Model-Predictive pH Control Using Real-Time NARX Approach

Thomas Pröll and M. Nazmul Karim
Dept. of Agricultural and Chemical Engineering, Colorado State University, Fort Collins, CO 80523

A new comprehensive real-time identification/control methodology based on the concept of nonlinear autoregressive exogenous input (NARX) models and adaptive, nonlinear, model-predictive control (ANMPC) is applied to a pH neutralization process. The existing NARX model theory has been extended by incorporating measured disturbances. NARX models have shown superior predictive characteristics in comparison to linear models. The proposed real-time methodology uses a pointer vector being created during an initial identification and model structure selection procedure. Using this pointer vector, which allocates the chosen elements from the pool of all possible linear and nonlinear combinations, one needs no explicit information about the model structure for the closed-loop control. The nonlinear programming problem encountered in ANMPC is solved by a gradient-based modified Marquardt and finite difference methods. The design procedure and explicit algorithms are discussed for the multiinput/multioutput case. A pH wastewater neutralization process used illustrates and verifies the procedure by computer simulations and real-time laboratory-scale experiments.

Introduction

A fundamental understanding of a typical industrial process is often limited by inaccessibility of system states, a very high number of parameters to account for the complexity of the system, unmeasured disturbances, measurement noise, and so on. Additionally, most processes encountered in industry also show nonlinear and time-variant behavior. To overcome most of these difficulties, Leontaritis and Billings (1985) introduced the nonlinear autoregressive moving average with exogenous input (NARMAX) model. They rigorously proved that a discrete-time, multivariable, nonlinear stochastic control system with m outputs and r inputs can be approximated arbitrarily well by a NARMAX model. A problem arises for the use of NARMAX [or nonlinear autoregressive exogenous input (NARX) obtained by dropping the moving average part] models due to the high number of possible candidate terms (Chen and Billings, 1989a,b; Korenberg et al., 1988). Several different approaches have been proposed to reduce the number of terms in the model, such as the structure determination which is complicated and time-consuming (Draper and Smith, 1981 and orthogonalization algorithms). Orthogonalization algorithms (Gram-Schmidt, modified Gram-Schmidt, Householder transformation) have proven to be more reliable and efficient by combining structure selection with parameter estimation (Korenberg, 1985; Kortmann and Unbehauen, 1987; Korenberg et al., 1988; Billings et al., 1988, 1989; Haber and Unbehauen, 1990).

The overall goal of this article is to develop a comprehensive identification/adaptive methodology for nonlinear time-varying processes. First, an extension to the original NARX model structure is introduced by incorporating measured disturbance terms. Even though the utilization of measured disturbance terms is nothing new (feedforward control, time-series analysis), this novel utilization extends the applicability of NARX models. However, the main contribution in terms of identification issues is to present the modified Gram-Schmidt (MGS) orthogonalization procedure within the framework of the proposed extended multivariable NARX model and to introduce a pointer vector. For the determination of the model complexity, a combined stopping criterion is used. In the dissection of the algorithm, it will be shown how a pointer vector is being created by simply storing the column number of the regressor matrix for the selected term. It is also demonstrated how to avoid ill-conditioning and how to incorporate explicit terms (a bias term) into the model structure. The pointer vector makes

it obsolete to have explicit knowledge about the model structure and is, in conjunction with the parameter estimates, a sufficient representation of the process.

Recent attempts to implement nonlinear models in model-predictive control (MPC) have shown superior performance, especially for applications in cases where large or frequent changes of operating conditions over nonlinear regions are expected (Garcia et al., 1989; Schmid and Biegler, 1990; McIntosh et al., 1991; Morningred et al., 1992; Henson and Seborg, 1992; Patwardhan et al., 1992; Hernández and Arkun, 1993). The essential feature of MPC is the minimization of a particular objective function with respect to the control and subject to constraints on output and manipulated variables (nonlinear programming problem or NLP). Limitations may arise due to the nonlinear model structure and, subsequently, the complex topology of the employed objective function. Therefore, if the topology of the objective function does not have a convex structure or has different local minima, classical quadratic minimization algorithms may fail.

This article also discusses the utilization of a modified Marquardt method (Marquardt, 1963; Edgar and Himmelblau, 1988) for solving the NLP problem. This gradient-based approach will always (for each time step and for each search step) guarantee a positive definite Hessian matrix and, therefore, convexity of the objective function. Backtracking is employed for each search direction to improve convergence time. A flow chart and an explicit algorithm listing is given. The development of the presented nonlinear identification/control methodology can be seen as an amalgamation and extension of previous techniques.

The computer-simulated and experimental application and verification of the derived nonlinear identification/adaptive control methodology on a pH neutralization process are presented, as well as comparisons for real-time implementation of the identification and control performance using linear ARX and NARX models.

System Representation

A measured disturbance term will be incorporated, in addition to the original NARX model definition of Leontaritis and Billings (1985) and a combined orthogonalization/model reduction algorithm which utilizes a combined performance criterion.

NARX model

The original NARX model definition of Leontaritis and Billings (1985) is:

$$y(t)=\bar{F}[y(t-1),\ \ldots,\ y(t-N_y),\ u(t-1),\ \ldots,\ u(t-N_u)]+\epsilon(t) \quad (1)$$

with the output, input and noise vector given by:

$$y(t)=\begin{bmatrix} y_1(t) \\ \vdots \\ y_m(t) \end{bmatrix},\quad u(t)=\begin{bmatrix} u_1(t) \\ \vdots \\ u_r(t) \end{bmatrix},\quad \epsilon(t)=\begin{bmatrix} \epsilon_1(t) \\ \vdots \\ \epsilon_m(t) \end{bmatrix}. \quad (2)$$

N_y and N_u are the maximum lags on all output and input channels. $\bar{F}$ is some nonlinear vector function, and $\{\epsilon(t)\}$ is some zero mean independent sequence.

The NARX model is a simplification of the NARMAX model, achieved by not incorporating delayed errors or residuals in the identification process. This simpler NARX model gives the advantage of tracking each output of a multiinput/multioutput (MIMO) process separately and independently at the expense of a slight loss of accuracy. In addition to the elements used in the original NARX model description, $[y(t-1),\ \ldots,\ y(t-N_y),\ u(t-1),\ \ldots,\ u(t-N_u)]$, a measured disturbance vector $d(t)=[d_1(t),\ \ldots,\ d_s(t)]^T$ is introduced. A MIMO/NARX model representation with $y(t)$ as the dependent variable vector and accommodating for a bias ($b(t)=[b_1(t),\ \ldots,\ b_m(t)]^T$) would then have the following form:

$$y(t)=b(t)+F[y(t-1),\ \ldots,\ y(t-N_y),\ u(t-1),\ \ldots,\ u(t-N_u),\ d(t-1),\ \ldots,\ d(t-N_d)]+\epsilon(t). \quad (3)$$

Generally, the nonlinear vector function $F=[f_1,\ \ldots,f_m]^T$ in Eq. 3 is not known. However, Chen and Billings (1989b) showed that these nonlinear functions $f_i(\cdot)$ can be arbitrarily well approximated by polynomial models which are specified by a maximum polynomial degree, $l\geq 1$. Each polynomial model has at least n terms, which would correspond to a maximum polynomial degree of exactly $l=1$. From the number of outputs, m, and inputs, r, measured disturbances, s, and the number of the corresponding maximum lags, N_y, N_u, and N_d, one gets n as:

$$n=m\cdot N_y+r\cdot N_u+s\cdot N_d. \quad (4)$$

The first $n+1$ linear elements, x_j, ($\forall j=0,\ \ldots,\ n$), of the polynomial which approximates the nonlinear function $f_i(\cdot)$, are given by:

$$\left.\begin{aligned} &x_0(t)=1 \\ &x_1(t)=y_1(t-1) && \cdots\ x_{m\cdot N_y}(t)=y_m(t-N_y) \\ &x_{m\cdot N_y+1}(t)=u_1(t-1) && \cdots\ x_{m\cdot N_y+r\cdot N_u}(t)=u_r(t-N_u) \\ &x_{m\cdot N_y+r\cdot N_u+1}(t)=d_1(t-1) && \cdots\ x_n(t)=d_s(t-N_d) \end{aligned}\right\} \quad (5)$$

Each additional term of the polynomial approximation is built from these n basic elements [without the bias term $x_0(t)$] by multiplying them together up to a polynomial degree of l. The resulting polynomial approximation can be written in the following compact form (for the ith dependent variable y_i):

$$y_i(t)=\sum_{j_1=0}^{n}\sum_{j_2=j_1}^{n}\cdots\sum_{j_l=j_{l-1}}^{n}\{[x_{j_1}(t)x_{j_2}(t)\ldots x_{j_l}(t)]\theta_j^{(i)}\}+\xi^{(i)}(t) \quad \forall i=1,\ \ldots,\ m \quad (6)$$

where the model parameters $\theta_j^{(i)}$, ($\forall\, j=0,\ \ldots,\ M$), are being placed successively as the elements of the parameter vector $\Theta^{(i)}=[\theta_0^{(i)},\ \ldots,\ \theta_M^{(i)}]^T$. From Eq. 6, one can clearly see, $l=1$ gives only linear elements and the bias term $b_i(t)=\theta_0^{(i)}$. Polynomial approximations with nonlinear elements are obtained for $l>1$. Each of the m models of Eq. 6 can be represented by a regression with the parameter vector, $\Theta^{(i)}=[\theta_0^{(i)},\ \ldots,\ \theta_M^{(i)}]^T$, and the

regressor matrix, $P=[p_0, \ldots, p_M]^T$, which is the same for every output. M is the number of all possible regressors given by:

$$\begin{Bmatrix} l=1: & M=n \\ l=2: & M=n+\sum_{j_2=0}^{n} j_2 \\ l=3: & M=n+\sum_{j_2=0}^{n} j_2+\sum_{j_2=0}^{n}\sum_{j_3=0}^{j_2} j_3 \\ & \vdots \quad \vdots \end{Bmatrix} \tag{7}$$

