AN ADAPTIVE NONLINEAR PREDICTIVE CONTROLLER*

J. DUANE MORNINGRED,† BRADLEY E. PADEN,‡ DALE E. SEBORG† and DUNCAN A. MELLICHAMP††

†Department of Chemical and Nuclear Engineering; ‡Department of Mechanical and Environmental Engineering, University of California, Santa Barbara, CA 93106, U.S.A.

(Received for publication 6 August 1991)

Abstract—The design and implementation of a new adaptive nonlinear predictive controller is presented using a general nonlinear model and variable transformations. The resulting controller is similar in form to standard linear model predictive controllers and can be tuned analogously. Alternatively, the controller can be tuned using a single parameter. The design is computationally efficient. The controller is updated on-line without recalculating the controller gain matrix, which involves a matrix inversion. The new controller is compared to a PI controller and to an adaptive linear predictive controller through simulations of a continuous stirred-tank reactor. The effects of modeling errors on the new controller are also shown with simulations.

1. INTRODUCTION

Computer process control, beginning in the 1960s, initially used traditional linear control algorithms. However, the highly nonlinear characteristics of many processes caused problems for these controllers. Consequently, in the 1970s, self-regulating controllers [1, 2] were developed to enhance explicit and implicit model-based controllers and controller tuning. The process industries, however, found few successes with the early, hard-to-tune adaptive techniques [3].

Meanwhile, moving-horizon and linear-programming methods were being revived in the nonadaptive, model-based predictive controllers (MPC), dynamic matrix control (DMC) [4], and model-predictive heuristic control [5]. Industrial successes with MPCs renewed academic interest in these methods and new formulations of MPCs emerged [6]. These multivariable controllers were based on easily understood process models, could incorporate constraints, and were relatively easy to tune. In addition, their performance seemed less sensitive to varying time delays, one of the major limitations of the early adaptive controllers. To improve the robustness of the adaptive controllers, some researchers began to employ extended-horizon strategies [7]. Likewise, predictive controllers were improved by incorporating adaptive techniques [8].

During the past several years, some researchers have started to focus on strategies that directly compensate for process nonlinearities in controller design [9, 10]. Naturally, these nonlinear control concepts have been introduced into adaptive-type controllers [11, 12] and into predictive-type controllers [13–16]. The nonlinear control problem has also been approached numerically using nonlinear optimization software packages [17, 18]. In this paper, an adaptive controller is formulated in a discrete-time nonlinear predictive control framework.

2. MODELS AND PARAMETER ESTIMATION

Many SISO processes can be accurately represented by the discrete, time-invariant model of eq. (1):

\[ y(t) = F[y(t-1), u(t-k-1), Z(t-1), M(t-1)] \]

where

\[ y(t-1) = [y(t-1), y(t-2), \ldots, y(t-n_y)]^T \]

\[ u(t-k-1) = [u(t-k-1), u(t-k-2), \ldots, u(t-k-n_u)]^T \]

\[ Z(t-1) = [z(t-1), z(t-2), \ldots, z(t-n_z)] \]

\[ M(t-1) = [m(t-1), m(t-2), \ldots, m(t-n_m)] \]

In this paper, the assumption is made that the nonlinear model, chosen with a priori process knowledge, represents the characteristics of the process over a wide range of operating conditions. If the model is linear with respect to the parameters, the process model in

*An earlier version of this paper was presented at the 1990 American Control Conference.
†Author to whom correspondence should be addressed.
eq. (1) then can be rearranged to give

$$y(t) = \theta^T(t - 1) \phi(t - 1) + \epsilon(t)$$

(6)

where $\theta(t - 1)$ is the parameter vector estimated at time $t - 1$, $\phi(t - 1)$ is the information vector at time $t - 1$, and $\epsilon(t)$ is the estimation error. With a NARMAX model, $\phi$ contains functions of past values of $y$, $u$, $z$, and $m$. RLS estimation of $\theta$ is used to adjust the model to local process conditions. This paper employs RLS with directional forgetting, as described by Kulhávý and Karný [20].

