dependent not signal-dependent (Billings and Voon 1987). To match the dynamics of the nonlinear system under test the identification based on the NARMAX model often involves power and cross-product terms of the inputs, outputs and noise. In the initial stage of processing many candidate regressors may emerge but most of these are often insignificant or linearly dependent. On-line structure detection can therefore be applied to determine which regressors should be included in the model at a particular time point. Like most nonlinear models, the structure of the NARMAX model does not satisfy the shift-invariant property which allows successive expansion of candidate variables and therefore techniques which utilize these properties for on-line modification of the model structure cannot be used. This excludes most of the well known algorithms derived for linear models. It becomes necessary therefore to derive new algorithms for adaptive on-line structure detection and parameter estimation for nonlinear systems which can only be described by nonlinear models.

3. Recursive *QR* Decomposition with a Sliding Window

In the GFEX (Givens rotation with Forward selection and EXponential windowing) algorithm (Luo at al 1994) the estimates of system structure and parameters can be considered as the result produced from a new measurement and a prior information. The latter contains previously processed data which is expressed concisely in an augmented matrix. At the time instant t this matrix is given by $[\mathbf{R}(t-1)\ \ \mathbf{v}_m(t-1)]$, where $\mathbf{R}(t-1)$ is an $m \times m$ upper triangular matrix and $\mathbf{v}_m(t-1)$ is an $m \times 1$ column matrix. This matrix and the new measurement at time t contain all the information in an exponential window up to time t. If the data contained in the window were unweighted this window would become rectangular. To obtain accurate solutions in this latter situation

and to avoid overflow in orthogonal decomposition the oldest measurement contained in the matrix $[\mathbf{R}(t-1)\ \mathbf{v}_m(t-1)]$ needs to be removed before the new measurement is added. This leads to the implementation of a sliding window method and forms the basis of the new algorithm presented in this study. At a specific time instant an augmented matrix produced by deleting the oldest data will be referred to as the backward-system and an augmented matrix produced by adding the new measurement will be referred as the forward-system at this time instant. The quantities corresponding to the two systems will be marked with the subscript f and b respectively.

3.1 Forward-Operation

Suppose that there are $t-1$ observations of the system $\mathbf{y}(t-1)=\mathbf{\Phi}(t-1)\boldsymbol{\theta}(t-1)$ which contains m regression variables. Augmenting the $t-1$ regression equations with a zero element denoted $\mathbf{O}_2$ and a $1\times m$ zero row vector $\mathbf{O}_1$ yields

$$\begin{bmatrix}\mathbf{y}(t-1)\\ \mathbf{O}_2\end{bmatrix}=\begin{bmatrix}\mathbf{\Phi}(t-1)\\ \mathbf{O}_1\end{bmatrix}\boldsymbol{\theta}(t-1)$$

Premultiplying the above equations by a $t\times t$ orthonormal matrix $\mathbf{Q}^T(t-1)$ (i.e. $\mathbf{Q}^T(t-1)\mathbf{Q}(t-1)=\mathbf{I}$) yields the QR orthogonal decomposition

$$\mathbf{v}(t-1)=\begin{bmatrix}\mathbf{v}_m(t-1)\\ \mathbf{v}_{t-1-m}(t-1)\\ \mathbf{O}_2\end{bmatrix}=\begin{bmatrix}\mathbf{R}(t-1)\\ \mathbf{O}_{t-1-m}\\ \mathbf{O}_1\end{bmatrix}\hat{\boldsymbol{\theta}}(t-1)+\mathbf{Q}^T(t-1)\boldsymbol{\varepsilon}(t-1) \tag{3-1}$$

where $\mathbf{v}_m(t-1)$ and $\mathbf{v}_{t-1-m}(t-1)$ have dimensions $m\times 1$ and $(t-1-m)\times 1$ respectively, $\mathbf{O}_{t-1-m}$ is a $(t-1-m)\times m$ zero matrix, $\boldsymbol{\varepsilon}(t-1)$ is the residual vector.

When the new measurement data at time t, $\boldsymbol{\phi}(t)$ and $y(t)$ (where $\boldsymbol{\phi}(t)$ is an $1\times m$ regression matrix and $y(t)$ is the output at time t) are added to improve the previous estimates, the regression equations can be written as

$$\begin{bmatrix}\mathbf{v}(t-1)\\ \mathbf{v}_{t-1-m}(t-1)\\ y(t)\end{bmatrix}=\begin{bmatrix}\mathbf{R}(t-1)\\ \mathbf{O}_{t-1-m}\\ \boldsymbol{\phi}(t)\end{bmatrix}\hat{\boldsymbol{\theta}}(t-1)+\mathbf{Q}^T(t-1)\boldsymbol{\varepsilon}(t-1)+\begin{bmatrix}\mathbf{O}_{\varepsilon 1}\\ \mathbf{O}_{\varepsilon 2}\\ \underline{\varepsilon}(t)\end{bmatrix}, \tag{3-2}$$

where $\mathbf{O}_{\varepsilon 1}$ and $\mathbf{O}_{\varepsilon 2}$ are $m\times 1$ and $(t-1-m)\times 1$ zero matrices respectively, and $\underline{\varepsilon}(t)$ is the a priori prediction error defined by $\underline{\varepsilon}(t)=y(t)-\boldsymbol{\phi}(t)\hat{\boldsymbol{\theta}}(t-1)$. Choose a $t\times t$ orthonormal matrix $\mathbf{Q}_t(t)$ to update the QR decomposition so that

$$\mathbf{v}(t)=\begin{bmatrix}\mathbf{v}_m(t)\\ \mathbf{v}_{t-1-m}(t)\\ v_t(t)\end{bmatrix}=\begin{bmatrix}\mathbf{R}(t)\\ \mathbf{O}_{t-1-m}\\ \mathbf{O}_1\end{bmatrix}\hat{\boldsymbol{\theta}}(t)+\mathbf{Q}^T(t)\boldsymbol{\varepsilon}(t)\ , \tag{3-3}$$

where $v_t(t)$ is the t' th element of $\mathbf{v}(t)$, the $m \times m$ upper triangular matrix is defined as

$$\mathbf{R}(t) = \begin{bmatrix} r_{11}(t) & r_{12}(t) & \cdots & r_{1m}(t) \\ 0 & r_{22}(t) & \cdots & r_{2m}(t) \\ \cdot & 0 & \cdots & \cdot \\ \cdot & & \cdots & \cdot \\ \cdot & & \cdots & \cdot \\ 0 & \cdots & 0 & r_{mm}(t) \end{bmatrix} , \tag{3-4}$$

the orthonormal matrix $\mathbf{Q}(t)=\mathbf{Q}(t-1)\mathbf{Q}_t(t)$ is given by

$$\mathbf{Q}(t) = [\mathbf{q}_1(t) \;\; \mathbf{q}_2(t) \;\; \cdots \;\; \mathbf{q}_m(t) \;\; \mathbf{q}_{m+1}(t) \;\; \cdots \;\; \mathbf{q}_t(t)] , \tag{3-5}$$

and the residual vector becomes

$$\boldsymbol{\varepsilon}(t) = \begin{bmatrix} \boldsymbol{\varepsilon}_m(t) \\ \boldsymbol{\varepsilon}_{t-1-m}(t) \\ \boldsymbol{\varepsilon}(t) \end{bmatrix} ,$$

where the t' th element of $\boldsymbol{\varepsilon}(t)$, $\varepsilon(t)$, is the residual (the a posteriori prediction error) defined as $\varepsilon(t) = y(t) - \boldsymbol{\phi}(t)\hat{\boldsymbol{\theta}}(t)$. The estimation of $\hat{\boldsymbol{\theta}}(t)$ is easily achieved by solving the triangular system $\mathbf{R}(t)\hat{\boldsymbol{\theta}}(t) = \mathbf{v}_m(t)$. This shows that removing the equations between the $(t-1-m)$' th and $(t-1)$' th row of (3-3) does not affect the solution. The residual sum of squares (RSS) can easily be obtained from

$$RSS(t) = \|\boldsymbol{\varepsilon}(t)\|^2 = \|\mathbf{y}(t)\|^2 - \|\mathbf{v}_m(t)\|^2 = \mathbf{y}^T(t)\mathbf{y}(t) - \mathbf{v}_m^T(t)\mathbf{v}_m(t) \tag{3-6}$$

and can therefore be used for on-line structure detection which will be described later. If the orthogonal decomposition is performed row by row in the transformation from Eqn (3-2) to (3-3) the equations between the $(t-1-m)$' th and $(t-1)$' th row of (3-2) do not need to change when the new upper triangular matrix $\mathbf{R}(t)$ and the new column vector $\mathbf{v}_m(t)$ are obtained. The operation which eliminates $\boldsymbol{\phi}(t)$ to a zero row and correspondingly transforms $y(t)$ to $v_t(t)$ can be executed in an augmented matrix form and is expressed as follows

$$\begin{bmatrix} \mathbf{R}(t-1) & \mathbf{v}_m(t-1) \\ \boldsymbol{\phi}(t) & y(t) \end{bmatrix} \quad \Rightarrow \quad \begin{bmatrix} \mathbf{R}(t) & \mathbf{v}_m(t) \\ \mathbf{O}_1 & v_t(t) \end{bmatrix}$$

Because the orthogonal transformation is performed row by row the orthogonal matrix $\mathbf{Q}_t(t)$ does not need to be stored and therefore neither does $\mathbf{Q}(t)$ (where $\mathbf{Q}(t)=\prod_{i=1}^{t}\mathbf{Q}_i(i)$), so that the recursive orthogonal decomposition can still be performed. When the augmented matrix system is used to express the above orthogonal transformation neither $\mathbf{Q}_t(t)$ nor $\mathbf{Q}(t)$ needs to be expressed explicitly, but the changes in $\mathbf{R}$ and $\mathbf{v}_m$ can clearly be observed and the continuous computation is easily executed. If new measurement

data need to be added to improve the estimates this augmented matrix system is easily used by putting each data set at the $(m+1)$'th row of the augmented matrix and then transforming this row of data. After the new data are processed the augmented matrix is updated successively and a recursive transformation is performed. At the beginning of every step $v_t(t)$ can be ignored and can be replaced by the 1×1 zero matrix $\mathbf{O}_2$. Consequently the result of Eqn (3-3) can be rewritten in the form of an augmented matrix for continuous computation, which is given in

$$\begin{bmatrix} \mathbf{R}(t) & \mathbf{v}_m(t) \\ \mathbf{O}_1 & \mathbf{O}_2 \end{bmatrix} \tag{3-7}$$

To illustrate this consider the following matrix

$$\underline{\mathbf{C}}_f = \begin{bmatrix} \mathbf{R}_{w_b}(t-1) & \mathbf{v}_{w_b,m}(t-1) \\ \phi(t) & y(t) \end{bmatrix} \tag{3-8}$$

where $\underline{\mathbf{C}}_f$ is a $(m+1)\times(m+1)$ matrix which includes the new measurements $\phi(t)$ and $y(t)$]), $R_{w_b}(t-1)$ and $\mathbf{v}_{w_b,m}(t-1)$ contain information in the past w_l-1 data up to time $t-1$ and w_l is defined as the size of the observation window (namely, the length of data to be processed at every computational step). Using the Givens rotation (Gentleman 1973, Gentleman 1974) the triangularization of (3-8) can be implemented using the following procedure:

For $k=1,2,....m$,

$$\underline{c}^{*}_{f,kk} = \sqrt{\underline{c}^{2}_{f,kk} + (\underline{c}^{(k-1)}_{f,m+1,k})^2} \tag{3-9a}$$

$$f_c = \frac{\underline{c}_{f,kk}}{\underline{c}^{*}_{f,kk}} \qquad f_s = \frac{\underline{c}^{(k-1)}_{f,m+1,k}}{\underline{c}^{*}_{f,kk}} \tag{3-9b}$$

For $p=k,k+1,...,m+1$,

$$\underline{c}^{*}_{f,kp} = \underline{c}_{f,kp}\, f_c + \underline{c}^{(k-1)}_{f,m+1,p}\, f_s \tag{3-9c}$$

$$\underline{c}^{(k)}_{f,m+1,p} = -\underline{c}_{f,kp}\, f_s + \underline{c}^{(k-1)}_{f,m+1,p}\, f_c \tag{3-9d}$$

where $\underline{c}^{(0)}_{f,m+1}=[\phi(t)\ \ y(t)]$, is the last row of $\underline{\mathbf{C}}_f$. This orthogonal QR transformation is performed row by row. For example the transformation of the i' th and j' th row is presented as follows

$$\textit{row } i\text{: } 0, ..., 0, \underline{c}_{f,i,k}, \underline{c}_{f,i,k+1}, ..., \underline{c}_{f,i,m+1} \quad => \quad 0, ..., 0, \underline{c}^{*}_{f,i,k}, \underline{c}^{*}_{f,i,k+1}, ..., \underline{c}^{*}_{f,i,m+1}$$

$$\textit{row } j\text{: } 0, ..., 0, \underline{c}_{f,j,k}, \underline{c}_{f,j,k+1}, ..., \underline{c}_{f,j,m+1} \quad => \quad 0, ..., 0, 0\ , \underline{c}^{*}_{f,j,k+1}, ..., \underline{c}^{*}_{f,j,m+1}$$

where * denotes the new quantities after transformation. Finally the matrix

$$\underline{C}_f^* = \begin{bmatrix} \mathbf{R}_{w_f}(t) & \mathbf{v}_{w_f,m}(t) \\ \mathbf{O}_1 & \mathbf{O}_2 \end{bmatrix} \tag{3-10}$$

is obtained where $\mathbf{R}_{w_f}(t)$ and $\mathbf{v}_{w_f,m}(t)$ contain the information in the w_l data window up to time t. The right-hand side of (3-10) represents the forward-system at time t.

