Adaptive predictive functional control of a class of nonlinear systems

Bin Zhang* Weidong Zhang

Department of Automation, Shanghai Jiaotong University, Shanghai 200030, China
(Received 8 March 2005; accepted 29 June 2005)

Abstract

This paper describes the use of pseudo-partial derivative (PPD) to dynamically linearize a nonlinear system, and aggregation is applied to the predicted PPD, resulting in a model-free adaptive predictive control algorithm for a nonlinear system. The algorithm design is only based on the PPD derived online from the input/output data of the controlled process, however it does provide bounded input/output sequence and setpoint tracking without steady-state error. A detailed discussion on parameter selection is also provided. To show the capability of the algorithm, simulations of a time-delay plant and a pH neutralization process show that the proposed method is effective for system parameter perturbation and external disturbance rejection. © 2006 ISA—The Instrumentation, Systems, and Automation Society.

Keywords: Predictive functional control; Nonlinear system; Aggregation; Dynamic linearization; Base functions

1. Introduction

During the past decade the area of nonlinear system control has been a forum for many researchers. This is motivated to a large extent by the fact that nonlinear systems are difficult to control, hence no general methods for algorithm design are available.

A discrete SISO nonlinear system can be typically described by the following equation:

\[ y_p(k + 1) = f(y_p(k), \ldots, y_p(k - n_y), u(k), \ldots, u(k - n_u)) \]

where \( n_y \) and \( n_u \) are orders of outputs \( y_p \) and inputs \( u \), respectively, and \( f \) denotes a nonlinear mapping function.

*Corresponding author. Tel: +86.21.62826946; fax: +86.21.54260762. E-mail address: zhangbin7701@sjtu.edu.cn

The open literature presents a variety of nonlinear control algorithms that are based on special nonlinear models such as Hammerstein model [1], Wiener model [2], and bilinear model [3], which have been at the forefront of nonlinear systems research. However, it is difficult to find an appropriate nonlinear model “\( f \)” to describe the real process. Additionally, the nonlinear optimization problem is rarely convex which adds to the difficulty of the online calculation. Therefore, researchers turned to neural networks [4] which do satisfactorily map a bounded nonlinear function, however there still exists the difficulty in real time of a rapid, reliable application of the control algorithm.

More recently, the nonlinear function is assumed to be generalized Lipschitz with respect to output and input, and a model-free learning adaptive algorithm is proposed by Hou and Huang [5] based on input and output information in which a new concept called pseudo-partial derivative
The PFC algorithm is used to linearize the nonlinear systems online. This idea was also used to develop an adaptive-predictive PI controller [6]. Model predictive control has been applied to nonlinear processes [7–10], resulting in predictive functional control (PFC) [11–14], which is a most promising model predictive control algorithm. The PFC algorithm achieves computational simplicity by using simpler but more intuitive design guidelines [13] and in the past decade has been successfully used in industrial applications. The advantages of fewer online calculations, a simpler algorithm and higher control precision are attributes of the PFC, which contribute to its industrial use.

Motivated by the work of Hou and Huang [5] and Tan et al. [6], this paper presents an extension of PFC to nonlinear system control in which the PPD concept is used to dynamically linearize the nonlinear system. In other words, the PPD re-linearizes the nonlinear model as the plant moves from one operating point to another, and uses the latest linear model as the internal model at each step, resulting in solving a quadratic performance (QP). Aggregation as part of the algorithm [15] is used to predict future values of the PPD. Then PFC is used to design the nonlinear adaptive control algorithm, in which only two coincidence points are selected to calculate the manipulated variable. The resultant controller has a simple structure, hence tuning is not a problem. The proposed algorithm can also provide bounded input/output sequences and track the setpoint without steady-state error. Last, the paper discusses the parameter tuning, and uses simulations for a long time delay plant and an experimental setup for the measurement of the acidity or alkalinity in a chemical reaction process to show that the proposed algorithm has higher precision and better robustness to parameter perturbation than a PI or PID controller.

2. Nonlinear adaptive predictive functional control algorithm (NPFC)

2.1. The dynamic linearized internal model

For a nonlinear system (1), the following two assumptions are necessary:

**Assumption 1 (A1).** The partial derivative of \( f(\cdot) \) with respect to the control input \( u(k) \) is continuous.