### 3. CONTROLLER DESIGN METHODS

Two types of adaptive nonlinear predictive controllers (ANPC) will be designed. The first type assumes a restriction on the NARMAX model and is tuned in a manner similar to that of linear MPCs. The second type removes this restriction and is tuned by selecting a parameter to place closed-loop poles.

#### 3.1. Design based on a NARMAX model with separable linear dynamics

Predictive controller designs that use a “dynamic matrix” [4] require a linear-dynamics model. Without linear dynamics, the subsequent controller design could not be tuned using the optimization and control horizon lengths. Consequently, the first ANPC design assumes that linear dynamics can be separated from the NARMAX model in eq. (1), as follows:

$$F[y(t - 1), u(t - k - 1), Z(t - 1), M(t - 1)] =$$

$$= q[1 - A(q^{-1})]y(t - 1) + G[y(t - 1),$$

$$u(t - k - 1), Z(t - 1), M(t - 1)]$$

(7)

where

$$A(q^{-1}) = 1 + a_1 q^{-1} + a_2 q^{-2} + \cdots + a_n q^{-n}$$

(8)

$$G[\cdot]$$ is a general, nonlinear function, and $q^{-1}$ is the backwards-shift operator. The poles of $A(q^{-1})$ must also be stable. The NARMAX model in eq. (7) is then expressed in a prediction form as

$$A(q^{-1}) \hat{y}(t) = G[\hat{y}(t - 1),$$

$$u(t - k - 1), Z(t - 1), M(t - 1)]$$

(10)

where the caret denotes a prediction. Equation (10) can be rearranged to give an $i$-step ahead prediction:

$$\hat{y}(t + i) = q^i[1 - A(q^{-1})]^{-1} \hat{y}(t)$$

$$+ q^i \sum_{j=0}^{i} [1 - A(q^{-1})] j G[\hat{y}(t - 1), u(t - k - 1),$$

$$Z(t - 1), M(t - 1)].$$

(11)

Next, following the linearization strategy of Hunt et al. [21], a new variable, $\nu(t - k - 1)$, is defined to represent the nonlinearity of eq. (10):

$$v(t - k - 1) \equiv G[\hat{y}(t - 1),$$

$$u(t - k - 1), Z(t - 1), M(t - 1)]$$

(12)

so that eq. (10) can be expressed as a linear model

$$A(q^{-1}) \hat{y}(t) = v(t - k - 1)$$

(13)

and eq. (11) can be expressed as

$$\hat{y}(t + i) = q^i[1 - A(q^{-1})]^{i+1} \hat{y}(t)$$

$$+ q^i \sum_{j=0}^{i} [1 - A(q^{-1})] j v(t - k - 1).$$

(14)

The use of eq. (12) provides an exact transformation of the nonlinear model to a linear model. Thus, eq. (14) can be rewritten to represent the nonlinear prediction in the matrix form:

$$\hat{y}(t) = \hat{y}P(t) + A \Delta v(t)$$

(15)

where $\hat{y}(t)$ is an $R$-vector of predicted outputs; and $\hat{y}P(t)$ is an $R$-vector of the effect of past input changes on the $R$ future outputs:

$$\hat{y}(t) = [\hat{y}(t + 1), \hat{y}(t + 2), \ldots, \hat{y}(t + R)]^T$$

(16)

$$\hat{y}P(t) = [\hat{y}P(t + 1), \hat{y}P(t + 2), \ldots, \hat{y}P(t + R)]^T.$$  

(17)

In eq. (15) $\Delta v(t)$ is an $L$-vector of the form

$$\Delta v(t) = \Delta v(t), \Delta v(t + 1), \Delta v(t + 2), \ldots,$$

$$\Delta v(t + L - 1)]^T$$

(18)

and $A$ is an $R \times L$ “dynamic matrix”:

$$A = \begin{bmatrix}
    a_1 & 0 & \cdots & 0 \\
    a_2 & a_1 & \cdots & 0 \\
    \vdots & \vdots & \ddots & \vdots \\
    a_L & a_{L-1} & \cdots & a_1 \\
    a_R & a_{R-1} & \cdots & a_{R-L+1}
\end{bmatrix}.$$  