The transformation operation of the GFEX algorithm (Luo et al 1994) is a forward operation similar to the transformation of (3-8). The above method only differs by using $\mathbf{R}_{w_b}(t-1)$ and $\mathbf{v}_{w_b,m}(t-1)$ instead of the two corresponding quantities in an exponential window (e.g. $\bar{\mathbf{R}}(t-1)$ and $\bar{\mathbf{v}}_m(t-1)$) and using a unity weighting factor in an observation window. As mentioned above if the forgetting factor λ is unity (a rectangular window) a forward operation will cause overflow in continuous computation (the size of a rectangular window becomes infinite). This follows from the formula (3-9a). For example in every computational step the first diagonal element of the new triangular matrix $\mathbf{R}(t)$ is always computed by squaring the first element of the new measurement $[\phi(t)\ \ y(t)]$ and then adding this to the square of the first diagonal element of the previously formed $\mathbf{R}(t-1)$. When $\lambda=1$ all quantities in the triangular matrix do not decay at every computational step so that the first diagonal element r_{11}, will tend to become infinite. By using a finite rectangular window this problem can be overcome by removing the oldest data from the matrix $[\mathbf{R}_{w_f}(t-1)\ \ \mathbf{v}_{w_f,m}(t-1)]$ which was formed in the forward-operation at time $t-1$ using w_l data up to time $t-1$. Since the complete regression matrix cannot be stored but only w_l records of the latest inputs, outputs and residuals some methods to backdate the window must be developed.

3.2 Backward-Operation

To delete data from the augmented matrix, Chambers (1971) used the reverse operation to add a row of data and Golub (1969) gave a method in which the row of data to be deleted was multiplied by a factor $\sqrt{-1}$ and then orthogonally transformed to form a backdated matrix.

Inspection of (3-9) shows that the procedure for deleting a row of data can be considered as the reverse implementation of adding a row of data. Suppose $[\phi(t)\ \ y(t)]$ is required to be eliminated from the resulting matrix $\underline{C}_f^*$, the problem is to find $\underline{c}_{f,kp}$ from $\underline{c}^*_{f,kp}$ and $\underline{c}^{(k-1)}_{f,m+1,p}$, $p=k,\ldots,m+1$. First $[\phi(t)\ \ y(t)]$ is put into the $m+1$ th row of $\underline{C}_f^*$. Rearranging (3-9) forms the following operations.

For $k=1,...,m$,

$$\underline{c}_{f,kk} = \sqrt{(\underline{c}^{*}_{f,kk})^2 - (\underline{c}^{(k-1)}_{f,m+1,k})^2} \tag{3-11a}$$

$$f_c = \frac{\underline{c}_{f,kk}}{\underline{c}^{*}_{f,kk}} \qquad f_s = \frac{\underline{c}^{(k-1)}_{f,m+1,k}}{\underline{c}^{*}_{f,kk}} \tag{3-11b}$$

For $p=k,k+1,...,m+1$,

$$\underline{c}_{f,kp} = (\underline{c}^{*}_{f,kp} - \underline{c}^{(k-1)}_{f,m+1,p} f_s) / f_c \tag{3-11c}$$

$$\underline{c}^{(k)}_{f,m+1,p} = -\underline{c}_{f,kp} f_s + \underline{c}^{(k-1)}_{f,m+1,p} f_c \tag{3-11d}$$

In sliding window algorithms, the oldest measurement $[\phi(t-w_l+1)\ y(t-w_l+1)]$ which needs to be deleted is put in the $m+1$ th row of $\underline{C}^{*}_{f}$ and then using (3-11) the matrix

$$\underline{C}^{*}_{b} = \begin{bmatrix} \mathbf{R}_{w_b}(t) & \mathbf{v}_{w_b,m}(t) \\ \mathbf{O}_1 & \mathbf{O}_2 \end{bmatrix} \tag{3-12}$$

can be obtained, where $\mathbf{R}_{w_b}(t)$ and $\mathbf{v}_{w_b,m}(t)$ correspond to w_l-1 data up to time t (these matrices or variables with the subscript b are associated with deleting the oldest measurement). The right-hand side of (3-12) is the backward-system at t. This method is simple but the operations of adding and deleting are not uniform and therefore it is not easy to implement. Therefore another deleting procedure will be derived as follows.

After deleting $[\phi(t-w_l+1)\ y(t-w_l+1)]$, the sum of squares of the outputs can be expressed as

$$\begin{aligned} \mathbf{y}^T_{w_b}(t)\mathbf{y}_{w_b}(t) &= \mathbf{y}^T_{w_f}(t)\mathbf{y}_{w_f}(t) - y^2(t-w_l+1) \\ &= (\Phi_{w_f}(t)\theta(t))^T \Phi_{w_f}(t)\theta(t) - (\phi(t-w_l+1)\theta(t))^T \phi(t-w_l+1)\theta(t) \end{aligned} \tag{3-13}$$

Writing the above in matrix form yields

$$[\mathbf{y}^T_{w_f}(t)\ \ \alpha\, y(t-w_l+1)] \begin{bmatrix} \mathbf{y}_{w_f}(t) \\ \alpha\, y(t-w_l+1) \end{bmatrix} = \theta^T(t) [\Phi^T_{w_f}(t)\ \alpha\, \phi^T(t-w_l+1)] \begin{bmatrix} \Phi_{w_f}(t) \\ \alpha\, \phi(t-w_l+1) \end{bmatrix} \theta(t)$$

Where α ($\alpha^2=1$) denotes an operator which is not used in the computational subroutines (3-16) but as a marker. Further

$$\begin{bmatrix} \mathbf{y}_{w_f}(t) \\ \alpha\, y(t-w_l+1) \end{bmatrix} = \begin{bmatrix} \Phi_{w_f}(t) \\ \alpha\, \phi(t-w_l+1) \end{bmatrix} \theta(t)$$

Orthogonally transforming $\mathbf{y}_{w_f}(t)$ and $\Phi_{w_f}(t)$ yields

$$\begin{bmatrix} \mathbf{v}_{w_f,m}(t) \\ \mathbf{v}_{w_f,w_l-m}(t) \\ \alpha y(t-w_l+1) \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{w_f}(t) \\ \mathbf{O}_{w_l-m} \\ \alpha\, \phi(t-w_l+1) \end{bmatrix} \hat{\theta}(t) + \mathbf{Q}^T(t)\varepsilon_{w_f}(t) + \begin{bmatrix} \mathbf{O}_{\varepsilon_3} \\ \mathbf{O}_{\varepsilon_4} \\ \alpha \underline{\varepsilon}_{w_f,b} \end{bmatrix}, \tag{3-14}$$

where $\mathbf{Q}^T(t)$ is the orthogonal transformation matrix, $\mathbf{O}_{w_l-m}$ is a $(w_l-m)\times m$ zero matrix, $\mathbf{v}_{w_f,m}(t)$ and $\mathbf{v}_{w_f,w_l-m}(t)$ are of dimension $m\times 1$ and $(w_l-m)\times 1$, $\boldsymbol{\varepsilon}_{w_f}(t)$ is the residual vector, $\mathbf{O}_{\varepsilon_3}$ and $\mathbf{O}_{\varepsilon_4}$ are the zero vectors with dimension $m\times 1$ and $(w_l-m)\times 1$, and $\underline{\varepsilon}_{w_f,b}(t)$ is a prediction error defined as

$$\underline{\varepsilon}_{w_f,b}(t) = y(t-w_l+1) - \boldsymbol{\phi}(t-w_l+1)\hat{\boldsymbol{\theta}}(t) \ .$$

The deleting procedure can now be performed on a concise form of (3-14)

$$\begin{bmatrix} \mathbf{R}_{w_f}(t) & \mathbf{v}_{w_f,m}(t) \\ \alpha\,\boldsymbol{\phi}(t-w_l+1) & \alpha\, y(t-w_l+1) \end{bmatrix} \tag{3-15}$$

by defining

$$\underline{\mathbf{C}}_{b_n} = \begin{bmatrix} \mathbf{R}_{w_f}(t) & \mathbf{v}_{w_f,m}(t) \\ \boldsymbol{\phi}(t-w_l+1) & y(t-w_l+1) \end{bmatrix} \ .$$

and

$$\underline{\mathbf{C}}_b = (Eqn\ \ 3\text{–}15) = \begin{bmatrix} \mathbf{I} & \\ & \alpha \end{bmatrix} \underline{\mathbf{C}}_{b_n} \ .$$

Thus the arithmetic for retriangularizing $\underline{\mathbf{C}}_b$ is to choose b_c and b_s in (3-16) so that the elements of the $m+1$ th row become zero. Such a computation which transforms $\underline{\mathbf{C}}_b$ into a backward-system $\underline{\mathbf{C}}_b^*$ defined in (3-12) can be performed as follows.

For $k=1,2,....,m$,

$$\underline{c}^*_{b_n,kk} = \sqrt{\underline{c}^2_{b_n,kk} - (\underline{c}^{(k-1)}_{b_n,m+1,k})^2} \tag{3-16a}$$

$$(\ because\ \ \underline{c}^*_{b_n,kk} = \sqrt{\underline{c}^2_{b_n,kk} + (\alpha\ \underline{c}^{(k-1)}_{b_n,m+1,k})^2}\ )$$

$$b_c = \frac{\underline{c}_{b_n,kk}}{\underline{c}^*_{b_n,kk}} \qquad \left(\ b_{s_l} = \frac{\alpha\ \underline{c}^{(k-1)}_{b_n,m+1,k}}{\underline{c}^*_{b_n,kk}} = \alpha\ b_s\ \right) \tag{3-16b}$$

where

$$b_s = \frac{\underline{c}^{(k-1)}_{b_n,m+1,k}}{\underline{c}^*_{b_n,kk}} \ . \tag{3-16c}$$

For $p=k,k+1,....,m+1$,

$$\underline{c}^*_{b_n,kp} = \underline{c}_{b_n,kp}\ b_c - \underline{c}^{(k-1)}_{b_n,m+1,p}\ b_s \tag{3-16d}$$

$$(\ because\ \ \underline{c}^*_{b_n,kp} = \underline{c}_{b_n,kp}\ b_c + \alpha\ \underline{c}^{(k-1)}_{b_n,m+1,p}\ b_{s_l}\ )\ ,$$

$$\underline{c}_{b_n,m+1,p}^{(k)} = \underline{c}_{b_n,kp}\, b_s - \underline{c}_{b_n,m+1,p}^{(k-1)}\, b_c \tag{3-16e}$$

$$\left(\underline{c}_{b_n,m+1,p}^{(k)} = -\underline{c}_{b_n,kp}\, b_{s_t} + \alpha\, \underline{c}_{b_n,m+1,p}^{(k-1)} b_c \ , \textit{but store } \underline{c}_{b_n,kp}\, b_s - \underline{c}_{b_n,m+1,p}^{(k-1)} b_c \right).$$

Finally $\underline{\mathbf{C}}_b^* = \underline{\mathbf{C}}_{b_n}^*$.