With the assumption that $N+1$ measurements are available ($N+1$ dependent/independent data pairs) the regression for the ith output can be written as:

$$\begin{bmatrix} y_i(t-N) \\ \vdots \\ y_i(t) \end{bmatrix} = \begin{bmatrix} 1 & x_1(t-N) & \cdots & x_n(t-N) & x_1^2(t-N) \\ \vdots & & & & \vdots \\ 1 & x_1(t) & \cdots & x_n(t) & x_1^2(t) \end{bmatrix}$$

$$\begin{matrix} x_1(t-N)x_2(t-N) & \cdots & x_1(t-N)x_n(t-N) \\ \vdots & & \vdots \\ x_1(t)x_2(t) & \cdots & x_1(t)x_n(t) \\ \cdots\ x_1^l(t-N) & \cdots & x_1^{l-1}(t-N)x_n(t-N) \\ \vdots & & \vdots \\ \cdots\ x_1^l(t) & \cdots & x_1^{l-1}(t)x_n(t) \end{matrix}$$

$$\begin{bmatrix} \cdots & x_n^l(t-N) \\ & \vdots \\ \cdots & x_n^l(t) \end{bmatrix} \begin{bmatrix} \theta_0^{(i)} \\ \vdots \\ \theta_M^{(i)} \end{bmatrix} + \begin{bmatrix} \xi^{(i)}(t-N) \\ \vdots \\ \xi^{(i)}(t) \end{bmatrix} \tag{8}$$

where $x_0(t)=1$ ($\forall\ t$) is used, see Eq. 5 and $\xi^{(i)}(t)$ is some modeling error. The $M+1$ column vectors of the regressor matrix P are defined as p_j, $\forall\ j=0, \ldots, M$. For the sake of shorter notation, the dependent variable, $z^{(i)}=[y_i(t-N), \ldots, y_i(t)]^T$, and the modeling error vector $\Xi^{(i)}=[\xi^{(i)}(t-N), \ldots, \xi^{(i)}(t)]^T$ are introduced. Therefore, Eq. 8 can be rewritten as:

$$z^{(i)}=P\Theta^{(i)}+\Xi^{(i)} \quad \forall\ i=1, \ldots, m. \tag{9}$$

Since the elements f_i of the nonlinear vector function, F, are not known *a priori*, the full model set (all possible terms of P) have to be considered at the beginning of the identification algorithm. The only variables which have to be chosen *a priori* are the maximum lags on the outputs, N_y, inputs, N_u, measured disturbances, N_d, and the maximum polynomial degree l. These variables have to be chosen intuitively or from *a priori* knowledge of the system such as maximum time lag. A full model set can easily involve an excessive number of terms, much of which may be redundant and need to be removed. Leontaritis and Billings (1987) solved this problem by using an optimal multiple selection method based on the theory of hypothesis testing. This basically involves the testing of all the possible subset models and is a very difficult and time-consuming task. Better suboptimal methods have to be employed to get a parsimonious model which uses only a small fraction of all the candidate terms in P but gives still a satisfactory representation of the actual process dynamics (Korenberg, 1985; Korenberg et al., 1988; Chen et al., 1989).

Table 1. Comprehensive Model Reduction Algorithm

First Stage:

1. Set $k=0$.
2. Copy all j columns of P, (p_j), to $\bar{p}_j^k$; ($\forall\ j=k, \ldots, M$).
3. Copy the observation vector z to $\bar{z}^k$.
4. Calculate $\gamma_k^j=\langle \bar{p}_j^k \bar{z}^k\rangle / \langle \bar{p}_j^k \bar{p}_j^k\rangle$; ($\forall\ j=k, \ldots, M$).
5. Compute $[\mathrm{err}]_k^j=(\gamma_k^j)^2\langle \bar{p}_j^k \bar{p}_j^k\rangle / \langle \bar{z}\bar{z}\rangle$; ($\forall\ j=k, \ldots, M$).
6. Obtain ℓ from: $[\mathrm{err}]_k^\ell=\max\{[\mathrm{err}]_k^j,\ k\le j\le M\}$
7. Interchange the ℓth column of $\bar{P}^k$ with the kth column of $\bar{P}^k$.
8. Copy the kth column of $\bar{P}^k$, $(\bar{p}_k^k)$, over to ω_k.
9. Compute $\alpha_{k,j}=\langle \omega_k \bar{p}_j^k\rangle / (\omega_k \omega_k\rangle$; ($\forall\ j=k+1, \ldots, M$).
10. Compute $\bar{p}_j^{k+1}=\bar{p}_j^k-\alpha_{k,j}\omega_k$; ($\forall\ j=k+1, \ldots, M$).
11. Compute $g_k=\langle \omega_k \bar{z}^k\rangle / \langle \omega_k \omega_k\rangle$.
12. Compute $\bar{z}^{k+1}=\bar{z}^k-g_k\omega_k$.
13. Compute $\hat{\Xi}=z-\Sigma_{j=0}^k g_j\omega_j$.
14. Compute $C(\hat{\Theta}_s)=(1/N+1)\langle \hat{\Xi}\hat{\Xi}\rangle$.
15. Compute $\mathrm{AIC}(4)_k=(N+1)\log C(\hat{\Theta}_s)+4(k+1)$, and $\mathrm{BIC}=(N+1)\log C(\hat{\Theta}_s)+(k+1)\log N$.
16. Set $k=k+1$, GOTO 2nd stage.

Second Stage:

1. Repeat steps 4 to 10 from first stage.
2. Interchange for ℓth column elements of A, $\alpha_{v,\ell}$, with the kth column elements, $\alpha_{v,k}$; ($\forall\ v=0, \ldots, k$).
3. Repeat steps 11 to 15 from 1st stage.
4. If $\mathrm{AIC}(4)_k \ge \mathrm{AIC}(4)_{k-1}$ or $\mathrm{BIC}_k \ge \mathrm{BIC}_{k-1}$ then GOTO 3rd stage; otherwise GOTO 2nd stage and set $k=k+1$.

Third Stage:

1. Set $M_s=k$.
2. Calculate $\hat{\theta}_{M_s}=g_{M_s}$.
3. Calculate $\hat{\theta}_j=g_j-\Sigma_{v=j+1}^{M_s}\alpha_{j,v}\hat{\theta}_v$; ($\forall\ j=M_s-1, \ldots, 0$).

For each of the m-output variables, the problem can be stated as: Find a subset $P_s^{(i)}$ of P and the corresponding parameter estimates $\hat{\Theta}_s^{(i)}$ which minimize $\|z^{(i)}-P_s^{(i)}\hat{\Theta}_s^{(i)}\|$. $\|\cdot\|$ is used to denote the Euclidian norm. Although the regressor matrix P is the same for every output, the subsets $P_s^{(i)}$ do not have to be equivalent.

Modified Gram-Schmidt orthogonalization

The modified Gram-Schmidt (MGS) orthogonalization algorithm for the selection of the parsimonious subset $P_s^{(i)}$ and the corresponding parameter estimate vector $\hat{\Theta}_s^{(i)}$ is presented. This compact algorithm is a summary of the work by Chen et al. (1989), with extensions to comply with a multivariable environment and for the extended NARX model. MGS has been chosen due to its numerical superiority to other orthogonalization procedures which are very sensitive to round off errors and possible ill-conditioning. However, MGS has the disadvantage of keeping the complete regressor matrix P ($N+1\times M+1$) stored over the whole orthogonalization procedure (Chen et al., 1989).

The steps in Table 1 have to be repeated for each ith ($i=1, \ldots, m$) output. For clarity, the superscript $^{(i)}$ (that is, $z^{(i)}$, $\theta^{(i)}$) has been dropped. For each ith output, the algorithm starts by copying the original regressor matrix P over to $\bar{P}$, and the corresponding dependent variable vector $z^{(i)}$ over to $\bar{z}$.

In the algorithm in Table 1 $\langle\cdot\rangle$ denotes the inner product. Since NARX models are used, each of the m subsystems of

Eq. 6 can be treated as decoupled from the others. Furthermore, it allows the use of a more complex performance criterion, that is, a combination of Akaike's information criterion (AIC) and Bayesian information criterion (BIC). Both criteria take into account the performance and complexity of the model (Akaike, 1970; Kashyap, 1977). The reason for using two different criteria is that the AIC criterion has been criticized because of its consistent overestimation of the true parameter vector. A dissection of the algorithm follows:

Stage 1:

- Steps 1 to 3 initialize the procedure as mentioned above and set the counter for the number of selected terms k equal to 0.
- Steps 4 to 6 select column ℓ of the regressor matrix $\overline{P}^k$ which reduces the sum of the squares of the residuals maximally at the kth iteration.
- Step 8 copies the selected column into the matrix $W = [\omega_0, \ldots, \omega_M]$.
- Steps 9 and 10 orthogonalize the remaining columns of $\overline{P}^k$ and build an updated version $\overline{P}^{k+1}$. Step 9 calculates the elements of an upper triangular matrix A:

$$A = \begin{bmatrix} 1 & \alpha_{0,1} & \alpha_{0,2} & \cdots & \alpha_{0,M} \\ & 1 & \alpha_{1,2} & \cdots & \alpha_{1,M} \\ & & \ddots & \ddots & \vdots \\ & \mathbf{0} & & 1 & \alpha_{M-1,M} \\ & & & & 1 \end{bmatrix}_{M+1 \times M+1} \tag{10}$$

This is corresponding to the Cholesky decomposition of the regressor matrix: $P = WA$.

- Steps 11 to 14 calculate the variance of the residuals, $C(\hat{\Theta}_s)$.
- Step 15 calculates the values of the AIC(4) and BIC criteria.

Stage 2:

- Step 2 has to be added to the procedure given in stage 1 to interchange the columns in A.
- Step 4 checks if a minimum of AIC(4) or BIC is obtained for this particular number of terms selected.

Stage 3:

- After the minimum of AIC(4) or BIC is obtained, $M_s^{(i)} + 1$ terms have been selected for this particular output y_i. The selected terms are identified by the different values for ℓ at each kth iteration and stored in the pointer vector $\ell^{(i)} = [\ell_{k=0}^{(i)}, \ldots, \ell_{k=M_s^{(i)}}^{(i)}]^T$. The elements of $\ell^{(i)}$ are pointers to the selected terms (columns) of P, the original regressor matrix.
- Steps 2 and 3 show how to calculate the corresponding parameter vector, $\hat{\Theta}_s^{(i)} = [\hat{\theta}_0^{(i)}, \ldots, \hat{\theta}_{M_s^{(i)}}^{(i)}]^T$, in a recursive manner.

A straightforward check for avoiding possible ill-conditioning which is not shown in the algorithm is also incorporated. If $\langle \overline{p}_j^k \overline{p}_j^k \rangle$ is less than a predetermined threshold (that is, 10^{-5}), $\overline{p}_j^k$ will not be considered as a candidate for ω_k (Billings et al., 1989).

The algorithm shown above gives the detailed structure of combining the MGS orthogonalization procedure with AIC and BIC for the proposed extended multivariable NARX models. With regard to having a bias term in the reduced model set, one can force the constant term, equivalent to column $\overline{p}_0$, into the model. This can be incorporated into the algorithm by explicitly setting $\ell_{k=0} = 0$ in step 6 of the first stage.