(19)

In eq. (15) $\Delta$ is the backward difference operator. The optimization horizon $R$ is the number of future process moves over which $L$ future control moves are optimized. $L$ is called the control horizon. The following relationships for $a_j$ and $\hat{y}P(t + i)$ can be derived:

$$\hat{y}P(t + i) = q^i[1 - A(q^{-1})]^{i+1} \hat{y}(t)$$

$$+ \sum_{j=0}^{i} h_j v(t - 1) + \sum_{j=i+1}^{i+N+k+1} h_j v(t + i - j)$$

(20)

$$a_j = \sum_{n=j}^{i} h_n$$

(21)

where $h_n = 0$ for $0 \geq n \geq k$ and $h_n$ is the coefficient of the $q^{-n}$ term of $[q^{-k} A(q^{-1})]^{-1}$ for $n > k$. Consequently, $h_n$ is merely the $n$th impulse–response coefficient of eq. (13) for a unit impulse in $v(t)$, and $a_j$ is the $j$th step–response coefficient for a unit step in $v(t)$.

A predictive controller can be designed by minimizing the objective function

$$J = [r(t) - \hat{y}(t)]^T Q^T Q [r(t) - \hat{y}(t)]$$

$$+ f \Delta v(t)^T \Delta v(t)$$

(22)
where \( r(t) \) is the \( R \)-vector of future set-points, \( Q \) is a weighting matrix, and \( f \) is the move suppression factor. Substituting eq. (15) into eq. (22) and minimizing with respect to \( \Delta v(t) \) yields

\[
\Delta v(t) = K_c E(t) \quad (23)
\]

\[
K_c = (\alpha^T Q^T Q \alpha + f I)^{-1} \alpha^T Q^T Q \quad (24)
\]

where \( K_c \) is the constant controller gain matrix and \( E(t) \) is the "no-future-change-in-\( u \)" set-point error vector,

\[
E(t) = r(t) - \hat{y}^p(t). \quad (25)
\]

In analogy with linear MPCs, only the first element of \( \Delta v(t) \) needs to be calculated. Using \( \Delta v(t), v(t - 1), \) and eq. (12), the controller output \( u(t) \) can be calculated. In addition, the tuning parameters for the "control law" in eq. (23), \( R, L, f, \) and \( Q \) are analogous to those of linear MPCs.

### 3.2. Design based on a general NARMAX model

For the second type of controller, the restriction of eq. (7), that linear dynamic terms can be separated from \( F'[\cdot] \), is removed. Removing this restriction corresponds to setting \( A(q^{-1}) \) of eq. (13) to 1 when such a polynomial cannot be separated. In such a case, the prediction model of eq. (14) reduces to

\[
y(t + i) = v(t - k - 1 + i). \quad (26)
\]

For this model, each \( a_j = 1 \), for \( j > k \). In effect, the "dynamic" matrix \( \alpha \) of eq. (18) reduces to a "static" matrix. In the linear analogy, all of the model poles and zeros are cancelled. This analogy provides insight as to why \( R \) and \( L \) are not useful tuning parameters in this case.