3.3 A Uniform Formula for the Two Operations

Combining (3-9) with (3-16) provides a uniform formula for adding and deleting a row of data. Define the matrix

$$\underline{\mathbf{C}} = \begin{cases} \underline{\mathbf{C}}_f & \text{for } \textit{adding a row} \\ \underline{\mathbf{C}}_{b_n} & \text{for } \textit{deleting a row} \end{cases} \tag{3-17a}$$

and the arithmetic coefficient

$$w_e = \begin{cases} 1 & \text{for } \textit{adding a row} \\ -1 & \text{for } \textit{deleting a row} \end{cases} \tag{3-17b}$$

Then for $k=1,\ldots,m$,

$$\underline{c}_{kk}^* = \sqrt{\underline{c}_{kk}^2 + (\underline{c}_{m+1,k}^{(k-1)})^2 w_e} \tag{3-17c}$$

$$d_{cc} = \frac{\underline{c}_{kk}}{\underline{c}_{kk}^*} \qquad d_{ss} = \frac{\underline{c}_{m+1,k}^{(k-1)}}{\underline{c}_{kk}^*} \tag{3-17d}$$

and for $p=k,\ldots,m+1$,

$$\underline{c}_{kp}^* = \underline{c}_{kp}\, d_{cc} + \underline{c}_{m+1,p}^{(k-1)}\, d_{ss}\, w_e \tag{3-17e}$$

$$\underline{c}_{m+1,p}^{(k)} = -\underline{c}_{kp}\, d_{ss}\, w_e + \underline{c}_{m+1,p}^{(k-1)}\, d_{cc}\, w_e \tag{3-17f}$$

Since the adding and deleting operations using (3-17) are simple and uniform, this method will be selected as the computational procedure for the new GFSL algorithm.

4. On-line Structure Detection

At time t, the GFSL algorithm deletes the oldest measurement data using (3-17) to produce a backward-system and then adds the newest measurement by means of the same formula to produce a forward-system, i.e.

$$\begin{bmatrix} \mathbf{R}_{w_f}(t) & \mathbf{v}_{w_f,m}(t) \\ \mathbf{O}_1 & \mathbf{O}_2 \end{bmatrix} \tag{4-1}$$

which includes the information over the w_l data up to time t in the observation window. Since the system structure at t is unknown, the computation has to involve all candidate variables and some of these variables may be linearly dependent so that the regression matrix $\Phi_{w_f}(t)$ may not be full rank. This means that $\Phi_{w_f}^T(t)\Phi_{w_f}(t)$ may be singular and there may no longer be a unique solution. In this situation $\Phi_{w_f}(t)$ can still be decomposed, but $\mathbf{R}_{w_f}(t)$ is a real upper triangular matrix with zero diagonal elements and m_r positive diagonal elements and the row vectors of $\mathbf{R}_{w_f}(t)$ corresponding to the zero diagonal elements become zero vectors. The method for on-line structure detection (Luo et al 1994) can be readily applied to this forward-system to determine the sub-model available to time t.

To match the condition of the finite window (the size of the window is w_l), rewriting (3-6) yields

$$RSS_{w_f,m_s(t)}(t) = \mathbf{y}_{w_f}^T(t)\,\mathbf{y}_{w_f}(t) - \mathbf{v}_{w_f,m_s(t)}^T(t)\,\mathbf{v}_{w_f,m_s(t)}(t) \tag{4-2}$$

where the subscript w_f denotes the quantities relative to a forward-system which includes the information of w_l data up to time t, $m_s(t)$ is the number of the selected regressors in the sub-model at time t. $m_s(t)$ is less than the number of the candidate variables, m. The quantities with the subscript $m_s(t)$ indicates that they are associated with the value of $m_s(t)$. Dividing (4-2) by $\mathbf{y}_{w_f}^T\mathbf{y}_{w_f}$ gives the normalized *RSS* (*NRSS*), given by

$$NRSS_{w_f,m_s(t)}(t) = 1 - \frac{\mathbf{v}_{w_f,m_s(t)}^T(t)\mathbf{v}_{w_f,m_s(t)}(t)}{\mathbf{y}_{w_f}(t)^T\mathbf{y}_{w_f}(t)} = 1 - \sum_{i=1}^{m_s(t)}(t)ERR_{w_f,i}(t)\ , \tag{4-3}$$

where $ERR_{w_f,i}(t) = v_{w_f,i}^2(t) \,/\, \mathbf{y}_{w_f}(t)^T\mathbf{y}_{w_f}(t)$. This is the error reduction ratio (*ERR*) of the orthogonal vector $\mathbf{q}_{w_f,i}$ which is obtained by transforming the first i columns of $\Phi_{w_f}(t)$ and $v_{w_f,i}$ represents the projection of the output in the direction of the ith orthogonal vector, $\mathbf{q}_{w_f,i}$. The value of $v_{w_f,i}^2$ can easily be used to select significant regressors from all the candidate regression variables by using a forward search procedure so that the selected variable minimizes $NRSS_{w_f,j}(t)$ at the j'th selection step. The selection will be continued for $m_s(t)$ steps until

$$NRSS_{w_f,m_s(t)}(t) \le \xi_s\ , \tag{4-4}$$

where ξ_s is a pre-set tolerance. If $m_s(t)$ orthogonal vectors are selected, this means that the first $m_s(t)$ columns of $\Phi_{w_f}(t)$ are determined as significant regressors.

To ensure clarity the notation "(t)" and subscript w_f will be ignored in the remainder of this section.

At the j'th selection step all candidate variables ϕ_p, $p=j,....,m$, are considered initially. Choose the candidate variable with the maximum of $v_{j(p)}^2$, $p=j,....,m$, as the j'th optimal regressor so that the RSS_j can be minimized. Such a computation for $v_{j(p)}^2$ can only use the columns of $\mathbf{R}$, $\mathbf{r}_p$, $p=j,....,m$, which have the following form

$$\mathbf{r}_p = [r_{1p}\ r_{2p}\ \cdots\ r_{jp}\ \cdots\ r_{pp}\ 0\ \cdots\ 0]^T\ .$$

and v_p, $p=j,....,m$. First set $v_{j(j)}=v_j$ and two auxiliary variables $r_{j(p)}^{(j)}=r_{jp}$ and $v_{j(p)}^{(j)}=v_j$. Calculate $v_{j(p)}$ using the following procedure:

For $i=j+1,....,p$,

$$r_{j(p)}^{(i)} = \sqrt{(r_{j(p)}^{(i-1)})^2 + r_{ip}^2} \tag{4-5a}$$

$$v_{j(p)}^{(i)} = v_{j(p)}^{(i-1)} \frac{r_{j(p)}^{(i-1)}}{r_{j(p)}^{(i)}} + v_i \frac{r_{ip}}{r_{j(p)}^{(i)}}\ . \tag{4-5b}$$

Finally $v_{j(p)}^2 = (v_{j(p)}^{(p)})^2$. For example, when $v_{j(k)}^2$ is the the maximum of all the $v_{j(p)}^2$, ϕ_k is selected as the j'th optimal regressor. Then exchange the positions of $\mathbf{r}_k^{(j-1)}$ and $\mathbf{r}_j^{(j-1)}$ in $\mathbf{R}^{(j-1)}$ to form $\mathbf{R}^{(j-1)*}$, where the matrices with the superscript $(j-1)$ indicate the forms of the $(j-1)$'th selection step. Since the permuted $\mathbf{R}^{(j-1)*}$ is not triangular, a reorthogonalization procedure must be applied to retriangularize the augmented matrix

$$\begin{bmatrix} \mathbf{R}^{(j-1)*} & \mathbf{v}_m^{(j-1)} \\ \mathbf{O}_1 & \mathbf{O}_2 \end{bmatrix} \tag{4-6}$$

where $\mathbf{v}_m^{(j-1)}$ is the orthogonalized output vector produced in the $(j-1)$'th selection step. The new augmented matrix

$$\begin{bmatrix} \mathbf{R}^{(j)} & \mathbf{v}_m^{(j)} \\ \mathbf{O}_1 & \mathbf{O}_2 \end{bmatrix} \tag{4-7}$$

can easily be formed by applying a Givens transformation (3-17) with $w_\epsilon=1$ to from the j'th row to m'th row of (4-6). The selection process is continued until (4-4) is satisfied.

At every selection step the procedures include the selection of the optimal regressors, the permutation of the columns of $\mathbf{R}$ and the retriangularization of the augmented matrix. Although the operation is only performed in $\mathbf{R}$ and $\mathbf{v}_m$, the effect is to select the optimal regressors in the regression matrix $\mathbf{\Phi}$.

5. Computational Procedures and Initialization

To completely describe the implementation of the new GFSL algorithm, a summary of the procedures will be given in this section. The initialization of the algorithm will also be discussed.

Suppose at time t, the forward-system at time $t-1$ is given by

$$\begin{bmatrix} \mathbf{R}_{w_f}(t-1) & \mathbf{v}_{w_f,m}(t-1) \\ \mathbf{O}_1 & \mathbf{O}_2 \end{bmatrix}, \tag{5-1}$$

the operations of the GFSL algorithm during the computational interval at time t can be described as

GFSL Algorithm

(i) If $t \le w_l$, go to Step (ii); otherwise perform the back-operation. To delete the data at time $t-w_l$ from (5-1), first put $[\phi(t-w_l)\ \ y(t-w_l)]$ in the $m+1$ th row of (5-1) to form

$$\begin{bmatrix} \mathbf{R}_{w_f}(t-1) & \mathbf{v}_{w_f,m}(t-1) \\ \phi(t-w_l) & y(t-w_l) \end{bmatrix} \tag{5-2}$$

Then with $w_e=-1$, apply the orthogonal transformation procedure (3-17) to achieve the backward-system

$$\begin{bmatrix} \mathbf{R}_{w_b}(t-1) & \mathbf{v}_{w_b,m}(t-1) \\ \mathbf{O}_1 & \mathbf{O}_2 \end{bmatrix}. \tag{5-3}$$

(ii) Add the data at time t in (5-3) to form

$$\begin{bmatrix} \mathbf{R}_{w_b}(t-1) & \mathbf{v}_{w_b,m}(t-1) \\ \phi(t) & y(t) \end{bmatrix}. \tag{5-4}$$

Setting $w_e=1$, using (3-17) retriangularizes (5-4) to form the forward-system

$$\begin{bmatrix} \mathbf{R}^{(0)}_{w_f}(t) & \mathbf{v}^{(0)}_{w_f,m}(t) \\ \mathbf{O}_1 & \mathbf{O}_2 \end{bmatrix}. \tag{5-5}$$

(iii) Choose the j'th optimal regressor ($j \ge 1$) with the maximum $v^2_{w_f,j(p)}(t)$, $p=j,\ldots,m$, using (4-5).

(iv) Based on the result of Step (iii), perform the permutation of the columns of $\mathbf{R}^{(j-1)}_{w_f}(t)$ and then the corresponding retriangularization.

(v) Compute $NRSS_{w,j}(t)$ which is defined as

$$NRSS_{w_f,j}(t) = 1 - \sum_{i=1}^{j} ERR_{w_f,i}(t) \tag{5-6}$$

where

$$ERR_{w_f,i}(t) = \frac{v^2_{w_f,i}(t)}{\mathbf{y}^T_{w_f}(t)\,\mathbf{y}_{w_f}(t)} \quad . \tag{5-7}$$

If $NRSS_{w_f,j}(t)$ satisfies the pre-set tolerance, namely, $NRSS_{w_f,j}(t) \le \xi_s$, the selection is terminated; otherwise utilize some statistical tests to check the selection and then decide whether to return to Step (iii) to involve more regressors. Suppose that $m_s(t)$ regressors have been selected, the new forward-system is rearranged based on the contribution of the regressors to the output as

$$\begin{bmatrix} \mathbf{R}_{w_f}(t) & \mathbf{v}_{w_f,m}(t) \\ \mathbf{O}_1 & \mathbf{O}_2 \end{bmatrix} = \begin{bmatrix} r_{w_f,11}(t) & \cdots & r_{w_f,1\,m_s(t)}(t) & \cdots & v_{w_f,1}(t) \\ 0 & \ddots & \vdots & & \vdots \\ \vdots & \ddots & \vdots & & \vdots \\ \vdots & & r_{w_f,m_s(t)\,m_s(t)}(t) & & v_{w_f,m_s(t)}(t) \\ \vdots & & 0 & \ddots & \vdots \\ \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & & 0 \end{bmatrix},$$

where $\mathbf{R}_{w_f}(t) = \mathbf{R}^{(m_s(t))}_{w_f}(t)$ and $\mathbf{v}_{w_f,m}(t) = \mathbf{v}^{(m_s(t))}_{w_f,m}(t)$.