This concludes the model selection part, which is based on an off-line identification from an initial batch of data. The vital information gained in this part concerning the real system is: a very accurate system representation in the form of a nonlinear discrete-time input-output model exists. The original regressor matrix P can now be deleted, except the last row which keeps the most recent regressors. Each ith ($\forall\, i = 1, \ldots, m$) model is represented by a subset of a large number of possible candidate terms and is accessible with the following two vectors:

1. The pointer vector, $\ell^{(i)} = [\ell_{k=0}^{(i)}, \ldots, \ell_{k=M_s^{(i)}}^{(i)}]^T$, gives the terms of P which were selected (number of the columns). From this information, the subset $P_s^{(i)}$ can be obtained.
2. The corresponding estimated parameter vector $\hat{\Theta}_s^{(i)} = [\hat{\theta}_0^{(i)}, \ldots, \hat{\theta}_{M_s^{(i)}}^{(i)}]^T$.

To account for future time-variations in the process, the parameter estimates are being updated immediately after the initial off-line identification is finished and closed-loop control starts. The model structure itself will not be updated.

Controller Design

The proposed controller design is based on the essential feature of MPC. This is the employment of an explicit model to predict the effect of future control actions on the outputs and minimizes a specified objective function (Patwardhan et al., 1990, 1992; Li and Biegler, 1988; Schmidt and Biegler, 1990). The model used to predict the future outputs will be the extended NARX model obtained from the identification procedure given above.

MPC does not guarantee stability *per se*, this can be guaranteed only by using a suitable choice of parameters in the objective function. Conditions on these parameters which ensure stability have been obtained by Garcia and Morari (1982, 1985a,b) and Garcia et al. (1989). Since the choice for these parameters in the approach given follows the same rules as for linear MPC and general predictive control (GPC) and is treated extensively in literature (Marchetti et al., 1983; Mohtadi and Clarke, 1986; Clarke et al., 1987a,b; Morari and Zafiriou, 1989; McIntosh et al., 1991; Mayne and Michalska, 1991; Clarke, 1991), it will be discussed only briefly. The main focus in this section will be the on-line utilization of the obtained extended MIMO/NARX model in an adaptive nonlinear predictive controller design.

Recursive identification

To track time variations of the process, a recursive identification algorithm with exponential forgetting is implemented. This makes the control design adaptive. Exponential forgetting works well only if the process is properly excited all the time. If no proper excitation exists for a longer period, the estimator will "forget" the proper values of the parameters, and the uncertainties will grow. This estimator windup may then cause a burst in the output of the process. To avoid this, a constant trace algorithm for the recursive identification has been implemented (Åström and Wittenmark, 1989). This algorithm does not update the covariance matrix and the parameter vector, $\hat{\Theta}_s^{(i)}$, for that particular output, if the difference of measured and predicted output is sufficiently small (that is, within the magnitude of the measurement noise). The m covariance matrices, $Q^{(i)}$, are initialized to a prespecified value according

to $Q^{(i)} = \rho I$. I is the identity matrix ($M_s^{(i)} + 1 \times M_s^{(i)} + 1$) and ρ a positive value. If one of the diagonal elements of $Q^{(i)}$ ($\forall\, i = 1, \ldots, m$) becomes less than $\rho/100$ during the closed-loop operation, this element gets reset to its original value ρ.

Control law

In conventional MPC, one is concerned with the solution of an unconstrained NLP problem every discrete time step. This NLP problem is given by the multistep, quadratic cost function, J:

$$\min_{\mathbf{u}(t),\mathbf{u}(t+1),\ldots,\mathbf{u}(t+N_c-1)} J = \left\{ \sum_{j=\tau_D}^{N_p} (\boldsymbol{e}_j^T \boldsymbol{\Gamma}_e \boldsymbol{e}_j) + \sum_{j=0}^{N_c-1} [\Delta \boldsymbol{u}^T(t+j) \boldsymbol{\Gamma}_u \Delta \boldsymbol{u}(t+j)] \right\}. \quad (11)$$

$\boldsymbol{\Gamma}_e$ and $\boldsymbol{\Gamma}_u$ are weighting matrices (unit matrices multiplied by a positive value), where the latter one is referred to as a move suppression factor. $\boldsymbol{e}_j$ is the future error vector at time $(t+j)$, for j greater then the dead time τ_D. The future error vector is calculated by subtracting the set point, y_{SP}, from the model predictions, $\hat{y}$:

$$\boldsymbol{e}_j = \begin{bmatrix} e_{1_j} \\ \vdots \\ e_{m_j} \end{bmatrix} = \begin{bmatrix} \hat{y}_1(t+j|t) \\ \vdots \\ \hat{y}_m(t+j|t) \end{bmatrix} - \begin{bmatrix} y_{SP_1}(t+j) \\ \vdots \\ y_{SP_m}(t+j) \end{bmatrix}, \quad \forall\, j = \tau_D, \ldots, N_p. \quad (12)$$

N_p is the prediction horizon and N_c the control horizon which is defined as $\boldsymbol{u}(t+N_c+j) = \boldsymbol{u}(t+N_c)$, ($\forall\, j = 1, \ldots, N_p - N_c$). The difference in the control moves, $\Delta \boldsymbol{u}(t)$, is given by:

$$\Delta \boldsymbol{u}(t) = \begin{bmatrix} u_1(t) \\ \vdots \\ u_r(t) \end{bmatrix} - \begin{bmatrix} u_1(t-1) \\ \vdots \\ u_r(t-1) \end{bmatrix}. \quad (13)$$

To obtain the N_p predictions based on the measurements up to time t, $\hat{y}(t+j|t)$, ($\forall\, j = 1, \ldots, N_p$), the updated parameter estimates, $\hat{\boldsymbol{\theta}}_s^{(i)}(t)$, are used. The last row of the regressor matrix $\boldsymbol{P}$ (most recent candidate terms) are copied to $\boldsymbol{P}'$. The m pointer vectors, $\ell^{(i)}$, select from $\boldsymbol{P}'$ the elements chosen by the initial identification and the subsets $\boldsymbol{P}_s^{(i)}$ are created to calculate the multistep predictions:

$$\hat{y}_i(t+j|t) = [\hat{\boldsymbol{\theta}}_s^{(i)}(t)]^T \boldsymbol{P}_s^{(i)} \quad \forall\, j = 1, \ldots, N_p \text{ and } i = 1, \ldots, m. \quad (14)$$

$\boldsymbol{P}'$ gets recursively updated after every prediction step by using the previous predictions, the constant disturbance terms, $\boldsymbol{d}(t+j) = \boldsymbol{d}(t)$, ($\forall\, j = 1, \ldots, N_p$), the manipulated variable vectors and all their possible combinations, as shown in Eq. 6. After the NLP problem is solved for the current time, the regressor vector $\boldsymbol{P}$ is getting updated with the new measurements (outputs and disturbances), the applied control and their combinations also according to Eq. 6.

The advantage of using the pointer vectors, $\ell^{(i)}$, becomes clear now. For solving the unconstrained NLP problem of Eq. 11, multiple function evaluations have to be performed for each time step t, independent of the method employed. Using the copied version of the regressor vector $\boldsymbol{P}'$ and the pointer vectors $\ell^{(i)}$, a real-time search for minimizing J can be performed without altering the actual regressor $\boldsymbol{P}$ and without the need to have the model structure explicitly available.

Solution to the NLP problem

The solution of Eq. 11 requires an unconstrained optimization method. However, constraints usually exist on the output and manipulated variables of the process. These constraints were imposed by adding large penalties to the cost function for a violation. Unconstrained methods for solving NLP problems include direct and indirect methods with first or second order and secant methods which approximate the Hessian matrix, $\boldsymbol{H}$, by means of function values and gradient values (Edgar and Himmelblau, 1988; Bryson and Ho, 1975; Hestenes, 1980; Fletcher, 1980). The gradient vector, ∇J, and the Hessian matrix, $\boldsymbol{H}$, are defined by:

$$\nabla J(t) = \left[\frac{\partial J(t)}{\partial u_1(t)}, \ldots, \frac{\partial J(t)}{\partial u_r(t)}, \frac{\partial J(t)}{\partial u_1(t+1)}, \ldots, \frac{\partial J(t)}{\partial u_r(t+N_c)} \right]^T \quad (15)$$

and

$$\boldsymbol{H}(t) = \nabla^2 J(t) = \begin{bmatrix} \dfrac{\partial^2 J}{\partial u_1^2(t)} & \cdots & \dfrac{\partial^2 J}{\partial u_1(t) \partial u_r(t+N_c)} \\ \vdots & & \vdots \\ \dfrac{\partial^2 J}{\partial u_r(t+N_c) \partial u_1(t)} & \cdots & \dfrac{\partial^2 J}{\partial u_r^2(t+N_c)} \end{bmatrix}. \quad (16)$$

Whichever method is chosen, one has to make sure that the Hessian matrix or its approximation, $\hat{\boldsymbol{H}}$ (for secant methods), stays always positive-definite. If positive definiteness is not guaranteed at some point, the search may become unbounded. Its solution is suggested by Marquardt (1963). He suggested to add a constant positive value, β, to the diagonal elements of $\boldsymbol{H}$ or $\hat{\boldsymbol{H}}$ which is just sufficiently large to ensure positive definiteness, but does not overwhelm $\boldsymbol{H}$ or $\hat{\boldsymbol{H}}$ itself.

Since concern is with the real-time application of the proposed algorithm in the absence of any analytical model, a fast numerical solution is sought. Probably one of the most successful updating relations is the secant-based Broyden, Fletcher, Goldfard and Shanno (BFGS) procedure (Fletcher, 1970; Edgar and Himmelblau, 1988). The BFGS procedure ensures that $\hat{\boldsymbol{H}}$ remains positive-definite if the approximation of the Hessian at the starting point is positive-definite. Since the starting point for the search is arbitrarily chosen to be the most recently applied control, [$u^{k=0}(t+j) = u(t-1)$, $\forall\, j = 0, \ldots, N_c$], no guarantee can be given regarding positive definiteness and Marquardt's or a related method has to be applied. The same is valid for any successive search step ($k > 0$). As mentioned

Table 2. Modified Marquardt Algorithm

1. Pick $u^0(t+j)=u(t-1)$, $(\forall\, j=0, \ldots, N_c)$, as the starting point. Pick a value η for the convergence criterion.
2. Set $k=0$. Let $\beta^0=10^3$.
3. Calculate $J_{(u^k)}\cdot u^k=[(u^k(t))^T, \ldots, (u^k(t+N_c))^T]^T$
4. Calculate $\nabla J_{(u^k)}$ numerically by central difference method.
5. Is $\|\nabla J_{(u^k)}\|<\eta$? If yes, terminate. If no, continue.
6. Calculate $s^k=-[H^k+\beta^k I]^{-1}\nabla J_{(u^k)}$. H^k is obtained numerically.
7. Calculate $J_{(u^k+s^k)}$.
8. Calculate $\hat{\lambda}=-(\nabla^T J_{(u^k)}s^k)/(2[J_{(u^k+s^k)}-J_{(u^k)}-\nabla^T J_{(u^k)}s^k])$ and check bounds on $\hat{\lambda}$.
9. Update $u^{k+1}=u^k+\hat{\lambda}s^k$.
10. Is $J_{(u^{k+1})}<J_{(u^k)}$? If yes, go to step 11. If no, go to step 12.
11. Set $\beta^{k+1}=(1/4)\beta^k$ and $k=k+1$. Go to step 3.
12. Set $\beta^{k+1}=2\beta^k$. Go to step 6.

before, a positive value β has to be found for each search step which alters the Hessian matrix. Therefore, updating the Hessian according to the BFGS or any related secant procedure is not feasible for the approach under consideration.