In the design of a linear pole-placement controller, all poles are cancelled except for the ones that are placed to specify the speed and shape of the closed-loop response. A pole-placement ANPC will be designed in a similar manner. Regardless of whether or not linear dynamics can be separated from the general NARMAX model eq. (1), a polynomial, \( [1 - \hat{A}(q^{-1})]y(t), \) can be added to and subtracted from the right-hand side to yield

\[
y(t) = q[1 - \hat{A}(q^{-1})]y(t - 1) - q[1 - \bar{A}(q^{-1})]y(t - 1) + F[y(t - 1), u(t - k - 1), Z(t - 1), M(t - 1)]. \quad (27)
\]

Combining eq. (27) with the following definition of \( G[y(t - 1), u(t - k - 1), Z(t - 1), M(t - 1)] \equiv F[y(t - 1), u(t - k - 1), Z(t - 1), M(t - 1)] - q[1 - \bar{A}(q^{-1})]y(t - 1) \quad (28)\)

yields the process model,

\[
y(t) = q[1 - \hat{A}(q^{-1})]y(t - 1) + G[y(t - 1), u(t - k - 1), Z(t - 1), M(t - 1)] \quad (29)
\]

which is identical in form to eq. (7). As shown in eqs (7)-(25), a predictive controller can be designed using this process model. The roots of \( \hat{A}(q^{-1}) \) can be specified to shape the closed-loop response. For example, choosing \( \hat{A}(q^{-1}) = 1 - pq^{-1}, \) where \( 0 > p > -1, \) and combining with the variable transformation of eq. (12), gives the following prediction model:

\[
\hat{y}(t + i) = q'[pq^{-1}]^{i+1}\hat{y}(t) + q' \sum_{j=0}^{i} [pq^{-1}]^{i}v(t - k - 1) \quad (30)
\]

from which the controller is designed. The values of \( h_n \) for this controller design are simply

\[
h_n = 0 \quad \text{for} \quad 0 \geq n \geq k \quad (31)
\]

Thus, little computational effort is needed to calculate \( h_n \).

However, the major computational advantage is not in the calculation of \( h_n \). Note that the controller gain matrix is now a function of the chosen \( \bar{A}(q^{-1}) \), and is no longer a function of the output model \( F'[\cdot] \). Thus, the controller gain matrix does not need to be recalculated when parameters of the output model are updated on-line. This approach avoids having to do a matrix inversion each time the controller is re-designed.

### 3.3. Controller tuning

For the first ANPC controller design, the tuning methods are analogous to those of linear MPCs \([4, 22]\). The second controller design seemingly adds another tuning parameter that must be selected. In practice, \( R \) is often chosen to correspond to the open-loop settling time of a process response to a step change in the manipulated input \([4]\). For a linear first-order process, the 99% response is about 5 times the dominant time constant plus the time delay of the process. Because the tuning pole, \( p \), ideally determines the settling time of the process, \( R \) can be chosen as the following function of \( p \):

\[
R = \frac{-5}{\log_e(p)} + k_{\text{max}} \quad (32)
\]

where \( k_{\text{max}} \) is the maximum expected process time delay. Thus, the choice for \( p \) specifies \( R \). As the effects of \( p \) on the closed-loop response are easily understood, the more difficult to select \( f \) and \( Q \) can be set to their limits of zero and the identity matrix, respectively. The result is that only \( p \) and \( L \) need to be selected. Consequently, the objective function in eq. (22) can be restated as

\[
J' = [r(t) - \hat{y}(t)]^T [r(t) - \hat{y}(t)] \quad (33)
\]

and the minimization solution for \( \Delta v(t) \) as

\[
\Delta v(t) = (\alpha^T \alpha)^{-1} \alpha^T E(t). \quad (34)
\]

Several researchers have reported \([18, 22]\) that the performance of a predictive controller designed with
\[ L > 1 \text{ is often not significantly better than one designed with } L = 1. \text{ Choosing } L = 1, \text{ eq. (34) can be rewritten in terms of } p \text{ as} \]

\[ \Delta v(t) = \sum_{i=k+1}^{R} \sum_{j=0}^{R-k-1} \frac{p^{n-k-1}}{p^{2j}} E(t + i) \quad (35) \]

where \( E(t + i) \) is the \( i \)th element of \( E(t) \).