(vi) Using back-substitution calculate the parameters $\hat{\theta}_i(t)$, $i=1,...,m_s(t)$, from $\mathbf{R}_{w_f,m_s(t)}(t)$ which is the top-left triangular portion of the above matrix and $\mathbf{v}_{w_f,m_s(t)}(t)$ which contains the first $m_s(t)$ elements.

(vii) Compute the residual at the time instant t

$$\varepsilon(t) = y(t) - \sum_{i=1}^{m_s(t)} \phi_{w_f,i}(t)\,\hat{\theta}_{w_f,i}(t) \,.$$

In the initialization of the GFSL algorithm let all the elements in $\mathbf{R}_{w_f}(0)$ and $\mathbf{v}_{w_f,m}(0)$ equal a small positive number to prevent division by zero.

The sub-models which are generated at different time points may be different, therefore, the candidate variables should be sufficient to describe the dynamics of the system under test in a wide range of operation. Since the GFSL algorithm can select significant regressors on-line it allows new candidate variables to be added or some useless candidate variables to be removed on-line. These operations can be realized by means of using some "empty" variables and modifying the dimension of the augmented matrix in the computational process, just like the GFEX algorithm (Luo et al 1994).

The range of the size of the window, w_l can be wide and depends on the knowledge about the system under investigation, e.g. the system dynamics, the selection of the sampling interval and the computational facilities. Roughly speaking, the faster the system structure varies, the smaller w_l. If the forgetting factor λ is chosen between 0.98 to 0.995 in exponential windowing algorithms for a system the asymptotic memory length N_a is 50 ~ 200 based on the concept of $N_a = \frac{1}{1-\lambda}$ (Clarke and Gawthrop 1975). As a reference w_l should be selected to be larger than 50 for the same system. If the computations becomes unstable w_l should typically be increased to provide more data in the window. This will be discussed in more detail later. Conversely a large w_l can lead to smooth estimates but may produce large estimation deviations to a sudden changes. If w_l is very large and the values of the candidate variables are also large a scaling procedure should be adopted to reduce the amplitude of the signals to be processed and hence to avoid overflow in the computations which may arise because this situation is similar to the computations of the GFEX algorithm with λ=1.

6. Properties of the GFSL Algorithm

The sub-models produced by the GFSL algorithm are local models based on the data window at specific time points. As old data are removed and new measurements are added the sub-models change and the local windowing effect is relatively obvious. However when the previous exponential windowing algorithm (GFEX) is used the data in an observation window decay exponentially. The sub-models from the GFEX algorithm therefore weight out the effect of old data and may better reflect the present system state.

The computation of adding a row using the Givens rotation method is numerically stable, but the deletion procedures described in (3-11) and (3-16) are potentially destabilising, even though there are no roundoff errors in the deletion process. For example, in (3-16), $\underline{c}^*_{b_n,kk}$ is computed from the square root of the difference between two numbers and then used as a denominator to produce b_c and b_s. Since $\underline{c}_{b_n,kk}$ and $\underline{c}^{(k-1)}_{b_n,m+1,k}$ are always stored in limited precision they cannot be recovered as in infinite precision and hence the difference between them will not be exact. If the elements in the row to be deleted are large so that $\underline{c}^*_{b_n,kk}$ approaches zero, the deletion operation may cause numerical instabilities. Such problems usually happen in circumstances where the size of the window, w_l, is small but the amplitude of the signals to be processed changes rapidly. Fortunately, most of applications with rectangular windowing methods use

relatively large w_l values.

The above deletion procedures involve m subtractions in computing the diagonal elements of $\mathbf{R}_{w_b}$ at every computational interval and the instability of the algorithm may increase following the increase in the number of candidate variables. Golub and Styan (1973) suggested a stable method for deleting a row which only involves one subtraction in the computation for the diagonal elements of $\mathbf{R}_{w_b}$, but requires $9m^2/2$ multiplications and $2m-1$ square roots. Considering the computational cost and complexity, it is better to use this method to process cases where roundoff errors are likely to be serious. Although this method is efficient, the sliding window methods should use relatively large windows and are best suited to systems where the difference of the amplitude of signals is not very large. This is because the stability of the operation is still limited by the condition number of $\mathbf{R}_{w_b}(t)$. It is better to monitor the occurrence of instabilities by measuring some important variables, e.g. $c^*_{b_n,kk}$ in (3-16).

In respect of the computational cost, it is difficult to give an exact comparison between the GFSL algorithm and the off-line orthogonal algorithms when they are applied in the rectangular windowing case, because the computational cost is always associated with the structure detection. Roughly speaking, the larger w_l, the more the computational savings using the GFSL algorithm. For example, when 10 regressors are selected from 20 candidate variables in a computational interval, the computational cost of the GFSL algorithm will be lower than the off-line modified Gram-Schmidt algorithm (Chen et al 1989) if $w_l > 50$. Compared with the exponential windowing algorithm (GFEX) which only has the adding operation, the GFSL algorithm requires more computation because of the deletion operation, but the cost is not very much more than that of detecting system structure.

7. Numerical Results

The GFSL algorithm is suited to slowly time-varying systems and works best with large windows. The results from two typical examples will be presented below to illustrate this algorithm.

Experiment 1: a simulation system

This is a slowly time-varying system described as

$$z(t) = \begin{cases} 0.5\, z(t-1) + u(t-1) & 1 - 250\ th\ \ point \\ 0.4\, z(t-1) + u(t-1) + 0.05u^2(t-1) & 251 - 500\ th\ \ point \\ 0.3\, z(t-1) + 0.9u(t-1) + 0.075u^2(t-1) & 501 - 750\ th\ \ point \\ 0.2\, z(t-1) + 0.8u(t-1) + 0.1u^2(t-1) & 751 - 1000\ th\ \ point \end{cases}$$

$$y(t) = z(t) + e(t) \ .$$

In the test, the input signal $u(t)$ was an independent uniformly distributed sequence with zero mean and variance 1.03, and the noise signal $e(t)$ was Gaussian white with zero mean and variance 0.01. The input and output signals are illustrated in Fig. 1. 1000 data generated by the above system were processed using the GFSL algorithm with the initial specification $n_y=n_u=n_\epsilon=2$, $n_l=2$, $\xi_s=0.01$, $w_l=200$. The sub-models at the 250, 500, 750 and 1000 th point are shown in Table 1. The predicted output and residuals plotted in Fig.2 also show that the fitting is quite good. Since ξ_s is usually unknown a priori, it is possible to miss some regressors which make a very small contribution when ξ_s is larger than the optimal cutoff. See for example the case at the 500'th point where $u^2(t-1)$ is missed. In contrast, a small ξ_s may lead to the appearance of insignificant terms in a selected sub-model, e.g. $y^2(t-1)$ at the 900'th point (Fig.3d). But such terms usually make a very small contribution to the output and emerge randomly.

Inspection of the residuals (Fig. 2b) shows that the errors always rise at the beginning of every variation in the system structure and then reduce gradually. From this the effect of the movement of the sliding window is evident. The curves of the parameters look smooth and exhibit the change in the system structure and parameters. When the contribution of $u^2(t-1)$ at the 500'th point increases this term is selected in the sub-model.

The on-line validity tests with a sliding window (Appendix) shown in Fig. 4 provide the validity of the sub-models. The tests use the smoothed values of the means with an average factor of 1.25 times the length w_l and τ=15. It is observed that $\rho_{\epsilon\epsilon}(t)$ rises in the interval between the 250 to 500'th point due to the missing of $u^2(t-1)$. There are several large jumps in the plots of $\rho_{u^2\epsilon}$ and $\rho_{u^2\epsilon^2}$. Most of these appear in cases when the data in the window belong to more than one system structure. In such a case the basic assumption in deriving the tests cannot be satisfied. This causes the results of the tests to fluctuate locally. Nevertheless, most of the portions of the plots are within the confidence intervals.

Experiment 2: a large pilot scale liquid level system

A description of this system is given by Billings and Voon (1986b). 1000 data are used in the present experiment and the input and output are illustrated in Fig. 5.

A time-varying NARMAX model was fitted using the GFSL algorithm with $n_y=n_u=n_\epsilon=2$, $n_l=2$, $\xi_s=0.0001$ and $w_l=150$. The number of candidate variables is 28. The results are shown in Fig. 6, 7 and Table 2 and indicates that the estimator can track the

changes in the system. At the 1000'th point $\sum_{t=1}^{1000}\varepsilon(t)$ is 2.468 and the sub-model only involves 6 terms. As a comparison, a global model is shown in Fig. 9, 10 and Table 3, this was produced by the off-line classical Gram-Schmidt algorithm with the initial parameters $n_y=n_u=4$, $n_\varepsilon=0$, $n_{l=3}$, AIC =4.0 (Akaike's Information Criterion). The off-line estimator selects 15 terms from 165 candidate variables and results in $RSS=2.571(=\sum_{t=1}^{1000}\varepsilon(t))$. Although there is no large difference between two values at the last point, the structure of the global model is more complex than that of the time-varying model. The result of the global model shows that the first two terms, $y(t-1)$ and $u(t-1)$, make large contributions to the output (totalling 98.65%). The result at the 1000th point of the time-varying model shows that the same terms make a 97.65% contribution in the last window. The curve of the estimated parameters (Fig. 7) illustrates that these two terms are always selected in the sub-models over all time. The variance of the residuals σ_ε^2 (2.404×10^{-3}) can be compared with the result (2.571×10^{-3}) produced from the off-line estimator. This also approaches the result (2.018×10^{-3}) produced from a recursive prediction error parameter estimator in which the system structure has been determined (Chen and Billings 1988).

Ignoring the instability at the initial stage the on-line validity tests shown in Fig. 8 suggest that the time-varying model is acceptable. It is also observed that the curves of $\rho_{u^2\varepsilon}$ and $\rho_{u^2\varepsilon^2}$ have several peaks. The fixed sub-models at these points may not be reasonable due to changes in the system. Another reason could be related to the basic assumption of the tests. For this, the analysis is similar to Experiment 1.

8. Conclusions

A recursive orthogonal QR algorithm based on a rectangular sliding window and referred to as the GFSL algorithm has been derived. This algorithm uses a unity weighting factor over a window of data and adaptively adjusts the model structure on-line. In continuous computation a series of sub-models over a sliding window are obtained and only a small amount of memory space is required. Compared with the previous exponential windowing algorithm this new GFSL algorithm requires data deletion at each step and this incurs additional computations. This small additional cost has to be offset by the improved detection of changing model structure. When the size of the sliding window is relatively large, the computational cost for updating the estimates in a window is lower than that of the off-line orthogonal QR algorithms.

The algorithm determines the contribution that the candidate variables make to the output to determine which variables should be included in the final model before the corresponding parameter estimates are updated. Since the information in an observation window is represented concisely in the augmented matrix, the completeness of the regression information can ensure the correct determination of the system structure. If the system structure does not change over time the resulting parameter estimates will converge to the true values in the corresponding rectangular window.

Because the GFSL algorithm uses stable Givens rotations, the numerical performance is similar to the exponential windowing algorithm in most applications. But when the size of the window is relatively small and the signals change rapidly, the data deletion procedures can sometimes cause numerical problems. The GFSL algorithm is therefore more suited to slowly time-varying systems where the input-output signals are relatively smooth and larger windows can be employed.

Acknowledgments

SAB gratefully acknowledges that part of this work was supported by SERC under grant GR/H/35286.