**Assumption 2 (A2).** The system is generalized Lipschitz, that is, satisfying \( |\Delta y_P(k+1)| \leq C|\Delta u(k)| \), for \( \forall k \) and \( \Delta u(k) \neq 0 \), where \( \Delta y_P(k+1) = y_P(k+1) - y_P(k) \), \( \Delta u(k) = u(k) - u(k-1) \) and \( C \) is a constant.

The Lipschitz constant \( C \) is often required to be known for the control design purpose. Based on the above assumption, the following result can be obtained.

**Theorem 1.** [5] For the nonlinear system (1), we assume that Assumptions (A1) and (A2) hold. Then there must exist \( G(k) \), called PPD, when \( \Delta u(k) \neq 0 \),

\[
\Delta y_P(k+1) = G(k)\Delta u(k),
\]

where

\[
|G(k)| \leq C \tag{3}
\]

With Theorem 1, Eq. (2) can be used as an internal model to predict future process outputs

\[
y(k+1) = y(k) + \hat{G}(k)\Delta u(k), \tag{4}
\]

where

\[
|\hat{G}(k)| \leq C, \tag{5}
\]

\( y(k) \) is the model output, and \( \hat{G}(k) \) is an estimate of \( G(k) \).

2.2. Predictive output and structured control variables

Using Eq. (4), at sampling time \( k + H_l \) inside the optimization horizon, the future output can be predicted by

\[
y(k + H_l) = y(k) + \sum_{j=1}^{H_l} \hat{G}(k + j - 1)\Delta u(k + j - 1). \tag{6}
\]

The PFC algorithm is different from other model predictive controls. Instead of calculating control signal with no restrictions, which may result in an unstable control signal, PFC uses structured future manipulated variables, that is, the future manipulated variables are parameterized by \( n_B \) base functions \( u_{Bj} \).
\[ u(k + n) = \sum_{j=1}^{n_B} \mu_j(k) u_{Bj}(n) \quad n = 0, 1, \ldots, H_j - 1, \]

where \( \mu_j(k) \) (\( j \in [1, n_B] \)) are unknown coefficients.

There are no restrictions for selecting these base functions, and the selection of base functions has no influence on the dynamic response or robustness and stability of the closed-loop system [12]. The base function can be selected as polynomial, sine, or exponential format. For many applications it is sufficient to describe the process input using a form such as \( u(k + n) = \mu_1(k) + \mu_2(k)n \) \( n = 0, 1, \ldots, H_j - 1 \), which results in

\[ u(k) = \mu_1(k), \]

\[ \Delta u(k + H_j - 1) = \Delta u(k + H_j - 2) = \cdots = \Delta u(k + 1) = \mu_2(k). \]

Therefore, the determination of control input \( u(k + n) \) \( n = 0, 1, \ldots, H_j - 1 \) means to find coefficients \( \mu_1(k) \) and \( \mu_2(k) \) (at each instant time \( k \), and only \( u(k) = \mu_1(k) \) is applied to the process.

Substituting Eq. (8) into Eq. (6) results in

\[ y(k + H_j) = y(k) + \sum_{j=2}^{H_j} \hat{G}(k+j-1) \mu_2(k) \]

\[ + \hat{G}(k)[\mu_1(k) - u(k - 1)]. \]

2.3. Optimization and control law equation

The PFC algorithm computes future process input so that the predicted process output can follow a reference trajectory. In the PFC algorithm, the reference trajectory is used to specify the desired future process behavior. For many applications a first-order exponential reference trajectory is sufficient:

\[ y_{\text{ref}}(k + H_j) = w(k + H_j) - \eta \frac{H}{H_{\text{ref}}}(w(k) - y_p(k)), \]

where \( \eta = \exp(-T_j/T_{\text{ref}}) \), \( T_{\text{ref}} \) is the desired response time of the closed loop system, \( w \) is the setpoint, and for constant value setpoint tracking \( w(k + H_j) = w(k) \).