A less powerful algorithm, but with the assurance of positive definiteness of H at each search step, k, is the modified Marquardt algorithm (Edgar and Himmelblau, 1988), where the elements of ∇J and $H=\nabla^2 J$ are obtained numerically by central difference methods. A listing of the modified Marquardt algorithm adapted to the current problem is given in Table 2.

Step 6 of the algorithm in Table 2 would assume a full Newton step, $\hat{\lambda}=1$, which is not often the best choice in minimizing an (objective) function in regions farther away from the minimum. Several methods of backtracking or trust regions can be used to obtain the right value for $\hat{\lambda}$. A quadratic interpolation of the objective function is given in step 8 to obtain an approximate value for the optimum step length, $\hat{\lambda}$, which should be kept in prespecified bounds: $-100.0\le\hat{\lambda}\le-0.1$ and $0.1\le\hat{\lambda}\le 100.0$. This backtracking method has the advantage of Newton's method of fast convergence even in regions farther away from the minimum as long as the objective function can be approximated reasonably well by a quadratic function.

A flow chart of this proposed algorithm for solving the NLP problem is given in Figure 1. After convergence of the NLP procedure (termination of the modified Marquardt algorithm), only the manipulated variable obtained for the current time, $u(t)$, is implemented. An offset-free response is achieved by the adaptive nature of the model which is getting updated every time a plant-model mismatch occurs.

Tuning parameters

The closed-loop performance of the system depends on the design parameters for the recursive identification scheme and the tuning parameters for the MPC. Having a large number of design or tuning parameters adds flexibility to the tuning procedure, but it can also cause problems due to "too many degrees of freedom" and proper tuning can become difficult. The number of tuning parameters should be kept low. Some general guidelines for selecting the proper tuning parameters are given below (Garcia and Morari, 1982; Garcia et al., 1989; McIntosh et al., 1991).

The two design parameters for the recursive identification scheme are the initial settings of the covariance matrices (one for each output channel) and the forgetting factor. Due to the fact that very accurate initial parameter estimates from the MGS procedure are available, the covariance matrices can be kept small. A forgetting factor of $\lambda<1$ is a common choice in order to track time variations.

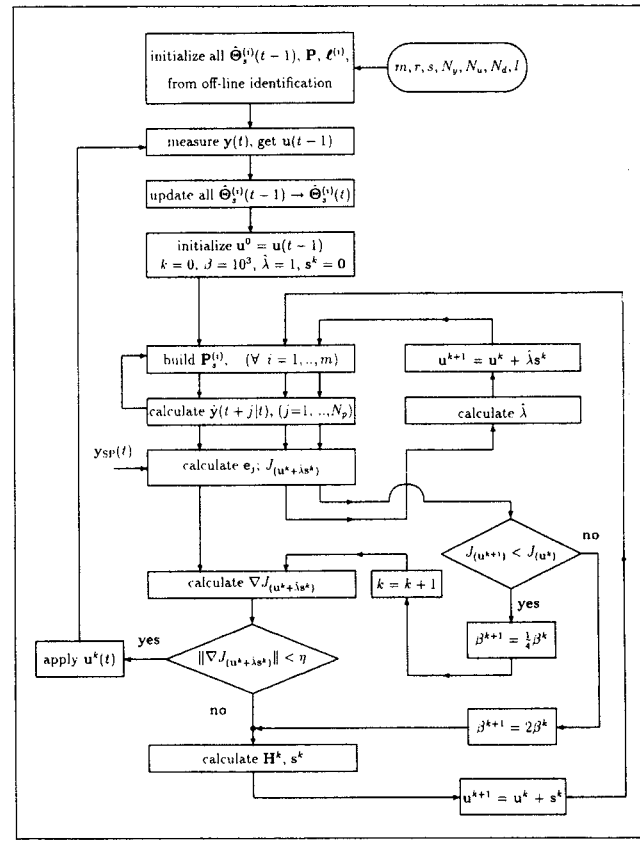

Figure 1. Flow chart for solving the NLP problem.

The tuning parameters for the MPC part are:

- Control horizon N_c
- Prediction horizon N_p
- Prediction weighting Γ_e
- Control weighting Γ_u.

The settings of one of the parameters usually influences the other parameters strongly. Therefore, one has to be very careful in the choice of the parameters with regard to a desired closed-loop performance, required CPU time for solving the NLP problem, and stability issues. The choice of the parameters are case-specific, and general rules are difficult to present without becoming controversial.

Wastewater Neutralization Experiments

The proposed identification/adaptive control methodology has been applied to a single-input single-output (SISO), measured disturbance, pH neutralization process. In this section, the process and the process model based on first principles are discussed, as well as the identification and control results for the computer simulations and for the real-time experiments.

Wastewater pH neutralization process

The Environmental Protection Agency (EPA) states in its report "Toxics in the Community, 1988," that acids-bases-

salts is the largest class of discharged wastes with 2.2 billion lb (1 billion kg) (Sarokin, 1990). An ever-growing demand on pH neutralization processes is being placed due to the concern for protecting aquatic life and human welfare, preventing materials from corrosion, and preliminary treatments of waste streams for further biological processing or recycling.

Control of pH is industrial wastewater streams is still a challenging task for the control engineer (Hoffmann, 1972; Shinskey, 1973; Trevathan, 1978; Brown, 1983a,b; Gustafsson, 1985; Pajunen, 1987; Hall and Seborg, 1989; Wright and Kravaris, 1991; Kalafatis et al., 1992; Henson and Seborg, 1992). The problems caused by the static nonlinearity of the pH (the required amount of reagent increases by a factor of 10 for each unit change in pH) are often overestimated in control applications. The more serious problem is caused by sudden concentration changes of buffer agents. These concentration changes are impossible to measure on-line and cannot be detected by a pH measurement. Such concentration changes shift the titration curve and, therefore, change the nonlinear characteristics of the process.

Gustafsson (1985) used a laboratory-scale pH neutralization system to investigate a reaction invariant and adaptive linear pH feedback control design. The same process setup and model equations have been used to verify the proposed identification/adaptive control methodology by computer simulations. A similar pH neutralization system as used by Gustafsson (1985) has been built and used for experimental validation on a laboratory scale, as shown in Figure 2.

The pH neutralization system is based on a fictitious chemical process waste stream, by adding various amounts of $NaHCO_3$ and NaOH to a constant stream of water. Sodium bicarbonate in the feed acts as a buffer and influences the slope of the titration curve greatly (Hamilton et al., 1969). Sulfuric acid is used to neutralize the feed stream in a continuous stirred-tank reactor (CSTR). To fit this neutralization process in the extended NARX model frame, the following definitions have been used (with $m=r=s=1$):

- The measured pH in the feed stream, pH_f is being referred to as a known disturbance.
- The measured pH in the tank, pH_t is the controlled variable.
- The added H_2SO_4 is being referred to as the manipulated variable. For the computer simulations, the manipulated variable was referred to as the flow of hydrogen ions in the control stream divided by the total flow through the reactor. During the real-time laboratory experiments, the manipulated variable has been the control stream flow rate, $\dot{V}_{H_2SO_4}$, delivered by pump **P3**, as indicated in Figure 2.

The fact that the process has a varying dead time due to the changes of the feed and the control stream flow rates has not been considered in the model equations but has been present in the real-time experiments.

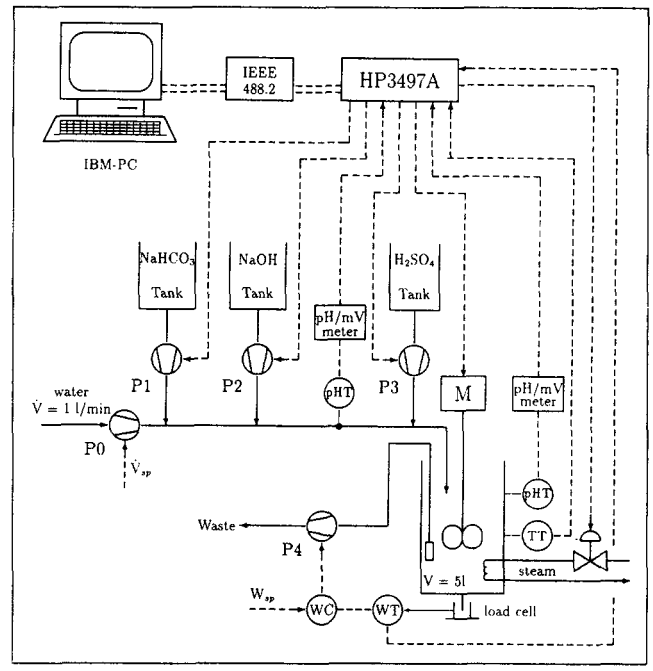

Figure 2. Lab-scale pH neutralization process.

IEEE 488.2: 400 kbytes/s read/write data acquisition interface bus, MS-DOS supported; **HP3497A**: Hewlett Packard data acquisition/control unit with the options: 010 (20 channel relay multiplexer); 120 (dual output 0 to ± 10 V D/A converter); 3 × 130 (dual output 4-20 mA D/A converter). **P0**: Peristaltic pump for main water flow (external set point). **P1** and **P2**: Peristaltic pumps for feed disturbances (remote controlled 4-20 mA). **P3**: Peristaltic pump for control stream (remote controlled 4-20 mA). **P4**: Peristaltic withdrawal pump (remote controlled 0-10 V). **pHT**: pH electrodes, calomel filled, sealed. **pH/mV**: pH meter with recorder output, accuracy 0.01pH. **M**: remote controlled (0-10 V) motor for the mixer. **TT**: thermocouple.