### 4. CSTR SIMULATION EXAMPLE

The simulated continuous stirred-tank reactor (CSTR) process consists of an irreversible, exothermic reaction, \( A \rightarrow B \), in a constant volume reactor cooled by a single coolant stream which can be modeled by the following equations:

\[ \dot{C}_A(t + d) = \frac{q(t)}{V} [C_{A0}(t) - C_A(t + d)] - k_0 C_A(t + d) \exp \left[ \frac{-E}{RT(t)} \right] \quad (36) \]

\[ \dot{T}(t) = \frac{q(t)}{V} [T_0(t) - T(t)] - \frac{(-\Delta H)k_0 C_A(t + d)}{\rho C_p} \times \exp \left[ \frac{-E}{RT(t)} \right] + \frac{\rho C_p}{\rho C_p V} q_c(t) \times \left\{ 1 - \exp \left[ \frac{-hA}{q_c(t) \rho C_p} \right] \right\} [T_0(t) - T(t)]. \quad (37) \]

The measured concentration has a time delay \( d = 0.5 \text{ min.} \) The objective is to control the measured concentration of \( A, C_A \), by manipulating the coolant flow rate \( q_c \). This model is a modified version of the first tank of a two-tank CSTR example by Henson and Seborg [23]. In the original model, the time delay was zero. The nominal parameter values appear in Table 1. The highly nonlinear characteristics of the CSTR can be seen in Fig. 1, which shows the open-loop concentration responses to step changes in the coolant flow rate. The coolant flow rate was changed from an initial value of 100 \text{ l min}^{-1} \text{ to 110, to 100, to 90, and back to 100, at 8 min intervals.}

**Table 1. Nominal CSTR parameter values**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measured product concentration</td>
<td>( C_A )</td>
</tr>
<tr>
<td>Reactor temperature</td>
<td>( T )</td>
</tr>
<tr>
<td>Coolant flow rate</td>
<td>( q_c )</td>
</tr>
<tr>
<td>Process flow rate</td>
<td>( q )</td>
</tr>
<tr>
<td>Feed concentration</td>
<td>( C_{A0} )</td>
</tr>
<tr>
<td>Feed temperature</td>
<td>( T_0 )</td>
</tr>
<tr>
<td>Inlet coolant temperature</td>
<td>( T_{co} )</td>
</tr>
<tr>
<td>CSTR volume</td>
<td>( V )</td>
</tr>
<tr>
<td>Heat transfer term</td>
<td>( hA )</td>
</tr>
<tr>
<td>Reaction rate constant</td>
<td>( k_0 )</td>
</tr>
<tr>
<td>Activation energy term</td>
<td>( E/R )</td>
</tr>
<tr>
<td>Heat of reaction</td>
<td>( \Delta H )</td>
</tr>
<tr>
<td>Liquid densities</td>
<td>( \rho_l, \rho_i )</td>
</tr>
<tr>
<td>Specific heats</td>
<td>( C_p, C_{pc} )</td>
</tr>
</tbody>
</table>

**Fig. 1. Open-loop composition response.**
4.1. The controller designs

The sampling period of all process measurements was assumed to be 0.1 min. This sampling period allowed the faster dynamics of the higher purity region to be sampled about 4 times per dominant time constant. The time delay of 0.5 min \((k = 5)\) was assumed known. The feed flow rate \(q\), the feed temperature \(T_0\), the coolant inlet temperature \(T_{io}\), and the reactor temperature \(T_r\) were assumed to be accurately measured. The inlet feed concentration \(C_{Ao}\) was unmeasured and assumed to be constant at its nominal value. All controllers were designed with a rate-of-change constraint of 2 \(1 \text{ min}^{-1}\) per sampling period on the controller output.

For the adaptive controllers, the first 6 min of each run were used to commission the estimator. During this period, the process was run open loop with an input PRBS of \(\pm 1 \text{ min}^{-1}\). The diagonal elements of the covariance matrix were set to \(10^6\) initially and reset to this value whenever the estimation error exceeded a specified error tolerance. A PRBS of \(\pm 0.5 \text{ min}^{-1}\) was also added to the controller output from the time the covariance matrix was reset until the parameters converged sufficiently to update the controller model. Following a covariance reset, another reset was not permitted for 2 min. The forgetting factor for the estimator was chosen to be 0.8.