The PFC algorithm requires an online optimizing method. When a QP index is used, the process inputs are calculated by minimizing the sum of the quadratic difference between the predicted process output and the reference trajectory at all coincidence points. The criterion takes the following form:

\[ \min J_p = \sum_{n=H_1}^{H_2} \left[ (y_{\text{ref}}(k + n) - y(k + n) - e(k + n)) \right]^2 
+ r \sum_{n=1}^{M} \Delta u^2(k + n - 1), \]

where \( r \) is a weighting efficient, \([H_1, H_2]\) \((H_2 > H_1)\) is a coincidence horizon, \( M \) \((M \leq H_2)\) is a control horizon, and \( e(k + n) \) is the prediction error compensation which is given by \( e(k + n) = e(k) = y_p(k) - y(k) \).

Substituting Eqs. (8)–(10) into Eq. (11), the calculation of the process input \( u(k) \) is straightforward provided \( \hat{G}(k+j) \) is known. Note, Eq. (5) states that for \( \forall k, |\hat{G}(k)| < C \), require that future predicted PPD \( \hat{G}(k+j) \) be bounded. To deal with this restriction the idea of aggregation is adopted to predict future PPD at \( \hat{G}(k+j) \).

Let the aggregated variable be the current PPD \( \hat{G}(k) \), then future predicted values of the PPD \{ \( \hat{G}(k+1), \hat{G}(k+2), \ldots, \hat{G}(k+j), \ldots, \hat{G}(k+H_j-1) \} \) can be described as the amplitude decaying sequence related to the aggregated variable, namely:

\[ \hat{G}(k+j) = \hat{G}(k)\lambda^j \quad (0 < \lambda \leq 1, j = 1, \ldots, H_j - 1) \]

where \( \lambda \) is an unknown decaying coefficient. Then, \( \hat{G}(k+j) \) can automatically meet the constraint requirement (5). Moreover, to facilitate the tuning of the controller, this paper sets \( \hat{G}(k) = \lambda \), then Eq. (12) can be modified to

\[ \hat{G}(k+j) = \lambda^{j+1} = \hat{G}(k)^{j+1} \quad (0 < \lambda < 1, j = 1, \ldots, H_j - 1). \]

However, the constraint for PPD \( \hat{G}(k) \) may cause an uncontrolled overshoot of \( \Delta u(k) \), thus a weight for the control input \( \Delta u(k) \) is introduced in the performance index Eq. (11). Since the control input (8) is used and the future predicted output is given by Eq. (9), here we set \( M = H_2 \).
For the assumption, \( \mu_1(k) \) and \( \mu_2(k) \) are unknown coefficients in Eq. (11), which requires that at least two coincidence points \( H_1T_s \) and \( H_2T_s \) should be selected. Using Eq. (8), Eq. (11) is rewritten as

\[
\begin{align*}
\min J_P &= [y_{ref}(k + H_1) - y(k + H_1) - e(k + H_1)]^2 \\
&+ [y_{ref}(k + H_2) - y(k + H_2) - e(k + H_2)]^2 \\
&+ r[\mu_1(k) - u(k)]^2 + r \sum_{n=2}^{M} \mu_2^2(k),
\end{align*}
\]

where

\[
u(k) = \mu_1(k) = \frac{[-A_1\hat{G}(k) - A_2\hat{G}(k) - ru(k - 1)]\left[S_1^2 + S_2^2 + r(M - 1)\right] - (-A_1S_1 - A_2S_2)[S_1\hat{G}(k) + S_2\hat{G}(k)]}{[S_1\hat{G}(k) + S_2\hat{G}(k)]^2 - [2\hat{G}^2(k) + r][S_1^2 + S_2^2 + r(M - 1)]}
\]

where \( A_i = w(k + H_i) + \eta_i^H(w(k) - y_p(k)) + \hat{G}(k)u(k - 1) - y_p(k) \) \( (i = 1, 2) \),

\[
S_i = \sum_{j=2}^{H_i} \hat{G}(k)^j, \quad (i = 1, 2), \quad M = H_2.
\]

Subject to Eq. (4), online searching for \( \hat{G}(k) \) is required, however in Eq. (2), many algorithms to estimate \( \hat{G}(k) \) can be used. We adopted the adaptive learning algorithm [5] for \( \hat{G}(k) \):

\[
\hat{G}(k) = \hat{G}(k - 1) + \frac{\Delta u(k - 1)}{\gamma + \Delta u^2(k - 1)}(\Delta y_p(k) - \hat{G}(k - 1)\Delta u(k - 1)),
\]

where \( \gamma > 0 \), and the initial value \( \hat{G}(0) \) of \( \hat{G}(k) \) are in the range of 0–1.