Process model

The model for simulating this neutralization process has been obtained by using the theory of reaction invariants and variants by Gustafsson and Waller (1983). The chemical reactions considered are:

$$H_2O \rightleftharpoons H^+ + OH^- \tag{17}$$

$$H_2CO_3 \rightleftharpoons H^+ + HCO_3^- \tag{18}$$

$$HCO_3^- \rightleftharpoons H^+ + CO_3^{2-} \tag{19}$$

with the reaction invariant variables as:

$$\omega_1 = c_H - c_{OH} - c_{HCO_3} - 2c_{H_2CO_3} \tag{20}$$

$$\omega_2 = c_{H_2CO_3} + c_{HCO_3} + c_{CO_3}. \tag{21}$$

$w=[w_1, w_2]^T$ is called the reaction invariant vector and does not depend on the extent of the reactions in Eqs. 17 to 19. We assume chemical equilibrium and instantaneous reactions. The equilibrium constants of the reactions (Eqs. 17 to 19) defined as follows have a value of (at 25°C):

$$K_w = [H^+][OH^-] = 10^{-14} \quad [mol^2/L^2] \tag{22}$$

$$K_{a_1} = \frac{[H^+][HCO_3^-]}{[H_2CO_3]} = 10^{-6.3} \quad [mol/L] \tag{23}$$

$$K_{a_2} = \frac{[H^+][CO_3^{2-}]}{[HCO_3^-]} = 10^{-10.1} \quad [mol/L] \tag{24}$$

The reactor used by Gustafsson (1985) has the following model equation (for a sampling rate of $\Delta t = 15$ s):

$$w(t+1) = 1.6989w(t) - 0.7040w(t-1) + 0.0027[w_f(t-3) + w_c(t-3)] + 0.0024[w_f(t-4) + w_c(t-4)] \quad (25)$$

where $w_f = [w_{f_1}, w_{f_2}]^T$ corresponds to the concentrations of the reaction invariants in the feed stream. $w_c = [u', 0]^T$, since the control stream contains only a strong acid. The relationship between the actual pH in the process, $pH' = -\log[H^+]$, and the reaction invariants, w_1 and w_2, is easily obtained as:

$$w_1 = 10^{-pH'} - 10^{pH'-14} - w_2 \cdot \frac{1 + 2 \cdot 10^{pH'-10.1}}{1 + 10^{6.3-pH'} + 10^{pH'-10.1}}. \quad (26)$$

A simple first-order process has been used to model the dynamics of the two pH electrodes (one in the feed stream, pH_f and the other in the tank, pH_t):

$$pH(t) = 0.6065pH(t-1) + 0.3935pH'(t). \quad (27)$$

Additionally, white noise with zero mean and a standard deviation of 0.03 pH unit has been superimposed on both of the pH measurements.

Identification Experiments. During the initial identification experiment, it is very important that the process data sufficiently cover the range of future operation regions. For the pH process under investigation, the data should include a fair range of the expected feed compositions and hence implicitly the different titration curves. In addition to this concept of persistent excitation, the data set has to be large enough to estimate the high number of model parameters.

Although the closed-loop control will employ an adaptive updating of the model parameters, the structure obtained from the identification procedure will remain unchanged. Therefore, a model structure obtained from a rich data set will have better generalization properties than a model obtained from a data set with very poor excitation. For the given process, a data set with $N = 500$ data points (equivalent to a little more than 2 h of open-loop operation) has been chosen which should be sufficient for models with usually much less than 30 terms. In obtaining the data set, special attention has been given to change the buffer concentration in the feed several times from low to high—the main influence on the titration curve and the gain of the process.

The changes in the feed composition and the changes on the manipulated variable were based on pseudo-random with random amplitude signals. The probability for a specific signal to change after every discrete time step has been set to 0.3 and the maximum amplitudes during identification have been:

$$\left.\begin{aligned} 0.0 \le c_{CO_3,f} \le 0.005 \quad & [\text{mol/L}] \\ 0.001 \le c_{OH,f} \le 0.01 \quad & [\text{mol/L}] \\ 0.0 \le u' \le 0.015 \quad & [\text{mol/L}] \end{aligned}\right\} \quad (28)$$

The maximum amplitudes for the two feed compositions ($c_{CO_3,f}$, $c_{OH,f}$) were chosen to cover approximately the anticipated future compositions of the fictitious waste stream. The total carbonate concentration level in the tank as the main influencing factor of the tank titration curve is obtained from the reaction invariant w_2, see Eq. 21. It varies during the identification experiment between 0.0 (at $t = 0$) and 0.0031 mol/L with a mean of 0.0018 and a standard deviation of 0.0007, which is assumed to be sufficient for the anticipated operation range.

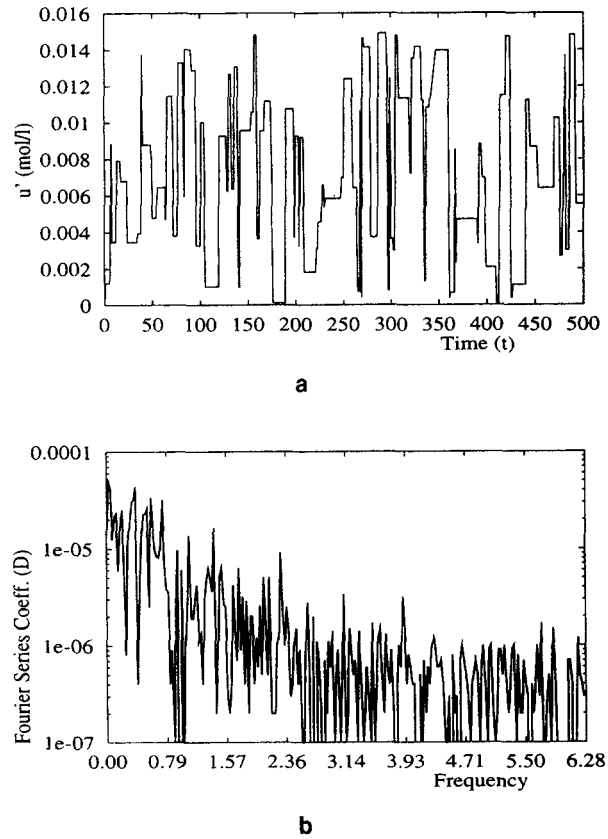

Figure 3. a. Time series of manipulated variable u' for the excitation of the process during identification; b. Fourier series frequency spectrum of time series in Figure 3a.

Figure 3a shows the actual signal used during the identification for the manipulated variable u', and Figure 3b shows its discrete Fourier series frequency spectrum. The spectrum reveals that a satisfactory frequency distribution is indeed present, especially in the lower-frequency domain which is of greater importance for the process under investigation.

After obtaining an acceptable data set, the signals have to be normalized before they can be used in the MGS orthogonalization procedure given in Table 1:

$$\left.\begin{aligned} d &= (pH_f - \mu_{pH_f})/\sigma_{pH_f} \\ y &= (pH_t - \mu_{pH_t})/\sigma_{pH_t} \\ u &= (u' - \mu_{u'})/\sigma_{u'} \end{aligned}\right\} \quad (29)$$

where μ and σ are the mean and standard deviation, respectively. The regression matrix, $\boldsymbol{P}$, for the MGS orthogonalization has been built by these 500 normalized dependent/independent data pairs using maximum lags of $N_y = N_u = N_d = 5$, and varying maximum polynomial degrees, l. Results of the

Table 3. Results of MGS Orthogonalization for Simulation Experiments

l	M	M_s	V	Terms Selected (Available Through Pointer Vector)
1	15	6	4.2E-2	1.0, $y(t-1)$, $y(t-2)$, $y(t-4)$, $u(t-1)$, $u(t-4)$, $d(t-3)$
2	135	9	3.8E-4	1.0, $y(t-1)$, $y(t-2)$, $y(t-4)$, $u(t-1)$, $u(t-4)$, $d(t-3)$, $y(t-5)u(t-4)$, $u(t-4)d(t-3)$, $d^2(t-4)$
3	815	12	3.5E-4	1.0, $y(t-1)$, $y(t-2)$, $u(t-1)$, $u(t-4)$, $d^2(t-3)$, $y^2(t-1)u(t-1)$, $y^2(t-1)u(t-4)$, $y^2(t-2)u(t-4)$, $u(t-2)u^2(t-4)$, $u(t-4)u^2(t-5)$, $u(t-4)d^2(t-2)$, $d^2(t-3)d(t-4)$

MGS for three different maximum polynomial degrees, ($l=1$, 2, 3), are shown in Table 3. M gives the total number of possible terms to choose from, M_s is the actual number of terms chosen, and V is the sum of the errors squared, $[y(t)-\hat{y}(t)]^2$.

Figure 4 shows the actual process output and the model outputs for $l=1$ and $l=2$. The linear model shows satisfactory fitting except in the very low and very high pH regions. Both nonlinear models (only $l=2$ is shown) give a very satisfactory fit throughout the whole pH range.

For the closed-loop operation of over 1,000 discrete time steps, the same disturbance pattern as used by Gustafsson (1985) has been applied.

Figure 5 shows the two extreme cases of feed compositions during closed-loop control by means of their titration curves. One can see the nonlinear variations the models will have to represent and hence the drastic changes in the gain for the controller which need to be determined.

Closed-Loop Control Experiments. For the numerical evaluation of the gradient and Hessian, a step size of 0.01 normalized u units (Eq. 29) has been used, and the termination parameter for the modified Marquardt method was $\eta=10^{-4}$. For the recursive identification, the forgetting factor was chosen to be $\lambda=0.95$ and the covariance matrix was initialized by

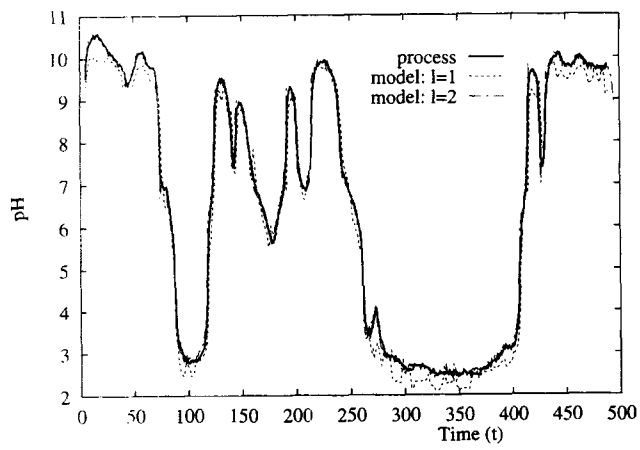

Figure 4. Identification results using different polynomial degree initializations.