Table 1: Experiment 1 (Simulation System, Time-Varying Model)

point	terms	θ	$\hat{\theta}$	*ERR*
250 th	$u(t-1)$	1.000000*E*+00	1.006801*E*+00	7.560013*E*−01
	$y(t-1)$	5.000000*E*−01	4.894617*E*−01	2.394044*E*−01
	$u^2(t-1)$	none	none	*
500 th	$u(t-1)$	1.000001*E*+00	1.008868*E*+00	8.194872*E*−01
	$y(t-1)$	4.000001*E*−01	4.153543*E*−01	1.714760*E*−01
	$u^2(t-1)$	5.000000*E*−02	none	*
750 th	$u(t-1)$	9.000001*E*−01	9.061748*E*−01	8.907980*E*−01
	$y(t-1)$	3.000001*E*−01	3.001955*E*−001	9.290199*E*−02
	$u^2(t-1)$	7.500000*E*−02	7.536118*E*−02	1.055724*E*−02
1000 th	$u(t-1)$	8.000001*E*−01	7.926466*E*−01	9.280992*E*−01
	$y(t-1)$	2.000001*E*−01	2.029505*E*−01	4.196139*E*−02
	$u^2(t-1)$	1.000000*E*−01	9.237280*E*−02	2.203432*E*−02

t = 1000 $\quad w_l$ = 200

ξ_s = 0.01 $\quad \varepsilon^T\varepsilon/y^Ty$ = 8.867051*E*−03

$\sum_{t=1}^{1000}\varepsilon^2(t)$ = 9.549729*E*+00 $\quad \sigma_\varepsilon^2$ = 9.267196*E*−03

Table 2: Experiment 2 (Liquid Level System, Time-Varying Model)

point	term	$\hat{\theta}$	*ERR*
1000	$y(t-1)$	7.187055*E*−01	9.479078*E*−01
	$u(t-1)$	3.907416*E*−01	2.857187*E*−02
	constant	−4.657357*E*−02	3.893125*E*−03
	$u(t-2)$	−1.299308*E*−01	1.126658*E*−03
	$y(t-1)u(t-1)$	−2.027587*E*−01	6.353696*E*−04
	$y(t-2)$	1.590202*E*−01	3.707633*E*−04

t = 1000 $\quad w_l$ = 150

ξ_s = 0.0001 $\quad \varepsilon^T\varepsilon/y^Ty$ = 6.026616*E*−03

$\sum_{t=1}^{1000}\varepsilon^2(t)$ = 2.467572*E*+00 $\quad \sigma_\varepsilon^2$ = 2.404027*E*−03

Table 3: Experiment 2 (Liquid Level System, Global Model, CGS)

term	$\hat{\theta}$	*ERR*	St.Dev.
$y(t-1)$	7.085734*E*−01	9.679613*E*−01	3.556876*E*−02
$u(t-1)$	3.884508*E*−01	1.854159*E*−02	1.247766*E*−02
$u(t-2)$	−8.500008*E*−02	3.367434*E*−03	1.359262*E*−02
$y(t-1)y(t-2)y(t-3)$	−2.881762*E*−02	1.084866*E*−03	5.630327*E*−03
$y(t-1)u(t-1)$	−3.518114*E*−01	7.035443*E*−04	3.157156*E*−02
$y(t-1)y(t-4)u(t-2)$	−2.742458*E*−01	1.069390*E*−03	4.316852*E*−02
$y(t-2)$	2.593803*E*−01	1.882639*E*−04	3.379767*E*−02
$y(t-2)u(t-2)$	3.137017*E*−01	2.081952*E*−04	5.555111*E*−02
$y(t-2)u(t-3)$	−1.969853*E*−01	1.245256*E*−04	5.406829*E*−02
$y(t-1)u(t-1)u(t-2)$	−1.887223*E*−01	1.159556*E*−04	3.118323*E*−02
$y(t-2)u(t-4)$	1.421650*E*−01	1.026571*E*−04	2.761783*E*−02
$y^2(t-1)$	−3.645250*E*−02	8.059901*E*−05	1.085798*E*−02
$u^2(t-4)$	−4.763843*E*−02	4.487067*E*−05	1.332959*E*−02
$y(t-1)y(t-4)u(t-3)$	1.335944*E*−01	3.938991*E*−05	3.223927*E*−02
$y^2(t-1)u(t-1)$	1.093884*E*−01	8.891241*E*−05	2.907003*E*−02

n = 1000	No. of terms = 165
criterion *AIC* =4.0	$\varepsilon^T\varepsilon/y^T y$ = 6.278502*E*−03
RSS = 2.570706*E*+00	σ_ε^2 = 2.571030*E*−03

Appendix

On-line Model Validity Tests with Sliding Windows

Because the weighting factors are unity in the sliding window estimation the choice of on-line model validity tests depends on the size of the window w_l, the dimension of memory space and the limit of computational time in every interval. Assume that the maximum number of the lags to be used is τ_{max}. When the latest $w_l+\tau_{max}$ records of inputs and residuals are available and the computational time available is sufficient, the off-line validity tests (Billings and Voon 1983, Billings and Voon 1986a) can be applied. If w_l is relatively large and the memory requirements and computational time for the off-line tests are excessive, the on-line validity tests for exponential windowing algorithms (Luo et al 1994) may be considered. In such an implementation, although the weighting factor λ is unity in a observation window, the

calculation for the forgetting factor η of the validity tests cannot use $\eta = 1/(2-\lambda)$, because substituting $\lambda=1$ into the above equation yields $\eta=1$ and hence the asymptotic memory length N_a ($N_a = \eta/(1-\eta)$), which is equal to the window length, becomes infinite. This is clearly not correct for the sliding window case because the number of data sets in a sliding window is always finite. Therefore if the size of the window w_l can be designed as the asymptotic memory length N_a the forgetting factor η can be determined using $\eta \approx \frac{w_l}{w_l + 1}$. For example, when w_l=100, η=0.99. If the computational time is limited but the memory space is sufficient to store the latest $w_l+\tau_{max}$ records of the input and residual, a set of on-line model validity tests with sliding windowing can be developed. This represents a compromise between completeness of the validity tests and complexity of the computation. The basic assumptions are that in a wide interval the means of the samples are close to zero and the means of the squares of samples do not vary greatly. Such conditions often exist in cases where a large window is applied to systems which vary slowly and smoothly.

Similar to the on-line validity tests for exponential windowing methods, a new set of functions are defined by taking the average of the off-line correlation functions (Billings and Voon 1983, Billings and Voon 1986a), which are associated with finite lags and the w_l data samples up to time t and are given as

$$\rho_{ab}(\tau,w_l,t) = \frac{1}{\tau+1}\sum_{k=0}^{\tau}\Psi_{ab}(k,w_l,t)$$

where τ is the number of the lags and

$$\Psi_{ab}(k,w_l,t) = \frac{1}{w_l}\sum_{i=t-w_l+1}^{t} a(i-k)b(i) \quad .$$

When the observation window slides over the data the newest measurements can be successively used in the tests and the oldest samples can be deleted from previously formed windows. From the results of these tests, it is possible to observe whether the fitted model is adequate or not in on-line identification. The normalized functions of the five validity tests are given as (the subscript s denotes normalization)

$$\rho_{\varepsilon\varepsilon_s}(\tau,w_l,t) = \frac{\rho_{\varepsilon\varepsilon}(\tau,w_l,t)}{\Psi_{\varepsilon\varepsilon}(0,w_l,t)} = \frac{\rho_{\varepsilon\varepsilon_r}(\tau,w_l,t)}{\rho_{\varepsilon\varepsilon_r}(0,w_l,t)} \qquad \text{(A-1)}$$

$$\rho_{u\varepsilon_s}(\tau,w_l,t) = \frac{\rho_{u\varepsilon}(\tau,w_l,t)}{\sqrt{\Psi_{uu}(0,w_l,t)\Psi_{\varepsilon\varepsilon}(0,w_l,t)}} = \frac{\rho_{u\varepsilon_r}(\tau,w_l,t)}{\sqrt{\rho_{uu_r}(0,w_l,t)\rho_{\varepsilon\varepsilon_r}(0,w_l,t)}} \qquad \text{(A-2)}$$

$$\rho_{\varepsilon(\varepsilon u)_s}(\tau,w_l,t) = \frac{\rho_{\varepsilon(\varepsilon u)}(\tau,w_l,t)}{\sqrt{\Psi^2_{\varepsilon\varepsilon}(0,w_l,t)\Psi_{uu}(0,w_l,t)}} = \frac{\rho_{\varepsilon(\varepsilon u)_r}(\tau,w_l,t)}{\sqrt{\rho^2_{\varepsilon\varepsilon_r}(0,w_l,t)\rho_{uu_r}(0,w_l,t)}} \qquad \text{(A-3)}$$

$$\hat{\rho}_{u^2\varepsilon_s^2}(\tau,w_l,t) = \frac{\rho_{u^2\varepsilon^2}(\tau,w_l,t)}{\sqrt{\Psi_{u^2u^2}(0,w_l,t)\ \Psi_{\varepsilon\varepsilon}(0,w_l,t)}} = \frac{\rho_{u^2\varepsilon_r^2}(\tau,w_l,t)}{\sqrt{\rho_{u^2u^2_r}(0,w_l,t)\ \rho_{\varepsilon\varepsilon_r}(0,w_l,t)}} \qquad \text{(A-4)}$$

$$\hat{\rho}_{u^2\varepsilon^2_s}(\tau,w_l,t) = \frac{\rho_{u^2\varepsilon^2}(\tau,w_l,t)}{\sqrt{\Psi_{u^2u^2}(0,w_l,t)\ \Psi_{\varepsilon^2\varepsilon^2}(0,w_l,t)}} = \frac{\rho_{u^2\varepsilon\varepsilon_r}(\tau,w_l,t)}{\sqrt{\rho_{u^2u^2_r}(0,w_l,t)\ \rho_{\varepsilon^2\varepsilon^2_r}(0,w_l,t)}} \qquad \text{(A-5)}$$

where

$$\rho_{\varepsilon\varepsilon_r}(\tau,w_l,t) = \Big(\sum_{i=t-w_l}^{t-1}\overline{\varepsilon_\tau(i)}\varepsilon(i)\Big) - \overline{\varepsilon_\tau(t-w_l)}\varepsilon(t-w_l) + \overline{\varepsilon_\tau(t)}\varepsilon(t)$$

$$\rho_{u\varepsilon_r}(\tau,w_l,t) = \Big(\sum_{i=t-w_l}^{t-1}\overline{u_\tau(i)}\varepsilon(i)\Big) - \overline{u_\tau(t-w_l)}\varepsilon(t-w_l) + \overline{u_\tau(t)}\varepsilon(t)$$

$$\rho_{\varepsilon(\varepsilon u)_r}(\tau,w_l,t) = \Big(\sum_{i=t-w_l}^{t-1}\overline{D_\tau(i)}\varepsilon(i)\Big) - \overline{D_\tau(t-w_l)}\varepsilon(t-w_l) + \overline{D_\tau(t)}\varepsilon(t)$$

$$\rho_{u^2\varepsilon_r}(\tau,w_l,t) = \Big(\sum_{i=t-w_l}^{t-1}\overline{C_\tau(i)}\varepsilon(i)\Big) - \overline{C_\tau(t-w_l)}\varepsilon(t-w_l) + \overline{C_\tau(t)}\varepsilon(t)$$

$$\rho_{u^2\varepsilon^2_r}(\tau,w_l,t) = \Big(\sum_{i=t-w_l}^{t-1}\overline{C_\tau(i)}\big(\varepsilon^2(i)-\overline{\varepsilon^2_{w_l}(i)}\big)\Big) - \overline{C_\tau(t-w_l)}\big(\varepsilon^2(t-w_l)-\overline{\varepsilon^2_{w_l}(t-w_l)}\big)$$

$$+ \overline{C_\tau(t)}\big(\varepsilon^2(t)-\overline{\varepsilon^2_{w_l}(t)}\big)\Big)$$

$$\rho_{\varepsilon\varepsilon_r}(0,w_l,t) = \Big(\sum_{i=t-w_l}^{t-1}\varepsilon^2(i)\Big) - \varepsilon^2(t-w_l) + \varepsilon^2(t)$$

$$\rho_{uu_r}(0,w_l,t) = \Big(\sum_{i=t-w_l}^{t-1}u^2(i)\Big) - u^2(t-w_l) + u^2(t)$$

$$\rho_{u^2u^2_r}(0,w_l,t) = \Big(\sum_{i=t-w_l}^{t-1}\big(u^2(i)-\overline{u^2_{w_l}(i)}\big)^2\Big) - \big(u^2(t-w_l)-\overline{u^2_{w_l}(t-w_l)}\big)^2 + \big(u^2(t)-\overline{u^2_{w_l}(t)}\big)^2$$

$$\rho_{\varepsilon^2\varepsilon^2_r}(0,w_l,t) = \Big(\sum_{i=t-w_l}^{t-1}\big(\varepsilon^2(i)-\overline{\varepsilon^2_{w_l}(i)}\big)^2\Big) - \big(\varepsilon^2(t-w_l)-\overline{\varepsilon^2_{w_l}(t-w_l)}\big)^2 + \big(\varepsilon^2(t)-\overline{\varepsilon^2_{w_l}(t)}\big)^2$$

$$\overline{\varepsilon_\tau(i)} = \frac{1}{\tau+1}\sum_{k=0}^{\tau}\varepsilon(i-k)\ ,\quad \overline{u_\tau(i)} = \frac{1}{\tau+1}\sum_{k=0}^{\tau}u(i-k)\ ,\quad \overline{D_\tau(i)} = \frac{1}{\tau+1}\sum_{k=0}^{\tau}\varepsilon(i-k-1)\,u(i-k-1)$$

$$\overline{C_\tau(i)} = \frac{1}{\tau+1}\sum_{k=0}^{\tau}[u^2(i-k)-\overline{u^2_{w_l}(i)}]\ ,\quad \overline{u^2_{w_l}(i)} = \frac{1}{w_l}\sum_{t_1=i-w_l+1}^{i}u^2(t_1)\ ,\quad \overline{\varepsilon^2_{w_l}(i)} = \frac{1}{w_l}\sum_{t_1=i-w_l+1}^{i}\varepsilon^2(t_1)$$

The symbol $\bar{\ }$ denotes the time average and the symbol $'$ denotes that the values have had the mean removed.