Since \( |\hat{G}(k-1)| \leq C \) and \( |G(k-1)| \leq C \), it is easy to obtain the following relation with Eq. (18):

\[
e(k + H_1) = e(k + H_2) = y_p(k) - y(k).
\]

Substituting Eqs. (8)–(10), (13), and (15) into Eq. (14), letting

\[
\frac{\partial J_P}{\partial \mu_1(k)} = 0, \quad \frac{\partial J_P}{\partial \mu_2(k)} = 0.
\]

The manipulated variable is given by

\[
|\hat{G}(k)| \leq \left| \frac{\gamma}{\gamma + \Delta u^2(k - 1)} \hat{G}(k - 1) \right| + \left| \frac{\Delta u(k - 1)}{\gamma + \Delta u^2(k - 1)} \Delta y_p(k) \right| \leq \left| \frac{\gamma}{\gamma + \Delta u^2(k - 1)} \hat{G}(k - 1) \right| + \left| \frac{\Delta u^2(k - 1)}{\gamma + \Delta u^2(k - 1)} \hat{G}(k - 1) \right| \leq \frac{\gamma}{\gamma + \Delta u^2(k - 1)} C + \frac{\Delta u^2(k - 1)}{\gamma + \Delta u^2(k - 1)} C = C.
\]

This inequality implies that the adaptive learning algorithm (18) for \( \hat{G}(k) \) also meets the constraint requirement (5).

Note that explicit control input constraints are not addressed in this paper, however when input and/or state-related constraints need be considered, the technique proposed by Abu el Ata-Doss [16] is usable.
3. Performance analysis of the closed loop control system and the algorithm implementation

In order to assure the convergence of the closed loop system, the following assumption is made.

**Assumption 3 (A3):** The PPD satisfies $G(k) > 0$ for all $k$.

Based on the previous assumption, the stability of the closed loop system is guaranteed in the following theorem.

**Theorem 2:** Subject to Assumptions (A1)–(A3), the algorithm Eq. (17) for the nonlinear system Eq. (1) is used to track the setpoint $w$, then coincidence points $H_i (i = 1, 2)$, the weighting coefficient $r > 0$, and the control horizon $M = H_2 > 1$ exist such that

$$\lim_{k \to \infty} |y_p(k + 1) - w| = 0$$

and $\{y_p(k)\}, \{u(k)\}$ are bounded sequences.

**Proof:** See the Appendix.

Note, the $G(k)$ in Eq. (A6b) is unknown at current time $k$ and $G(k) < C$. If we take the following inequality as a criterion for selecting controller parameters, then the condition (A6b) always holds

$$\hat{G}(k)^2 > C\hat{G}(k)(1 - \gamma^H) \quad (i = 1, 2).$$

Theorem 2 actually provides a criterion to select controller parameters. That is, existing parameters $r > 0$, $H_i (i = 1, 2)$, and $M = H_2 > 1$ insure that Eqs. (A4), (A6a), and (21) are applicable. Further, from Eq. (A3) increasing $r$ leads to $p$ decreasing, and Eq. (A2) shows that increasing $r$ results in a slower tracking of the setpoint. Theorem 2 also shows that $T_{ref}$ has no influence on stability. The rapidity of $T_{ref}$ will influence the dynamic response and robustness of the closed-loop system. The shorter $T_{ref}$, the more active the controller will be with larger amplitude variations [12].

Note that parameter estimation for $\hat{G}(k)$ is convergent and the coincidence points $H_1, H_2$, the input horizon $M = H_2$, the weighting coefficient $r$, and the reference trajectory time $T_{ref}$ are easily selected using Eqs. (A4), (A6a), and (21). Since the developed method has no plant structural requirement, the closed-loop control scheme is very robust. In practice, $H_1, H_2$, the initial value $G(0)$, and $T_{ref}$ can be fixed, and $r$ is used to tune so that an optimal compromise between performance and robustness can be reached.

Now we summarize the design procedure of the proposed NPFAC as follows:

**Step 1:** (Initialization) At time $k = 0$, take the initial value $\hat{G}(0)$ of PPD $\hat{G}(k)$ are in the range of 0–1, and set $\gamma > 0$.