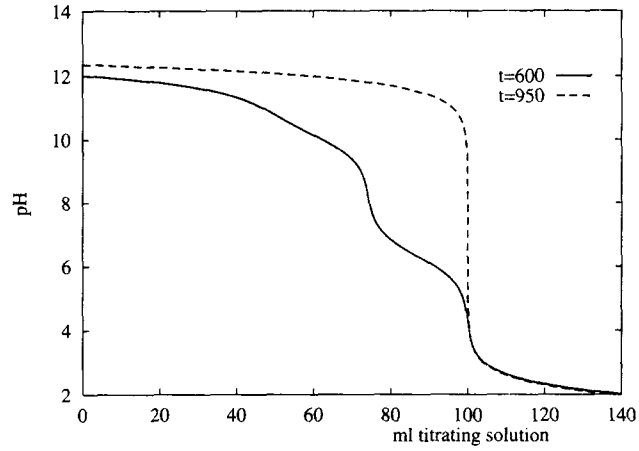

Figure 5. Theoretical titration curves for two extreme disturbance cases during closed-loop operation.

$Q=\rho I$, with $\rho=100$. The control and prediction horizons were chosen to be $N_c=1$ and $N_p=6$, and the process time delay was set to $\tau_D=2$. Several tuning runs were used to obtain information about the settings for Γ_e and Γ_u. A satisfactory choice was found for $\Gamma_e=10$ and $\Gamma_u=1$. First, the performance of the MPC is presented when a linear model with all possible terms (15 and a bias term) and unknown parameter estimates [$\hat{\theta}_j(t=0)=0$, $\forall\, j=0, \ldots, 15$] are used. This would be equivalent of not using an *a priori* identification for the structure determination and evaluation of the initial parameter estimates. The closed-loop output variable and the measured disturbances are shown in Figure 6a. Figures 6b and 6c show the controlled variable for the cases where $l=1$ and $l=2$ of Table 3, respectively. Figure 6d shows the manipulated variable for the simulation experiment of Figure 6c.

Table 4 shows that between $t=50$ and $t=200$ and above $t=700$ very steep titration curves are present in the feed. The control utilizing the uninitialized model in Figure 6a is especially unsatisfactory in those regions. Oscillations, as often described for pH feedback systems, occurred (Shinskey, 1973; Gustafsson and Waller, 1982). MPC based on the linear model as shown in Figure 6b gives a more satisfactory control, although oscillations occurred for $t>850$. Use of the NARX model ($l=2$, Table 3) gives a very satisfactory control performance. The NARX model wth $l=3$, however, did not show additional significant improvements. A comparison of these results with those by Gustafsson (1985) suggest improvements in the overall control performance. Further simulation experiments using the model of $l=2$ show that changes on the prediction horizon, N_p, have strong influence on the closed-loop performance. $N_p<3$ gives very poor results, whereas $N_p\geq 4$ gives improving and satisfactory results (for $\Gamma_e=10$ and $\Gamma_u=1$). Prediction horizons chosen too large would result in a too sluggish controller. For a prediction horizon of $N_p=6$, the controller shows reasonable fast performance and is very robust with respect to change in Γ_e and Γ_u. Keeping Γ_e at a value of 10, the movement suppression factor Γ_u could be increased to 20 and still show oscillations no larger than ± 0.5 pH units from the set point of $pH_t=7$.

To investigate how the adaptive MPC controller performs

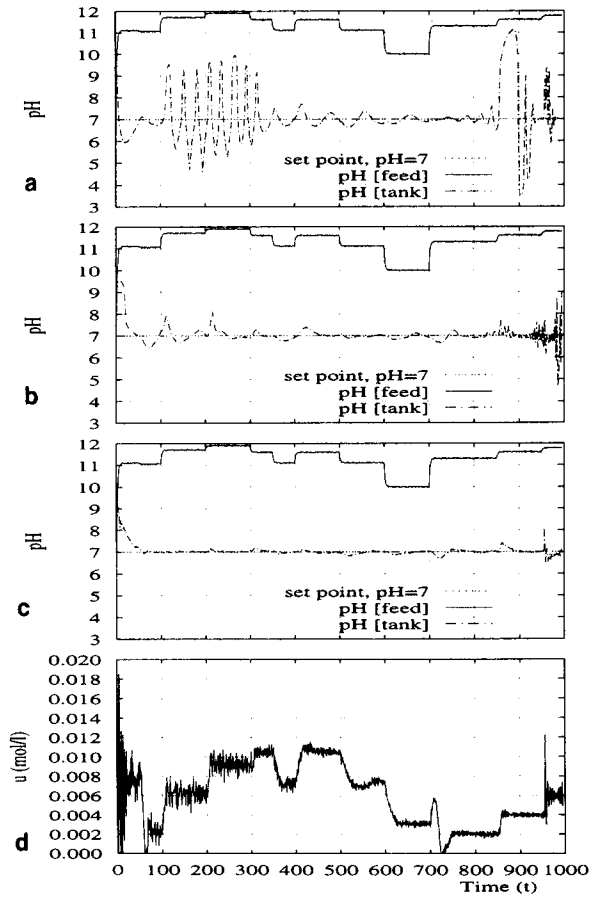

Figure 6. Output variable $y = pH_t$ and disturbance $d = pH_f$ for closed-loop control results of computer simulation experiments.

a. Linear model, all possible terms for $N_y = N_u = N_d = 5$ considered. Initial parameter estimates $\hat{\theta} = 0$. b. Linear model, obtained from MGS with $l = 1$. c. Nonlinear model, obtained from MGS with $l = 2$. d. Manipulated variable for Figure 6c.

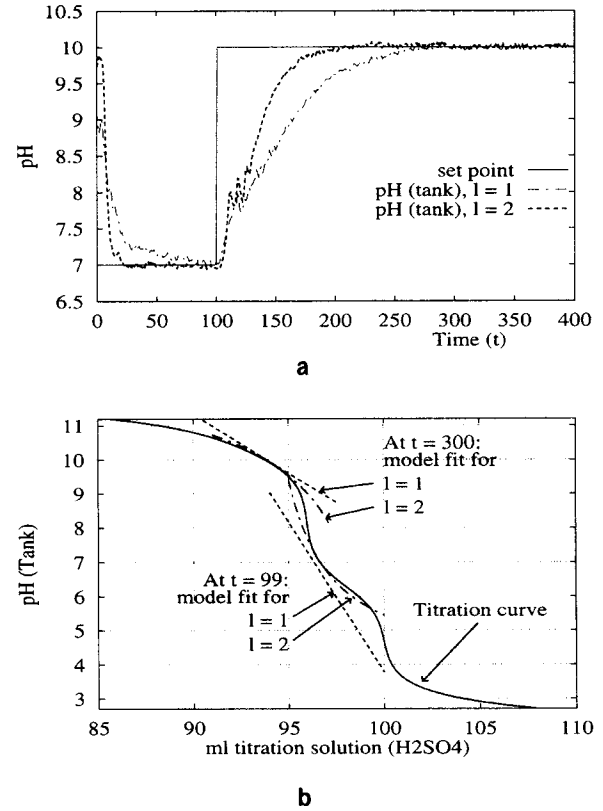

Figure 7. Set point change for output variable y from 7 to 10 pH at $t = 100$; b. Comparison of the model accuracy for $t = 99$ and $t = 300$ for the set point change given in Figure 7a.

for changes in the set point the following experiment has been conducted: the feed composition was kept constant at $c_{CO_3,f} = 0.001$ and $c_{OH,f} = 0.002$ mol/L, but the set point was changed from $pH_t = 7$ to $pH_t = 10$ at $t = 100$. In Figure 7a, the controlled variables for using the models $l = 1$ and $l = 2$ of Table 3 are shown. Both controllers perform satisfactorily with slight disturbances around $pH_t = 8$ where the titration curve shows an inflection point. Figure 7b compares in a snapshot the models at $t = 99$ ($pH_t = 7$) and $t = 300$ ($pH_t = 10$) to the actual process. The advantage of using nonlinear models is obvious by their better approximation of the titration curve and, therefore, the process dynamics. The model data in Figure 7b are obtained by knowing the model structure (Table 3), the model parameters at the specified time, and by calculation of the stationary point (Hernández and Arkun, 1993). The linear model at $t = 99$ was of the following form:

$$\left.\begin{aligned} y(t) &= 0.0150 + 1.940y(t-1) - 0.944y(t-2) \\ &+ 0.171y(t-4) + 0.039u(t-1) - 0.004u(t-4) + 0.060d(t-3). \end{aligned}\right\} \tag{30}$$

The stationary point can be calculated by setting $y(t) = y(t-1) = \cdots = y(t-N_y) = y^*$ and doing the same for the disturbance, d^*, and the manipulated variable, u^*. Knowing the disturbance value ($d^* = -0.109$ normalized units), Eq. 30 gives the linear relationship $y^* = -0.0508 - 0.2096u^*$, as shown in Figure 7b for $t = 99$ (in normalized units). The nonlinear model ($l = 2$) gave the nonlinear function $y^* = (0.001138 - 0.0075u^*)/(0.045 + 0.0104u^*)$. The process operation point, using the actually applied value for the manipulated variable, was matched closely by the models, except for the linear model at $t = 99$.

Table 4. Feed Concentrations for Closed-Loop Control (mol/L)

t	$c_{CO_3,f}$	$c_{OH,f}$	t	$c_{CO_3,f}$	$c_{OH,f}$
0	0.006	0.006	400	0.006	0.009
50	0.001	0.002	500	0.006	0.006
100	0.001	0.006	600	0.006	0.002
200	0.001	0.009	700	0.0	0.002
300	0.006	0.009	850	0.0	0.004
350	0.006	0.006	950	0.0	0.006

Laboratory experiments

The pH neutralization process is given in Figure 2. The concentrations for the disturbance tanks were chosen according

to the computer simulations: 0.085 mol/L and $NaHCO_3$, 0.25 mol/L NaOH. The control tank concentration was 1.25 mol/L H_2SO_4. The continuous stirred-tank reactor (CSTR) has a working volume of 5 L which was kept constant during the experiments by a local controller, supervised from the computer-data acquisition unit. A user-friendly graphical interface has been developed in C on an IBM-PC 80386, 25 MHz, with a 80387 math compressor and 8 Mbytes of RAM. Using the internal clock of the computer, the sampling/control interval was set to 15 s, equivalent to the model for the computer simulations. However, due to the delay caused by solving the NLP problem (time span from sampling to implementing the control) variations in the interval length occurred. For the control based on the models obtained from $l=1$ and $l=2$, the delays were within a second. For $l=3$, the delays increased up to 6 s which was unacceptable and no results were obtained for this case. The problem can be solved by the implementation of a faster computer CPU, better memory management, and a simpler graphical interface. The memory requirement for the program (including graphics, data acquisition interface commands, identification and control part) is roughly 800 kbytes, encoded in approximately 4,000 lines. However, during execution, the full 8 Mbytes of RAM is required.