The nine instruments with the subscript r in Eqn. (A-1 - A-5) provide a measure of the model validity for the recursive computation with a sliding window. The five correlation functions involve nine recursive operations and twelve calculations for means ($\overline{\varepsilon_\tau(i)}$, $\overline{u_\tau(i)}$ $\overline{D_\tau(i)}$, $\overline{C_\tau(i)}$, $\overline{u^2_{w_l}(i)}$ and $\overline{\varepsilon^2_{w_l}(i)}$, $i=t-w_l, t$). These average values can also be computed recursively so the computational cost is reduced. For example, when w_l is relatively large, the calculation of $\overline{\varepsilon^2_{w_l}(i)}$ can be obtained using

$$\overline{\varepsilon^2_{w_l}(t)} = \overline{\varepsilon^2_{w_l}(t-1)} - \frac{\varepsilon^2(t-w_l)}{w_l} + \frac{\varepsilon^2(t)}{w_l} \quad .$$

For cases where w_l is a relatively small value the means should be replaced by the smoothed values calculated over a length of 2 ~ 4 times w_l. Typically τ is usually 5 ~ 20. Therefore, the implementation of these tests not only has low computational cost but also requires small memory space.

Based on the results of the off-line model validity tests (Billings and Voon 1983, Billings and Voon 1986a), the values of the functions $\rho_{\varepsilon\varepsilon_s}(\tau,w_l,t)$, $\rho_{u\varepsilon_s}(\tau,w_l,t)$, $\rho_{\varepsilon(\varepsilon u)_s}(\tau,w_l,t)$, $\hat{\rho}_{u^2\varepsilon_s}(\tau,w_l,t)$ and $\hat{\rho}_{u^2\varepsilon^2_s}(\tau,w_l,t)$ all lie within the range from -1 to 1. If the fitted model is adequate at time t, $\rho_{\varepsilon\varepsilon_s}(\tau,w_l,t)$ should be near zero and the other functions should be zero. If the size of the window is large the standard deviation of the correlation is $1/\sqrt{w_l}$ and the 95% confidence intervals are defined as approximately $\pm 1.96/\sqrt{w_l}$.

References

Billings, S.A. and W.S.F. Voon (1983): Structure detection and model validity tests in the identification of nonlinear systems, IEE Proc., 1983, Pt.D, Vol. 130, pp. 193-199.

Billings, S.A. and W.S.F. Voon (1986a): Correlation based model validity tests for nonlinear models, Int.J.Systems Sci., 1986, Vol. 15, pp. 601-615.

Billings, S.A. and W.S.F. Voon (1986b): A prediction error and stepwise regression estimation algorithm for nonlinear systems; Int.J.Control, Vol. 44, pp. 803-822.

Billings,S.A. and W.S.F. Voon (1987): Piecewise linear identification of nonlinear systems, Int. J. Control, Vol. 46, No. 1, pp. 215-235, 1987.

Billings, B.A., S. Chen (1989): Extended model set, global data and threshed model identification of severely nonlinear systems, Int. J. Control. 1989, Vol. 50, No. 5, pp. 1897-1923.

Chambers, J.M. (1971): Regression updating. J. American Statistical Association, Vol. 66, No. 336, pp. 744-748, 1971.

Chen, S. and S.A. Billings (1988): A recursive prediction error parameter estimator for nonlinear models. Research Report No.326, Dept. of Control Engineering, University of Sheffield, U.K.

Chen, S. and S.A. Billings (1989): Representations of nonlinear systems: the NARMAX model. Int. J. Control, Vol. 48, 1989.

Chen, S., S.A. Billings, W. Luo (1989): Orthogonal least squares methods and their application to non-linear system identification, Int.J.Control, Vol. 50, pp. 1873-1896, 1989.

Clarke, D. and P.J. Gawthrop (1975): Self-tuning controller. Proc. Inst. Elec. Emg., Vol. 122, pp. 929-934, 1975.

Gentleman, W.M. (1973): Least squares computation by Givens transformation without square roots. J. Inst. Math. and Appl., Vol. 12, pp. 329-336, 1973.

Gentleman, W.M. (1974): Regression problems and the QR decomposition. Bulletin Inst. Math. and Appl., Vol. 10, pp. 195-197, 1974.

Golub, G.H. (1969): Matrix decompositions and statistical calculations. in R.C. Milton and J.A. Nelder (eds.), Statistical Computation, pp. 365-395, Academic Press, New York, 1969.

Golub, G.H. and G.P.H. Styan (1973): Numerical computations for univariate linear models. J. Statist. Comput. Simul., Vol. 2, pp. 253-274, 1973.

Leontaritis, I.J. and S.A. Billings (1985): Input-output parametric models for nonlinear systems, Part I: Deterministic nonlinear systems; Part II: Stochastic nonlinear systems; Int. J. Control, Vol. 41, pp. 303-344, 1985.

Liu, Y.P., M.J. Korenberg, S.A Billings. and M.B. Fadzil (1987): The nonlinear identification of a heat exchanger. 26th IEEE Conference on Decision and Control, Dec.9-11, 1987, Los Angeles, U.S.A.

Luo, W., S.A. Billings and K.M. Tsang (1994): On-line structure detection and parameter estimation with exponential windowing for nonlinear systems. (to be

submitted to Int. J. Control). Research Report No.503, Department of Automatic Control and Systems Engineering, University of Sheffield, U.K.

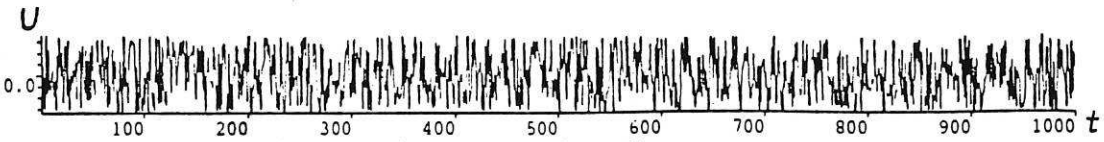

Fig. 1a Real Input (Experiment 1)

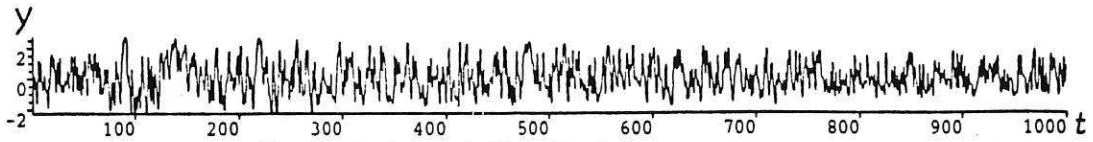

Fig. 1b Real Output (Experiment 1)

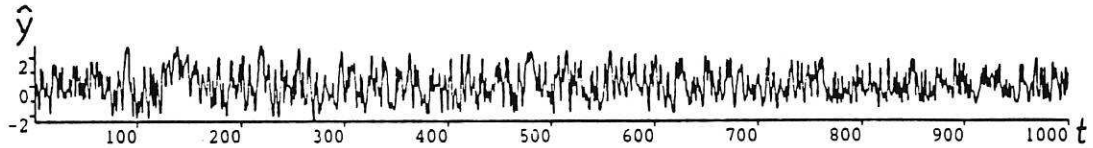

Fig. 2a Predicted Output (Experiment 1)

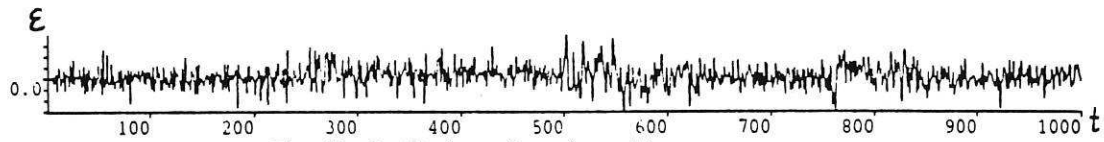

Fig. 2b Residuals (Experiment 1)

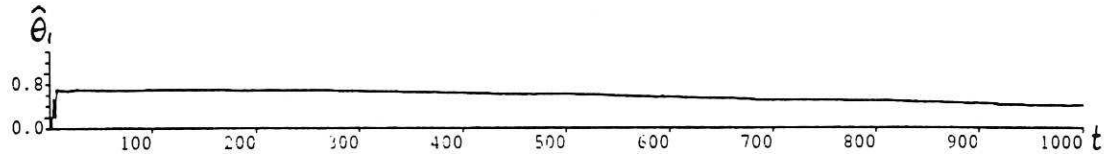

Fig. 3a Parameter of y(t-1) (Experiment 1)

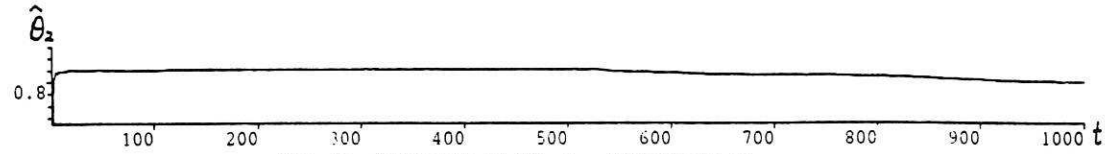

Fig. 3b Parameter of u(t-1) (Experiment 1)

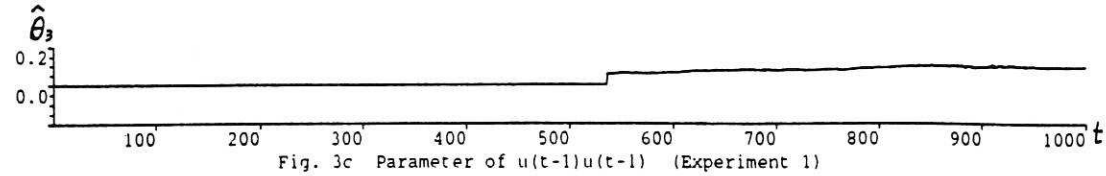

Fig. 3c Parameter of u(t-1)u(t-1) (Experiment 1)

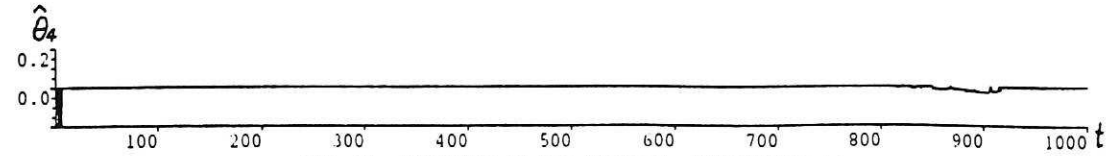

Fig. 3d Parameter of y(t-1)y(t-1) (Experiment 1)