4.1.1. The adaptive nonlinear predictive controller

Because the input \(q_c(t)\) does not appear in eq. (36), \(\hat{C}_A(t)\) is differentiated again. The resulting equation is

\[
\hat{C}_A(t + d) = \left\{ \frac{q(t)}{V} + k_0 \exp \left[ \frac{-E}{RT(t)} \right] \right\} \hat{C}_A(t + d) + \frac{E k_0}{RT^2(t)} \exp \left[ \frac{-E}{RT(t)} \right] \hat{T}(t) C_A(t + d).
\]

(38)

\(q_c(t)\) appears in eq. (38) after the substitution of eq. (37) for \(\hat{T}(t)\). A discrete output model in which \(q_c\) appears can be written using an implicit second-order interpolation equation [24]:

\[
C_A(t) = C_A(t - 1) + \Delta t \hat{C}_A(t) - \frac{(\Delta t)^2}{2} \hat{C}_A(t).
\]

(39)

Equation (39) can be rewritten in a predictive form as the following implicit output model:

\[
\hat{C}_A(t) = \hat{C}_A(t - 1) + \Delta t f_1(t) - \frac{(\Delta t)^2}{2} f_2(t)
\]

(40)

where

\[
f_1(t) = \hat{C}_A(t), \quad f_2(t) = \hat{C}_A(t)
\]

(41)

and \(f_1(t)\) and \(f_2(t)\) are functions of past outputs and inputs. Equation (40) and the future values of \(f_1(t)\) and \(f_2(t)\) are evaluated using eqs (36)-(38) with the improved Euler method [24].

Parameter estimation

Because the output model was an approximation to the process, an assumption was made that its mismatch with the process over different operating conditions could be accounted for with condition-dependent coefficients \(\beta_1\) and \(\beta_2\) on the derivatives and a bias term \(\delta\). Thus, the output model was reexpressed as

\[
\hat{C}_A(t) = \hat{C}_A(t - 1) + \beta_1 \Delta tf_1(t) - \beta_2 \frac{(\Delta t)^2}{2} f_2(t) + \delta.
\]

(42)

The three coefficients, \(\beta_1, \beta_2,\) and \(\delta\), were then estimated by RLS with directional forgetting. The initial values of the three parameter estimates were set to 1, 1, and 0, respectively. The covariance matrix was reset whenever the estimation error exceeded 0.0004 mol/l.

Controller design

The pole-placement method of design was used. For this controller, eq. (43) was used as the output function \(F[t]\), and \(\hat{A}(q^{-1})\), the tuning polynomial, was chosen to be \((1 - 0.8q^{-1})^2\). This choice attempts to give the closed-loop system a 99.9% response time of about 4 min. Based on the chosen \(\hat{A}(q^{-1})\), \(R\) was set to 40. The remaining tuning parameters were \(L = 1, f = 0,\) and \(Q = I\). The prediction model was updated with the estimates of \(\beta_1, \beta_2,\) and \(\delta\) whenever the estimation error remained below 0.0002 mol/l for five consecutive samples. The following equations describe the prediction model for which the controller was designed.

\[
\hat{C}_A(t) = 1.6\hat{C}_A(t - 1) - 0.64\hat{C}_A(t - 2) + v(t - k - 1)
\]

(43)

where

\[
v(t - k - 1) = \hat{C}_A(t - 1) + \beta_1 \Delta tf_1(t) - \beta_2 \frac{(\Delta t)^2}{2} f_2(t) + \delta
\]

\[
- 1.6\hat{C}_A(t - 1) + 0.64\hat{C}_A(t - 2).
\]

(44)

4.1.2. The adaptive linear predictive controller (ALPC)

Prediction model

A 100-term step response model was used for the predictions and the controller design. One hundred terms were needed to provide a 93% complete response (180 for a 99% response). This prediction model was obtained by long division of a second-order difference equation model that was updated on-line.