Identification Experiments. Identical pseudorandom sequences with random amplitudes were used to excite the system for the same period ($N=500$) as for the computer simulation experiments. The three signals for $c_{CO_3,f}$, $c_{OH,f}$, and u', now driving the pumps ***P1***, ***P2*** and ***P3***, respectively, had to be scaled to provide the required feed concentrations. The scaling was based on simple calibrations of the pump feed rates and the known signal range for the pumps of 4 to 20 mA. The collected data for pH_f and pH_t have been normalized as in Eq. 29, the control signal has been normalized between 0 and 1 as follows:

$$u=[\dot{V}_{H_2SO_4}-(\dot{V}_{H_2SO_4})_{min}]/[(\dot{V}_{H_2SO_4})_{max}-(\dot{V}_{H_2SO_4})_{min}]. \quad (31)$$

Table 5 gives the results for the MGS orthogonalization procedure, similar to Table 3 but using the data obtained from the laboratory experiments:

All three models ($l=1$, 2, 3) gave very satisfactory representation of the actual process data. However, the terms for the model, $l=3$ are not entered since it could not be implemented in closed-loop control due to the aforementioned CPU problems.

Closed-loop Control Experiments. The initial settings of $\eta=10^{-4}$, $\lambda=0.95$, $\rho=100$, $N_c=1$, $N_p=6$ and $\tau_D=2$ were identical to the computer simulations. Figure 8a shows the laboratory control performance utilizing the linear model ($l=1$, Table 5) with the weighting parameters $\Gamma_e=10$ and $\Gamma_u=1$. Figures 8b and 9c show the control based on the model with a maximum monomial degree of $l=2$ and the settings $\Gamma_e=10$, $\Gamma_u=1$ and $\Gamma_e=10$, $\Gamma_u=0.1$, respectively. Figure 8d gives the manipulated variable for the experiment in Figure 8c. The control performance based on the linear model shows satisfactory performance during the first part of the small feed buffer concentrations ($t=50$ to $t=300$), but tends to become unsatisfactory and oscillatory after $t=700$. The pH_t drops to 3 between $t=900$ and $t=950$, although a stable satisfactory control was regained after $t=960$. Figures 8b and 8c indicate

Table 5. Results of MGS Orthogonalization for Laboratory Experiments

l	M	M_s	V	Terms Selected (Available Through Pointer Vector)
1	15	9	8.7E-3	1.0, $y(t-1)$, $y(t-2)$, $y(t-3)$, $u(t-1)$, $u(t-3)$, $u(t-4)$, $d(t-1)$, $d(t-3)$, $d(t-4)$
2	135	10	5.3E-3	1.0, $y(t-1)$, $y(t-2)$, $u(t-1)$, $d(t-1)$, $y(t-1)u(t-4)$, $y(t-2)u(t-4)$, $y(t-3)u(t-4)$, $y(t-4)u(t-4)$, $u(t-1)u(t-3)$, $d^2(t-1)$

satisfactory control utilizing the nonlinear model ($l=2$). The experimental run of Figure 8c with a smaller move suppression factor, ($\Gamma_u=0.1$), gave better results than using the higher value for Γ_u, ($\Gamma_u=1.0$ in Figure 8b). The control became clearly faster due to less weight on the control moves Δu.

At various times, samples were withdrawn from the feed stream after the two inlets for sodium hydroxide and sodium bicarbonate, but before the pH_f, flows through the cell. These samples (50 mL) were titrated with a 0.025 mol/L H_2SO_4. The

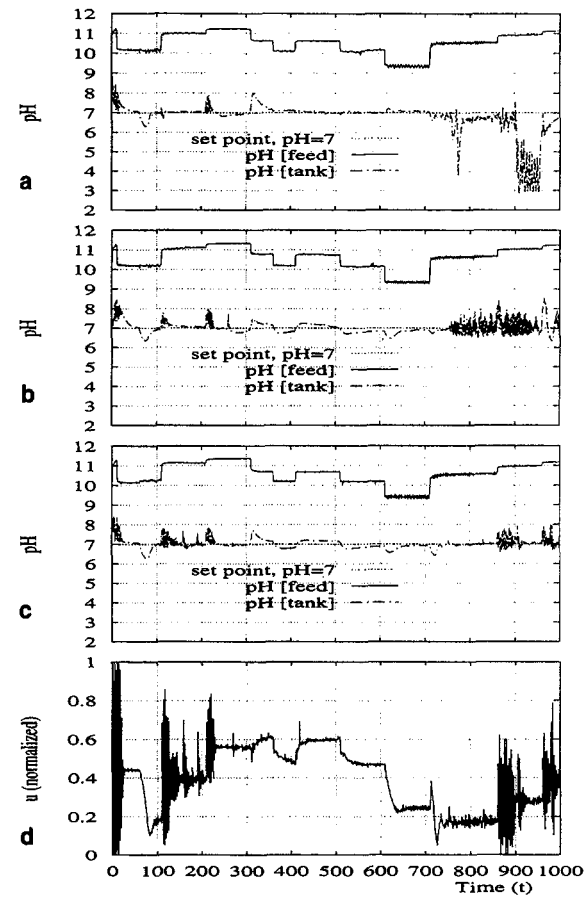

Figure 8. Output variable $y=pH_t$ and disturbance $d=pH_f$ for the closed-loop control results of laboratory experiments.

a. Linear model, $\Gamma_e=10$, $\Gamma_u=1$; b. nonlinear model ($l=2$), $\Gamma_e=10$, $\Gamma_u=1$; c. nonlinear model ($l=2$), $\Gamma_e=10$, $\Gamma=0.1$; d. manipulated variable for Figure 8c.

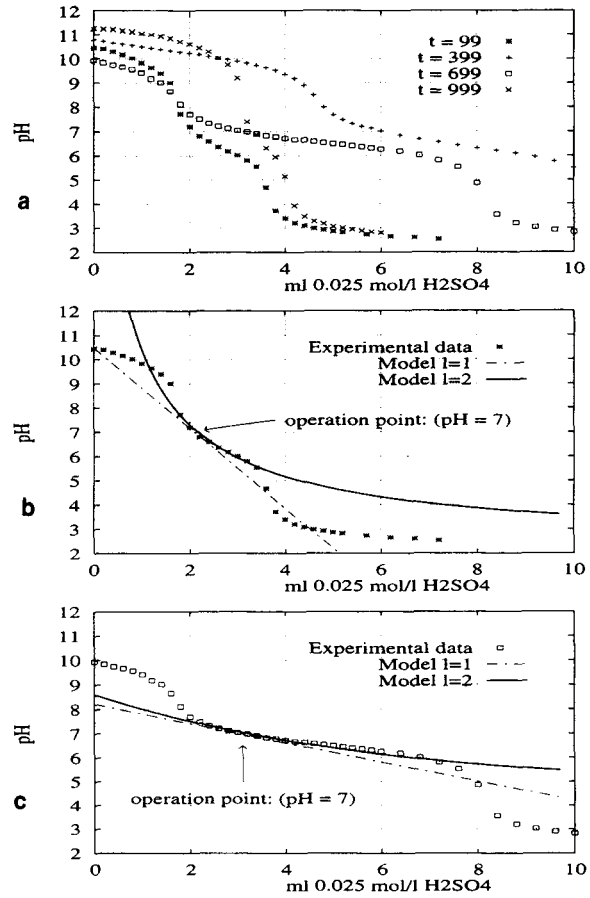

Figure 9. a. Titration curves from feed stream at time $t = 99$, 399, 699 and 999 from the experimental run in Figure 8c; b. linear and nonlinear model fits for $t = 99$; c. linear and nonlinear model fits for $t = 699$.

results for $t = 99$, $t = 399$, $t = 699$ and $t = 999$ are shown in Figure 9a. The samples for $t = 99$ and $t = 999$ have a low concentration of buffer ($NaHCO_3$) present and, therefore, the steep titration curve. Because municipal water is always slightly carbon-buffered, the titration of the sample taken at $t = 999$ does not show the almost vertical curve as in Figure 5. Only pure H_2O would give the theoretically vertical function. Samples withdrawn at $t = 399$ and $t = 699$ had an equivalent high amount of buffer agent, but different amounts of NaOH which cause the horizontal shift in the titration curve.

For two samples, $t = 99$ and $t = 699$, the same analysis of the model validity in Figure 7b is performed using the concept of the stationary points. The results for $t = 99$ and $t = 699$ are shown in Figures 9b and 9c, respectively. Clearly, a better fit of the process dynamics is shown by the nonlinear model, especially for the steeper titration curve at $t = 99$. This agrees with the actual performance of the nonlinear-model-based controller which shows advantages primarily in the regions of low buffer concentrations.

Conclusions

In this work, the objective was to develop a real-time identification/adaptive model predictive control methology based on NARX models. This was accomplished by extending the existing theory of MIMO NARX models by incorporating a measured disturbance term into the model structure and by means of utilization of a model-predictive control environment. Explicit algorithms for the identification and the MPC are given to implement the proposed methodology, which utilizes a newly defined pointer vector. The issue of solving the nonlinear programming problem which arises in the model-predictive control part has been addressed and investigated.

The proposed methodology has been verified on a wastewater pH neutralization process, using computer simulations and experiments conducted on a laboratory-scale process. The issues of good excitation during the identification procedure and how satisfactory the extended NARX model adapts to time-varying process dynamics are demonstrated. In both cases (computer simulations and laboratory experiments), very satisfactory performance has been achieved. The results suggest that control performance improvements are possible by using nonlinear ARX models instead of linear ones with relatively easy implementation.

Acknowledgment

The authors wish to acknowledge Colorado State University Experimental Station and the Colorado Commission for Higher Education (Excellence Award in Water Resource Education) for partially funding this research project.