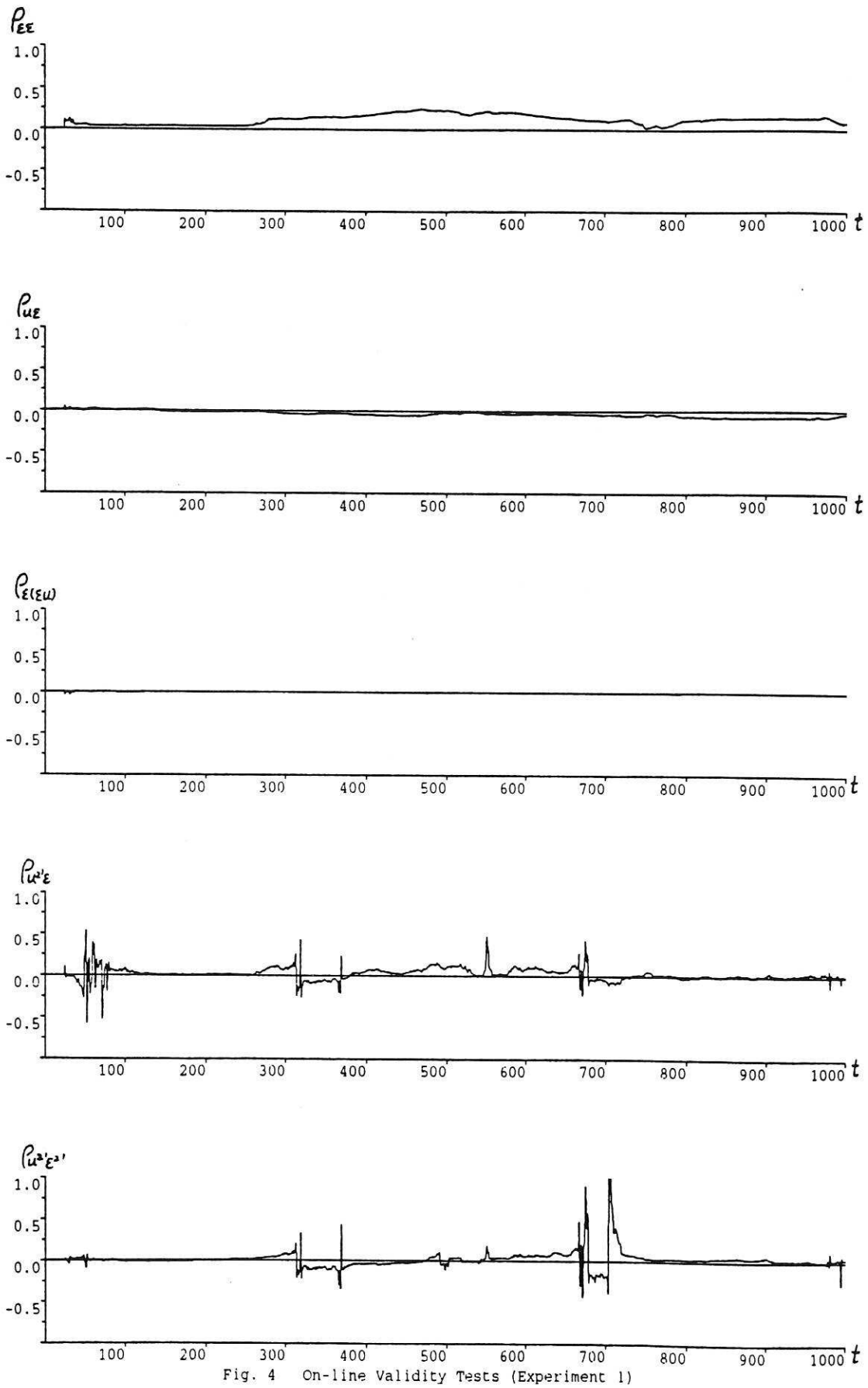

Fig. 4 On-line Validity Tests (Experiment 1)

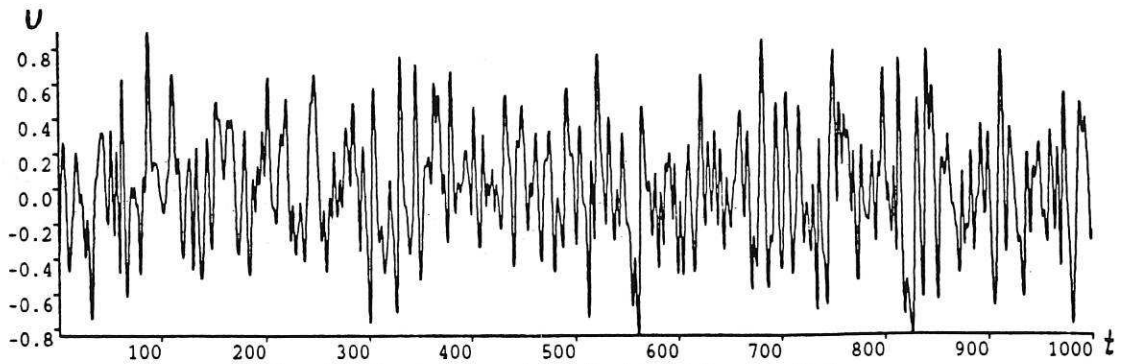

Fig. 5a Input (Experiment 2, Liquid Level Sytem)

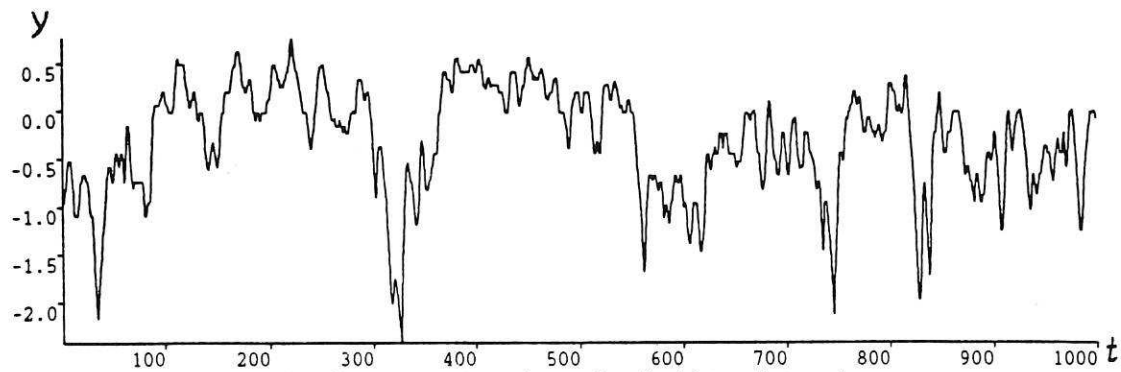

Fig. 5b Output (Experiment 2, Liquid Level Sytem)

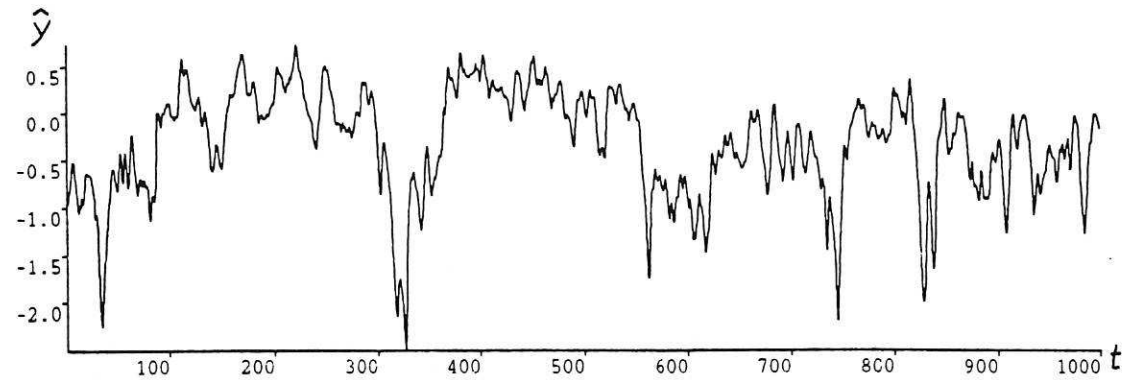

Fig. 6a Predicted Output (Experiment 2, Liquid Level Sytem)

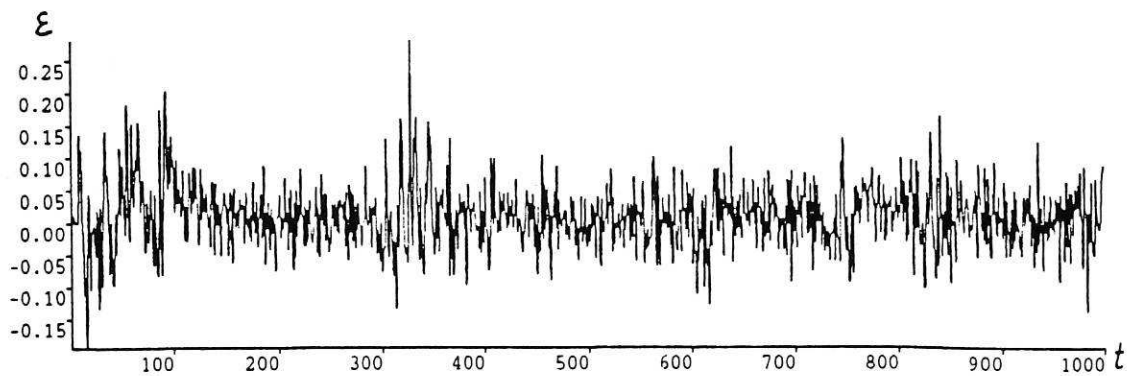

Fig. 6b Residuals (Experiment 2, Liquid Level Sytem)

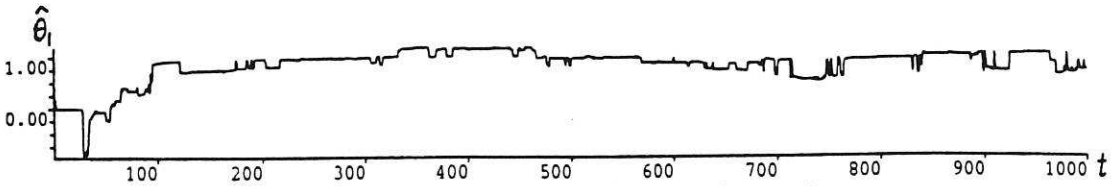

Fig. 7a Parameter of y(t-1) (Liquid Level System)

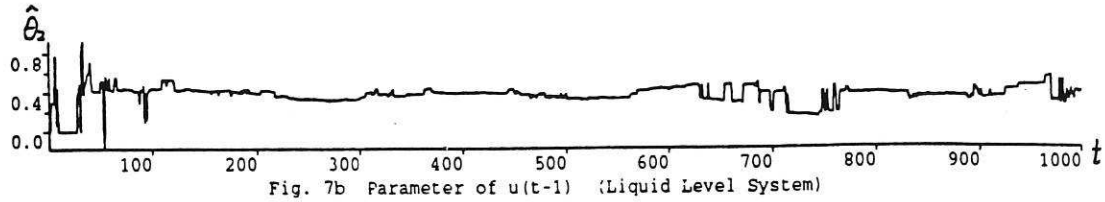

Fig. 7b Parameter of u(t-1) (Liquid Level System)

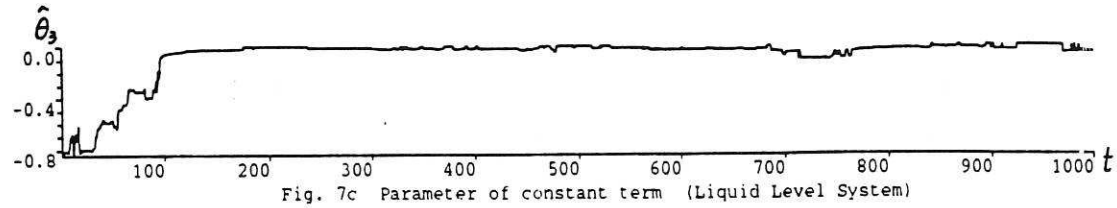

Fig. 7c Parameter of constant term (Liquid Level System)