**Step 2:** Find appropriate coincidence points $H_1, H_2$, and the closed-loop response time $T_{ref}$ by using the criterion (A4) and (21). Further, for a time delay system both the two coincidence points $H_1$ and $H_2$ should be selected larger than the dead-time in samples. Set the input horizon $M = H_2$ and the weighting coefficient $r$ to verify inequality (A6a).

**Step 3:** At time $k \geq 1$, collect the process input/output data, and find $\hat{G}(k)$ by using adaptive learning algorithm (18). Substitute $\hat{G}(k)$ into (A4), (A6a), and (21), if these three conditions fail, return to Step 2. Otherwise, go to Step 4.

**Step 4:** Use the control law equation (17) to calculate the process control input and apply it to the process.

**Step 5:** At the next point, repeat Steps 3 and 4.

4. Illustrative examples

The following long time delay plant and pH measurement of acidity or alkalinity process are used to show the effectiveness of the proposed algorithm.

**Example 1:** Consider a plant described by

$$P(s) = \frac{K}{(s + 1)^5}e^{-\theta s}$$

in which $K=1$ and $\theta=15$. Our goal is to use the proposed NPFCA to control such a large time-delay plant and to show that the proposed control method gives a better performance than a PI or PID controller. A comparison to the methods of Aström-Hagglund’s PI tuning (A-H PI) [17], Sigurd Skogestad’s IMC-PI and IMC-PID tuning (SIMC-PI, SIMC-PID) [18] is given. Using the above-mentioned tuning methods, the recommended controller parameters are as follows:

A-H PI: $G_c(s) = 0.2115 + 0.0286/s$

SIMC-PI: $G_c(s) = 0.0455(1 + 1/1.5s)$

SIMC-PID: $G_c(s) = 0.0322(1 + 1/s)(1 + 1.5s)$

For the proposed NPFCA, we choose the sampling time $T_s = 1$ s. Follow the design procedure in
Section 3, we take the initial value of PPD $\hat{G}(0) = 0.9$, and $\gamma = 0.9$. In Step 2, since the plant considered contains a large time delay, the coincidence points $H_1$ and $H_2$ should be selected larger than the dead-time. Then by Eqs. (A4), (A6a), and (21), the controller tuning parameters used here are assigned at coincidence points $H_1 = 20$, $H_2 = 22$, the input horizon $M = H_2 = 22$, the desired closed-loop response time $T_{ref} = 1$ s, and the weighting coefficient $r = 75$.

For a step change in the unit setpoint and a load disturbance at $t = 0$ s and $t = 185$ s, respectively, Fig. 1 presents these step responses of the closed-loop control system. One can see that for setpoint changes, as well as disturbance inputs the A-H PI method, SIMC-PI, and SIMC-PID methods do not reach the final set point at about time $t = 300$ s, whereas the newly developed method provides significant improvement in both setpoint response and disturbance rejection. Further, it is easy to verify that at each sampling time for the proposed controller parameters the conditions (A4), (A6a), and (21) always hold.

Fig. 2 shows the responses to process parameter perturbation: $K = 1.3$ and $\theta = 16$. Note that the SIMC-PI and SIMC-PID methods are very oscillatory, and A-H method provides poor disturbance rejection. The proposed method provides a smooth setpoint response and an acceptable disturbance rejection. The improvement on performance is due to that the proposed algorithm has no plant structural requirement and adopts an adaptive learning algorithm to on-line estimate PPD $G(k)$.

**Example 2:** A pH neutralization process can be modeled by the following equation [14]:

$$x(k) = f_1(u(k)) = u(k) - (1.207 + r_1)u^2(k) + 1.15u^3(k), \quad (23a)$$

$$y(k) = \frac{(0.0185 + r_2)z^{-2} + (0.0173 + r_3)z^{-3} + 0.00248z^{-4}}{1 - (1.558 + r_4)z^{-1} + 0.597z^{-2}}, \quad (23b)$$

where $r_1$, $r_2$, $r_3$, and $r_4$ are time-varying parameters of the process in which the initial values are set to zero. Selecting the initial value of the PPD to $\hat{G}(0) = 0.98$, $\gamma = 0.9$, coincidence points $H_1 = 17$, $H_2 = 25$, the input horizon $M = 25$, the sampling time $T_s = 1$ s, the desired closed loop response time $T_{ref} = 1$ s, and $r = 300$, the step responses for setpoint tracking are shown in Fig. 3.