Notation

A = matrix from Cholesky decomposition
b = element of bias vector
$\boldsymbol{b}$ = bias vector
c = concentration
$C(\,)$ = variance of residuals
$\boldsymbol{d}$ = measured disturbance vector
D = Fourier series coefficient
$\boldsymbol{e}$ = error vector from predictions
$f(\,)$ = nonlinear function
$\boldsymbol{F}, \boldsymbol{F}$ = nonlinear vector functions
g = inner product ratio of orthogonalized vectors
$\boldsymbol{H}$ = Hessian matrix
$\boldsymbol{I}$ = identity matrix
J = performance index
k = step number
K = equilibrium constant
l = maximal monomial degree
ℓ = element of pointer vector
$\boldsymbol{\ell}$ = pointer vector
m = number of outputs
M = number of candidate terms
n = number of linear terms
N = number of dependent/independent pairs
N_0 = maximum lag
N_c = control horizon
N_p = prediction horizon
$\boldsymbol{p}, \bar{\boldsymbol{p}}$ = column vectors of regressor matrix
pH′ = actual pH
pH = measured pH
$\boldsymbol{P}'$ = updated regressor vector
$\boldsymbol{P}, \bar{\boldsymbol{P}}$ = regressor matrix (vector during control part)
$\boldsymbol{Q}$ = covariance matrix
r = number of inputs
s = number of disturbances
$\boldsymbol{s}$ = search direction
t = discrete time

u = input vector
V = sum of errors squared
V = volumetric flow rate
w = reaction invariant concentration
w = reaction invariant vector
W = matrix from Cholesky decomposition
$x_{(i)}$ = NARX model element
y = output vector
y_{SP} = set point vector
$z, \bar{z}$ = dependent variable

Greek letters

α = elements of A
β = Marquardt parameter
γ = inner product ratio
Γ = weighting matrix
Δ = discrete difference
∇ = gradient operator
ϵ = error vector of NARX model
η = termination threshold
θ = model parameter
Θ = parameter vector
λ = forgetting factor
$\hat{\lambda}$ = corrected step size for Marquardt algorithm
μ = mean
Ξ = modeling error vector
ρ = covariance initialization value
σ = standard deviation
τ_D = time delay
ω = column vectors of W

Superscripts and subscripts

$*$ = stationary point
$\hat{}$ = estimated value
c = control
d = disturbance
f = feed
(i) = ith output
r = set point
s = submodel
t = tank
T = transpose
u = control
w = water
y = output

Literature Cited

Akaike, H., "Statistical Predictor Identification," *Ann. Stat. Math.*, **22**, 203 (1970).

Åström, K. J., and B. Wittenmark, *Adaptive Control*, Addison-Wesley, New York (1989).

Billings, S. A., M. J. Korenberg, and S. Chen, "Identification of Non-linear Output-affine Systems Using an Orthogonal Least-squares Algorithm," *Int. J. Sys. Sci.*, **19**, 1559 (1988).

Billings, S. A., S. Chen, and M. J. Korenberg, "Identification of MIMO Non-linear Systems Using a Forward-regression Orthogonal Estimator," *Int. J. Control*, **49**(6), 2157 (1989).

Brown, M., "Lessons in pH Control of Industrial Effluent," *Process Eng.*, **64**, 53 (1983a).

Brown, M., "Tuning-in to the Dynamics of Effluent Neutralization," *Process Eng.*, **64**, 49 (1983b).

Bryson, A. E., and Y. C. Ho, *Applied Optimal Control: Optimization, Estimation, and Control*, Hemisphere, New York (1975).

Chen, S., and S. A. Billings, "Recursive Prediction Error Parameter Estimator for Non-linear Models," *Int. J. Control*, **49**(2), 569 (1989a).

Chen, S., and S. A. Billings, "Representation of Non-linear Systems: the NARMAX Model," *Int. J. Control*, **49**(3), 1013 (1989b).

Chen, S., S. A. Billings, and W. Luo, "Orthogonal Least Squares Methods and their Application to Non-linear System Identification," *Int. J. Control*, **50**(5), 1873 (1989).

Clarke, D. W., "Adaptive Generalized Predictive Control," *Proc. CPC-IV*, San Padre Island, TX (1991).

Clarke, D. W., C. Mohtadi, and P. S. Tuffs, "Generalized Predictive Control: I. The Basic Algorithm," *Automatica*, **23**, 137 (1987a).

Clarke, D. W., C. Mohtadi, and P. S. Tuffs, "Generalized Predictive Control: II. Extensions and Interpretations," *Automatica*, **23**, 149 (1987b).

Draper, N. R., and H. Smith, *Applied Regression Analysis*, Wiley, New York (1981).

Edgar, T. F., and D. M. Himmelblau, *Optimization of Chemical Processes*, McGraw-Hill, New York (1988).

Fletcher, R., "A New Approach to Variable Metric Algorithms," *Comput. J.*, **13**, 317 (1970).

Fletcher, R., *Practical Methods of Optimization*, Vol. 1, Wiley, New York (1980).

Garcia, C. E., and M. Morari, "Internal Model Control: 1. A Unifying Review and Some New Results," *I&EC Process Des. and Dev.*, **21**, 308 (1982).

Garcia, C. E., and M. Morari, "Internal Model Control: 2. Design Procedure for Multivariable Systems," *I&EC Process Des. and Dev.*, **24**, 472 (1985a).

Garcia, C. E., and M. Morari, "Internal Model Control: 3. Multivariable Control Law Computation and Tuning Guidelines," *I&EC Process Des. and Dev.*, **24**, 484 (1985b).

Garcia, C. E., D. M. Prett, and M. Morari, "Model Predictive Control: Theory and Practice—A Survey," *Automatica*, **25**(3), 335 (1989).

Gustafsson, T. K., "An Experimental Study of a Class of Algorithms for Adaptive pH Control," *Chem. Eng. Sci.*, **40**(5), 827 (1985).

Gustafsson, T. K., and K. V. Waller, "Oscillations in Feedback Systems for pH Control," Technical Report 82-5, Åbo Akademi, Dept. of Chemical Engineering, Åbo, Finland (1982).

Gustafsson, T. K., and K. V. Waller, "Dynamic Modeling and Reaction Invariant Control of pH," *Chem. Eng. Sci.*, **38**(3), 389 (1983).

Haber, R., and H. Unbehauen, "Structure Identification of Nonlinear Dynamic Systems—A Survey on Input/Output Approaches," *Automatica*, **26**, 651 (1990).

Hall, R. C., and D. E. Seborg, "Modelling and Self-tuning Control of a Multivariable pH Neutralization Process: I. Modelling and Multiloop Control," *Proc. of Amer. Contr. Conf.*, p. 1822, Pittsburgh (1989).

Hamilton, L. F., S. G. Simpson, and D. W. Ellis, *Calculations of Analytical Chemistry*, McGraw-Hill, New York (1969).

Henson, M. A., and D. E. Seborg, "Nonlinear Adaptive Control of a pH Neutralization Process," *Proc. Amer. Cont. Conf.*, p. 2586, Chicago (1992).

Hernández, E., and Y. Arkun, "Control of Nonlinear Systems Using Polynomial ARMA Models," *AIChE J.*, **39**(3), 446 (1993).

Hestenes, M., *Conjugate Direction Methods in Optimization*, Springer-Verlag, Berlin (1980).

Hoffmann, F., "How to Select a pH Control Systems for Neutralizing Waste Acids," *Chem. Engr.*, 105 (Oct. 30, 1972).

Kalafatis, A., L. Wang, and W. R. Cluett, "Identification and Robust Control of pH in a CSTR Using Step Response Data," *Proc. IFAC Symp.*, p. 297, College Park, MD (1992).

Kashyap, R., "Bayesian Comparison of Different Classes of Dynamic Models using Empirical Data," *IEEE Trans. Autom. Control*, **22**, 715 (1977).

Korenberg, M. J., "Orthogonal Identification of Nonlinear Difference Equation Models," *Proc. Symp. on Circuits and Systems*, Louisville, KY (1985).

Korenberg, M. J., S. A. Billings, Y. P. Liu, and P. J. McIlroy, "Orthogonal Parameter Estimation Algorithm for Non-linear Stochastic Systems," *Int. J. Control*, **48**(1), 193 (1988).

Kortmann, M., and M. Unbehauen, "Ein neuer Algorithmus zur automatischen Selektion der optimalen Modellstruktur bei der Identifikation nichtlinearer Systeme," *Automatisierungstechnik*, **35**, 481 (1987).

Leontaritis, I. J., and S. A. Billings, "Input-Output Parameter Models for Non-linear Systems: 1. Deterministic Non-linear Systems: 2. Stochastic Non-linear Systems," *Int. J. Control*, **41**, 303 (1985).

Leontaritis, I. J., and S. A. Billings, "Model Selection and Validation Methods for Non-linear Systems," *Int. J. Control*, **45**, 311 (1987).

Li, W. C., and L. T. Biegler, "Process Control Strategies for Constrained Nonlinear Systems," *Ind. Eng. Chem. Res.*, **27**, 1421 (1988).
Marchetti, J. L., D. A. Mellichamp, and D. E. Seborg, "Predictive Control Based on Discrete Convolution Models," *Ind. Eng. Chem. Process Des. Dev.*, **22**, 488 (1983).
Marquardt, D., "An Algorithm for Least-Squares Estimation on Nonlinear Parameters," *S.I.A.M. J. Appl. Math.*, **11**, 431 (1963).
Mayne, D. Q., and H. Michalska, "Model Predictive Control of Nonlinear Systems," *Proc. Amer. Contr. Conf.*, p. 2343, Boston (1991).
McIntosh, A. R., S. L. Shah, and D. G. Fisher, "Analysis and Tuning of Adaptive Generalized Predictive Control," *Can. J. of Chem. Eng.*, **69**, 97 (1991).
Mohtadi, C., and D. W. Clarke, "Generalized Predictive Control, LQ or Pole Placement: A Unified Approach," *Proc. Control and Decision Conf.*, p. 1536, Athens, Greece (1986).
Morari, M., and E. Zafiriou, *Robust Process Control*, Prentice Hall, Englewood Cliff, NJ (1989).
Morningred, J. E., B. E. Paden, D. E. Seborg, and D. A. Mellichamp, "An Adaptive Nonlinear Predictive Controller," *Chem. Eng. Sci.*, **47**(4), 755 (1992).
Pajunen, G. A., "Comparison of Linear and Non-linear Adaptive Control of a pH-Process," *IEEE Control Sys. Mag.*, **2**, 39 (1987).
Patwardhan, A. A., G. T. Wright, and T. F. Edgar, "Nonlinear Model-Predictive Control of Distributed-Parameter Systems," *Chem. Eng. Sci.*, **47**(4), 721 (1992).
Patwardhan, A. A, J. B. Rawlings, and T. F. Edgar, "Nonlinear Model Predictive Control," *Chem. Eng. Comm.*, **87**, 123 (1990).
Sarokin, D., "Toxics in the Community, 1988: National and Local Perspective," Technical Report, U.S. EPA (1990).
Schmid, C., and L. T. Biegler, "Application of Multistep Newton-type Controllers to Fluid Catalytic Cracking," *Proc. Amer. Control Conf.*, San Diego (1990).
Shinskey, F. G., *pH and pIon Control in Process and Waste Streams*, Wiley, New York (1973).
Trevathan, V. L., "Advanced Control of pH," *Proc. ISA Conf.*, p. 62, Philadelphia (1978).
Wright, R. A., and C. Kravaris, "Nonlinear Control of pH Processes Using Strong Acid Equivalent," *Ind. Eng. Chem. Res.*, **30**, 1561 (1991).

Manuscript received Dec. 21, 1992, and revision received Jun. 9, 1993.