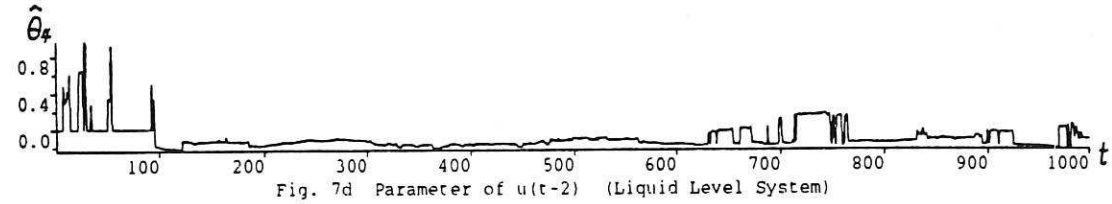

Fig. 7d Parameter of u(t-2) (Liquid Level System)

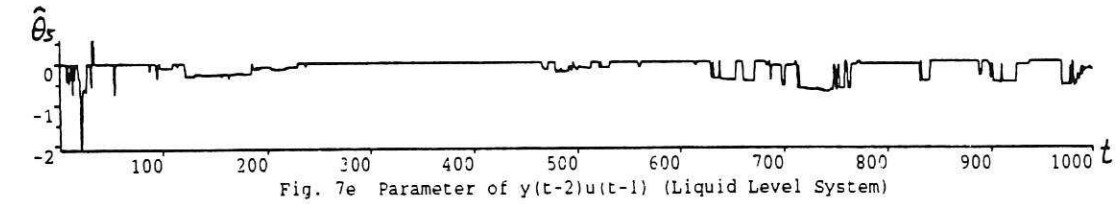

Fig. 7e Parameter of y(t-2)u(t-1) (Liquid Level System)

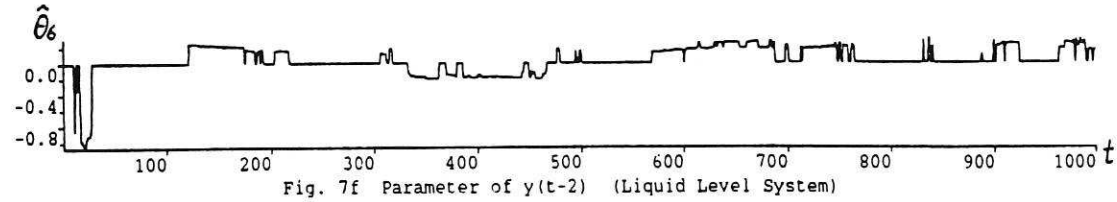

Fig. 7f Parameter of y(t-2) (Liquid Level System)

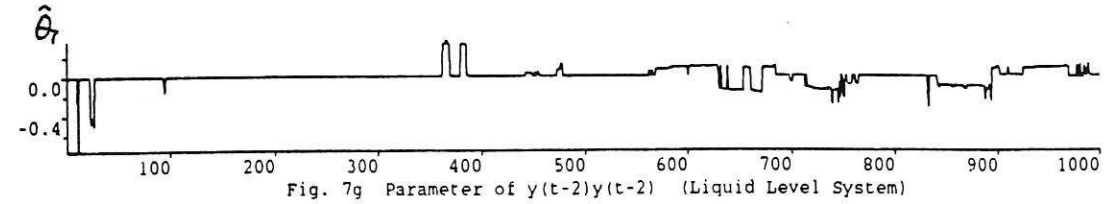

Fig. 7g Parameter of y(t-2)y(t-2) (Liquid Level System)

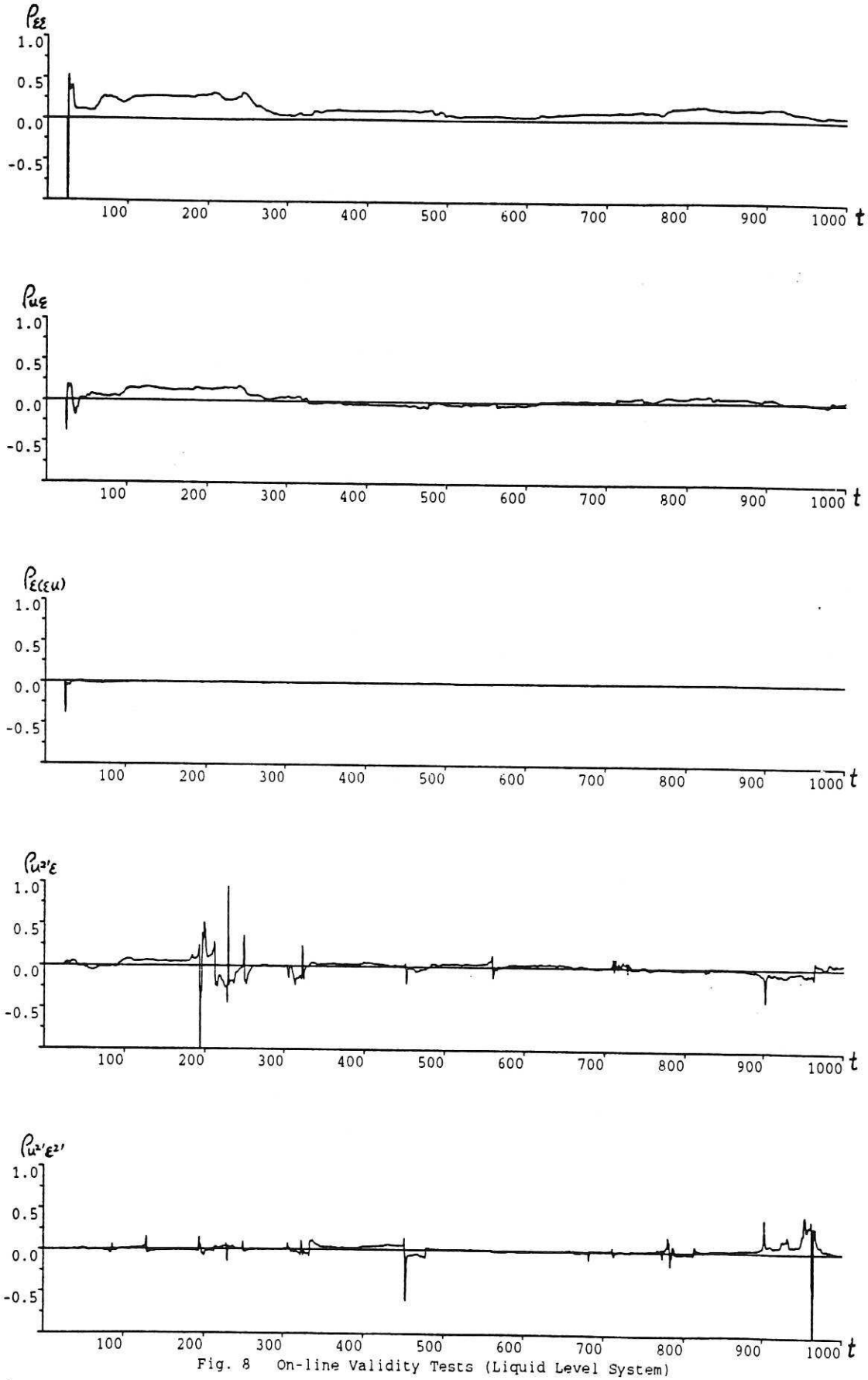

Fig. 8 On-line Validity Tests (Liquid Level System)

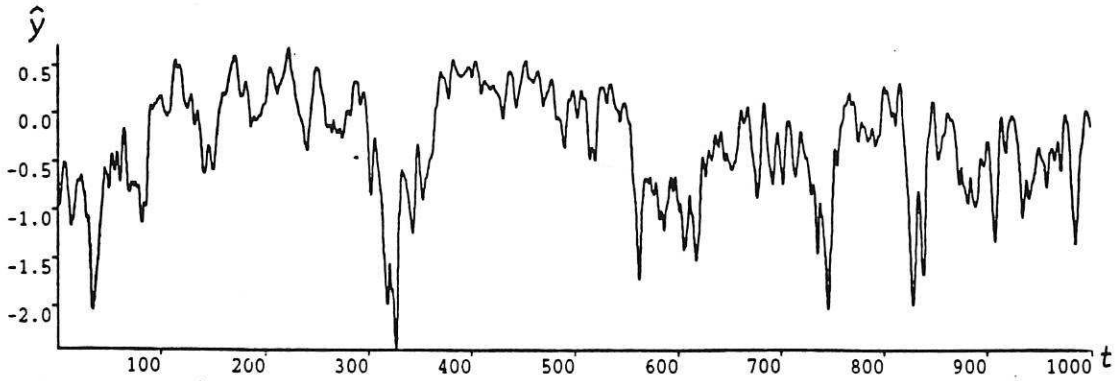

Fig. 9a Predicted Output (Liquid Level System, Global Model)

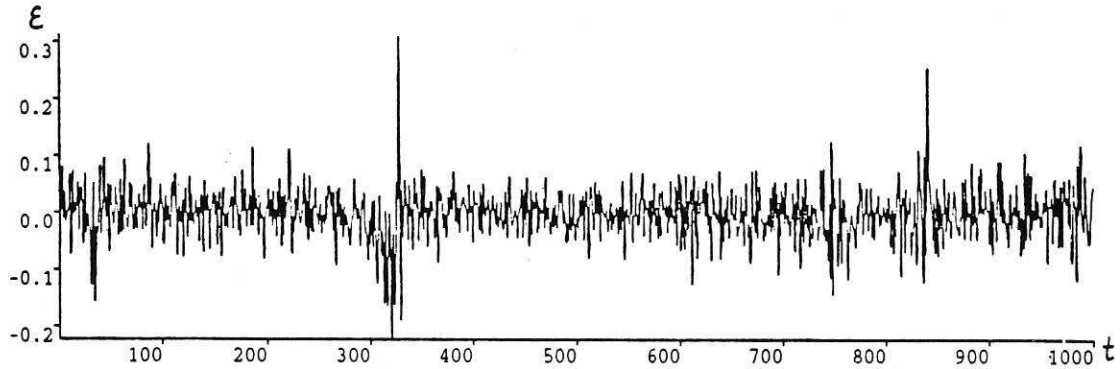

Fig. 9b Residuals (Liquid Level System, Global Model)

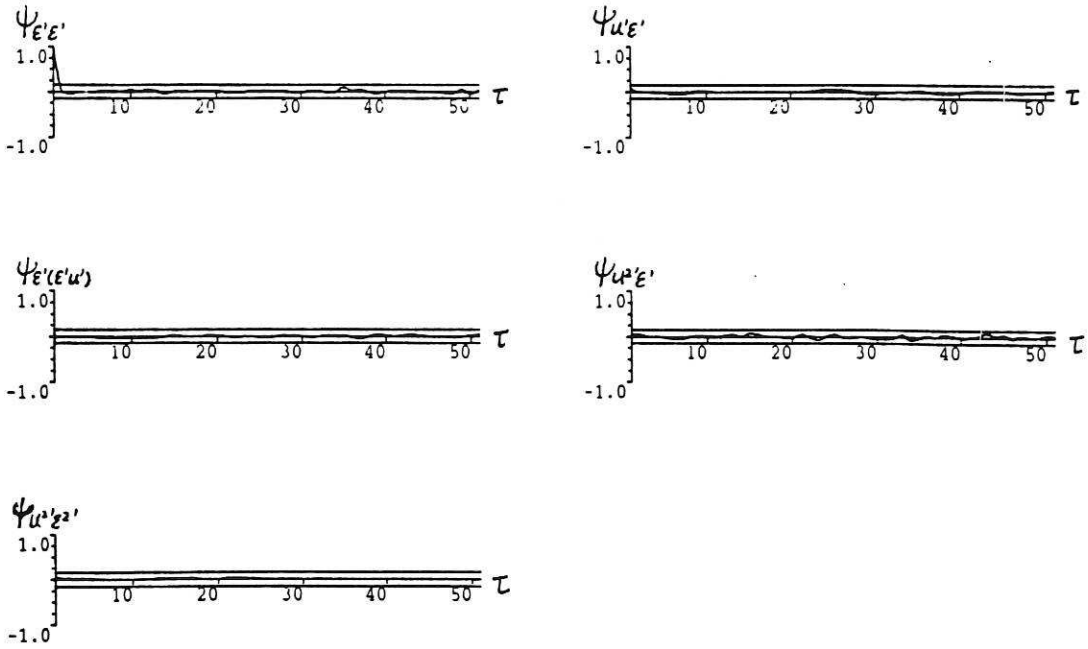

Fig. 10 Off-line valdity tests (Liquid Level System, Global Model)