It can be found that the proposed method does achieve excellent results. Fig. 4 shows the response to a unit output disturbance at time $t = 250$ s with no change in controller parameters. Again the proposed controller performs very well. The proposed control system can track setpoint without steady-state error although there is an existing external disturbance. In addition, for the
case of process parameter perturbation occurring at different times $t=200 \ s$ ($r_1=0.1$, $r_2=0.01$) and $t=400 \ s$ ($r_3=0.001$, $r_4=-0.008$), the step response is shown in Fig. 5. One can see that the proposed algorithm has excellent robustness.

5. Conclusions

The PPD was used to dynamically linearize a nonlinear process, and aggregation was used to predict the PPD, resulting in an adaptive predictive functional control algorithm (APFCA) for nonlinear processes. The proposed algorithm was tested on two processes and was shown to clearly outperform existing algorithms.

A theorem, which illustrates that the designed control system can track the setpoint with zero error and input/output sequences are bounded, was derived in this paper. A merit of the proposed controller algorithm is that it does not require the structure of the plant or any further external forcing for purposes of model development. Note, the results presented in this paper can be extended to MIMO nonlinear processes, and the proposed scheme can be easily implemented.

Appendix: Proof of theorem 2

For constant value setpoint tracking, the error between the output $y_P$ and the constant setpoint $w$ can be written as follows:

$$ E(k+1) = |y_P(k+1)-w| = |y_P(k)-w + G(k)u(k) - u(k-1)|. \tag{A1} $$

Substituting Eq. (17) into Eq. (A1) results in Eq. (A2).

$$ E(k+1) \leq |1-\rho|E(k), \tag{A2} $$

where

$$ \rho = \frac{\xi}{\nu}, $$

$$ \xi = G(k)\hat{G}(k)(S_2-S_1)[(1-\eta^H)S_2-(1-\eta^H^2)S_1]+G(k)\hat{G}(k)r(M-1)(2-\eta^H-\eta^H^2), $$

$$ \nu = \hat{G}^2(k)(S_1^2+S_2^2-2S_1S_2)+[2\hat{G}^2(k)+r]r(M-1)+rS_1^2+rS_2^2. \tag{A3} $$

From Assumption 3 and Eq. (18), it is easy to know that $\hat{G}(k)>0$. Note, $S_2>S_1$, Assumption 3.
states $G(k)\hat{G}(k) > 0$, and Eq. (10) shows that $0 \leqslant (1 - \eta_H^i) < 1$ $(i = 1, 2)$, therefore, if we set $r > 0$, $M = H_2 > 1$, and $H_i (i = 1, 2)$ such that

\[(1 - \eta_H^i)S_2 > (1 - \eta_H^2)S_1 \quad \text{(A4)}\]

results in $\rho > 0$.

Furthermore, from Eq. (A3) one can derive Eq. (A5).

\[v - \bar{\xi} = S_1^2[\hat{G}(k)^2 - G(k)\hat{G}(k)(1 - \eta_H^i)] + S_2^2[\hat{G}(k)^2 - G(k)\hat{G}(k)(1 - \eta_H^i)] + \sum_{i=1}^{2} [\hat{G}(k)^2\{r(M-1) - S_1S_2\} + r^2(M-1) + r(S_1^2 + S_2^2). \quad \text{(A5)}\]

If $r > 0$, $M = H_2 > 1$, and $H_i (i = 1, 2)$ are selected such that

\[r(M-1) > S_1S_2 \quad \text{(A6a)}\]

and

\[
\rho_1 = \frac{\hat{G}(k)(S_2 - S_1)[(1 - \eta_H^i)S_2 - (1 - \eta_H^2)S_1] + \hat{G}(k)r(M-1)(2 - \eta_H^i - \eta_H^2)}{\hat{G}^2(k)(S_1^2 + S_2^2 - 2S_1S_2) + [2\hat{G}^2(k) + r]r(M-1) + rS_1^2 + rS_2^2}
\]

and a bound for $\rho$ and $G(k)$ exist, $\rho_1$ will be bounded.