Parameter estimation

Six parameters of a linear difference-equation model were estimated: three numerator coefficients, two denominator coefficients, and a bias term. The initial parameter estimates were zero. The covariance matrix was reset whenever the estimation error exceeded 0.00025 mol/l.
Controller design
The ALPC was designed with the following tuning parameters: \( L = 1, R = 100, Q = I, \) and \( f = 9 \) times the sum of the squares of the step response coefficients \( (f \approx 0.01) \). \( f \) was chosen so large because the control action was too vigorous for the set-point change to increase \( C_A \). The prediction model parameters were updated whenever the estimation error remained below 0.0001 mol/l for five consecutive samples. In addition, a static reactor-temperature compensator was added to the controller output to provide a fair comparison with the ANPC:

\[
\Delta u_{ff} = K_{ff}[T(t) - T(t - 1)].
\]

\( \Delta u_{ff} \) is the compensator output and \( K_{ff} = -1.0 \text{ min}^{-1} \text{ K}^{-1} \) is the compensator gain.

4.1.3. The PI controller. The initial PI controller tuning parameters were selected using the ITAE criterion for set-point changes. The controller gain was adjusted to give a good response for the set-point change to increase \( C_A \) from 0.1 to 0.15 mol/l. The
controller gain was $52 \text{ l}^2\text{ mol}^{-1}\text{ min}^{-1}$ and the integral time was 0.46 min. In addition, the static reactor-temperature compensator in eq. (45) was again used.

4.2. Simulation results

4.2.1. Small modeling errors in the ANPC. The first comparison is between the ANPC, the ALPC, and the PI controller for set-point tracking performance. The set point was changed at 8 min intervals from 0.1 mol/l to 0.15, to 0.1, to 0.05, and back to 0.1. The ANPC prediction model was designed using the correct values of all process parameters. Consequently, the prediction model is as accurate as the second-order discretization in eq. (41) permits. As can be seen in Fig. 2, the corresponding response of the ANPC is excellent: the responses to all of the set-point changes are symmetric and the response times are as designed (about 4 min).

Figure 3 compares the disturbance rejection performance of the same three controllers. The unmeasured feed concentration disturbance changes from 1.0 mol/l to 0.95 at 8 min, and back to 1.0 at 24 min. The measured coolant temperature disturbance decreases by $10^\circ\text{C}$ at 16 min and returns to its
nominal value at 32 min. Again, the nonlinear controller easily outperforms the other controllers.

4.2.2. Large modeling errors in the ANPC. The next simulation compares the set-point tracking performance of the ANPC and its corresponding (nonadaptive) nonlinear predictive controller (NPC). Both controllers were designed with significantly erroneous model parameters: the activation energy was assumed to be 5% higher than the actual value, the rate constant was 15% lower, the heat of reaction was 20% higher, and the heat transfer coefficient was 20% higher. As can be seen in Fig. 4, the NPC controller becomes unstable without adaptation. The ANPC, although not yielding symmetric, critically damped responses, still settles fairly quickly with reasonable amounts of overshoot.

Figure 5 compares the disturbance rejection performance of the ANPC and the NPC. Both controllers perform well under these circumstances. Neither controller performs significantly better than the other for three of the four disturbances. The ANPC, although not yielding symmetric, critically damped responses, still settles fairly quickly with reasonable amounts of overshoot.

5. CONCLUSIONS

A new adaptive nonlinear predictive controller has been proposed. The controller is tuned in a manner similar to linear model-predictive controllers or by placing poles of the closed-loop response. The new controller is computationally efficient and can perform well, even when initially designed with modeling inaccuracies. Simulations show that the new control strategy can outperform PI control, adaptive, linear predictive control, and nonadaptive, nonlinear predictive control strategies.

Acknowledgements—The authors gratefully acknowledge the financial support of the National Science Foundation (NSF CBT-8605233). Appreciation is expressed to Mike Henson for helpful discussions.

REFERENCES