Combing Eqs. (17) and (A3) resulting in

\[\Delta u(k) = u(k) - u(k-1) = \rho_1[w - y_P(k)] \quad \text{(A10)}\]

and

\[|\Delta u(k)| \leqslant \rho_1 \max E(k) \quad \text{(A11)}\]

where $\rho_1 \max$ is the upper bound of $\rho_1$. Using the absolute triangle inequality property to Eq. (A10), it results,

\[\hat{G}(k)^2 > G(k)\hat{G}(k)(1 - \eta_H^i)(i = 1, 2). \quad \text{(A6b)}\]

Then $v > \bar{\xi} > 0$, and is $\rho = \xi/v < 1$, and $0 < 1 - \rho < 1$.

By Eq. (A2), the following inequality holds:

\[E(k+1) \leqslant |1 - \rho|E(k) \leqslant |1 - \rho|^2E(k-1) \leqslant \cdots \leqslant |1 - \rho|^{k+1}E(0). \quad \text{(A7)}\]

Then

\[
\lim_{k \to \infty} |y_P(k+1) - w| = \lim_{k \to \infty} E(k+1) = \lim_{k \to \infty} (1 - \rho)^{k+1}E(0) \quad \text{(A8)}
\]

Since $0 < 1 - \rho < 1$ and $E(0)=|y_P(0) - w|=|w|$, it is easy to know by (A8) that Eq. (20) holds and the control system track setpoint with no steady state error.

Moreover, we use the following form for $\rho_1$:

\[\rho = G(k)\rho_1, \quad \text{(A9)}\]

where

\[|u(k)| \leqslant |u(k) - u(k-1)| + |u(k-1)| \leqslant |\Delta u(k)| + |u(k-1) - u(k-2)| + |u(k-2)| \leqslant \cdots \leqslant |\Delta u(k)| + |u(k-1)| + \cdots + |\Delta u(2)| + |u(1)| \quad \text{(A12)}\]

Thus by Eq. (A11) $\{y_P(k)\}, \{u(k)\}$ are bounded sequences.

\[\square\]

Acknowledgments

The authors would like to thank the National Natural Science Foundation of China (60274032), Specialized Research Fund for the Doctoral Pro-
gram of Higher Education (SRFDP)
(20030248040) for financial support of the re-
search project.

References

Nonlinear model predictive control using Ham-
Application of wiener model predictive control
(WMPC) to a pH neutralization experiment. IEEE
continuous-time systems identification via hartley-
based modulating functions. Automatica 4, 499–503
[4] Zhang, J., Developing robust non-linear models
through bootstrap aggregated neural networks. Neuro-
computing 25, 93–113 (1999).
adaptive control of a class of SISO nonlinear systems.
Proceedings of the American Control Conference,
Adaptive-predictive PI control of a class of SISO sys-
tems. Proceedings of the American Control Confer-
ce, California, 3848–3852, 1999.
[7] Maqni, L. and Scattolini, R., Stabilizing model predic-
tive control of nonlinear continuous time systems.
based predictive control through dynamic non-linear
(2002).
[9] Kwon, Wook Hyun, Han, SooHee, and Ahn, Choon
Ki, Advances in nonlinear predictive control: A survey
on stability and optimality. International Journal of
[10] Qin, S. J. and Badgwell, T. A., An overview of non-
linear model predictive control applications. Nonlinear
Model Predictive Control, edited by Allgöwer, F. and
[11] Richalet, J., Industrial application of model based pre-
[12] Ernst, T. E. F. H. C., First principle modeling and pre-
dictive functional control of enthalpic processes, Ph.D.
thesis, Delft University of Technology, Delft, The
Netherlands, 1996.
with predictive functional control of unstable pro-
cesses. Proceedings of the American Control Confer-
[14] Zhang, Q. L., Predictive functional control and its ap-
[15] Xi, Y. G., The principles and methods in large scale
[16] Abu el Ata-Doss, S., Finai, P., and Richalet, J., Hand-
ing input and state constraints in predictive functional
control. Proceedings of the 30th Conference on Deci-
[17] Astrom, K. J. and Hagglund, T., Industrial adaptive
controller based on frequency response techniques.
[18] Skogestad, S., Simple analytical rules for model re-
duction and PID controller tuning. J. Process Control