ISA TRANSACTIONS®

ISA Transactions 42 (2003) 207–226

Predictive feedback control

Leonardo L. Giovanini*

Electronic and Control Research Group, Universidad Tecnológica Nacional, Regional School at Villa Maria, Av. Universidad 450, 5900 Villa María, Argentina

(Received 19 July 2001; accepted 29 April 2002)

Abstract

In this work a new method for designing predictive controllers for linear single-input/single-output systems is presented. It uses only one prediction of the process output J time intervals ahead to compute the correspondent future error. Then, the predictive feedback controller is defined by introducing a filter which weights the last w predicted errors. In this way, the resulting control action is computed by observing the system future behavior and also by weighting present and past errors. This last feature improves the closed-loop performance to disturbance rejection as shown through simulations of two linear systems and a nonlinear continuous stirred tank reactor. © 2003 ISA—The Instrumentation, Systems, and Automation Society.

Keywords: Predictive control; Model-based control; Feedback output

1. Introduction

The use of different kinds of linear models to predict the future behavior of the process output has stimulated the development of a wide group of control methods known as model predictive control (MPC). Many MPC approaches have been proposed along the past three decades (generalized predictive control [1], dynamic matrix control [2], model algorithmic control [3]), most of them based on a receding-horizon strategy, i.e., at each sampling instant k the following actions are taken: (a) the plant model is used to predict the output response to a hypothetical set of future control signals, (b) a function including the cost of future control actions and future deviations from a reference trajectory is optimized to give the *best future control sequence*, and (c) the first movement of the control sequence is applied. These operations are repeated at time $k+1$.

The main advantage of MPC is its ability to address (a) long time delay, (b) inverse response, (c) significant nonlinearities, (d) multivariable interactions, and (e) constraints. The widespread use and success of MPC applications described in the literature [4] attest to the improved performance of MPC compared to the classical control algorithm for control of difficult process dynamics. However, alternative algorithms have been developed to address the same problems [analytical predictor algorithm [5] (APA), predictor controller [6] (PC), simplified predictive control [7], variable horizon predictor [8] (VHP)] because MPC implementations require sophisticated tools that do not allow us to apply predictive control at all levels of control systems. All of these algorithms employ, in different ways, only one prediction of the future error to compensate time delay and interactions. For example, simplified predictive control and PC

*Tel.: 54-353-4537500; fax: 54-353-4535498. *E-mail address*: lgiovani@fich1.unl.edu.ar.

predict the system output at its steady-state value (prediction time is set equal to the convolution length) and then develop the controller structure. On the other hand, APA predicts the controlled variable on dead time plus one sample and then uses the predicted error as input into the controller. Finally, VHP uses the predicted error as the input into the controller, like APA, but the prediction can be freely chosen from the hole prediction horizon and it does not impose any constraint on the controller that can be used.

In this work a new method for designing a predictive is presented. The approach is based on the use of only one prediction of the system output, instead of the complete trajectory: it uses a prediction of the process output J time intervals ahead to compute the correspondent future error. The proposed controller, called *predictive feedback controller*, uses the last w predicted errors instead of using plain feedback errors, as in classical feedback controllers. Hence the resulting control action is computed by observing the system future behavior and also by weighting present and past errors. So, this control strategy combines the predictive capacity, which results in good performance for set-point changes and time delay systems, with the classical use of the feedback information which improves the system performance for disturbance rejection.

The organization of the paper is as follow: in Section 2 the expressions for a general J-step ahead output prediction are presented. In Section 3 the basic formulation for the single-prediction controller design is derived. Furthermore, a relationship between the controller parameter and the settling time of closed-loop response is established. In Section 4 the closed-loop stability and performance of the single-prediction controller are analyzed. In Section 5 a direct feedback mode is introduced in order to improve the overall system performance. Besides, the relationship between the proposed controller and other predictive control algorithms is established. The closed-loop stability and performance of the resulting controller are analyzed in Section 6. In Section 7 we show the results obtained from the application of the proposed algorithm to a nonlinear continuous stirred tank reactor. Finally, the conclusions are presented in Section 8.

2. Single-step output predictor

In many predictive control techniques, the model more frequently used to develop the predictor is the discrete convolution truncated to N terms [4,9]. The reason is twofold: (a) the convolution summation gives the model output explicitly and (b) the main impulse response coefficients are relatively easy to obtain. In particular, for single-input/single-output (SISO) systems

$$\hat{y}(J,k)=\sum_{i=1}^{N}\tilde{h}_i u(k+J-i),\quad J<N, \tag{1}$$

predicts the output value J sampling intervals ahead, k represents the current time instant $t=kt_S$ (t_S is the sampling interval), $\tilde{h}_i$, $i=1,2,\ldots,N$ are the impulse response coefficients, and $u(k+J-i)$, $i=1,2,\ldots,N$ is the sequence of inputs to be considered. However, most frequently $\hat{y}(J,k)$ is not calculated directly from Eq. (1) but from a modified expression that includes the prediction for the current time $\hat{y}(0,k)$. For this, notice that Eq. (1) can also be written as a function of the predicted value for the previous sampling time $J-1$,

$$\hat{y}(J,k)=\hat{y}(J-1,k)+\sum_{i=1}^{N}\tilde{h}_i\Delta u(k+J-i), \tag{2}$$

where $\Delta u(k+J-i)=u(k+J-i)-u(k+J-i-1)$. Then, successive substitutions of $\hat{y}(J-1,k)$ by previous predictions gives

$$\hat{y}(J,k)=\hat{y}(0,k)+\sum_{l=1}^{J}\sum_{i=1}^{N}\tilde{h}_i\Delta u(k+l-i). \tag{3}$$

This equation defines a *J-step ahead predictor*, which includes future control actions. Since future control actions are unknown, the predictor (3) is not realizable. To turn it realizable a statement must be made about how the control variable is going to move in the future. For example, the simplest rule is to set all of them equal to zero,

$$\Delta u(k+j)=0,\quad \forall j=0,\mathbf{1},\ldots,J, \tag{4}$$

which implies that the control variable will not move in future sampling instants. Then, Eq. (3) becomes

$$\hat{y}^0(J,k)=\hat{y}(0,k)+\sum_{l=1}^{J}\sum_{i=l+1}^{N}\tilde{h}_i\Delta u(k+l-i), \tag{5}$$

where the superscript 0 recalls that condition (4) is included. The new expression (5) defines a realizable *open-loop J-step ahead predictor* whose $\mathcal{Z}$ transform is given by (see Appendix A)

$$\hat{y}^0(J,z)=P(J,z)u(z), \tag{6}$$

where $P(J,z)$ is the transfer function of the open-loop predictor given by

$$P(J,z)=\tilde{a}_J z^{-1}+\sum_{i=J+1}^{N}\tilde{h}_i z^{J-i},$$

and $\tilde{a}_J$ is the Jth coefficient of step response.

The prediction $\hat{y}^0(J,k)$ is updated by adding

$$\tilde{d}(J,z)=y(J,z)-\hat{y}(J,z). \tag{7}$$

This term lumps together possible unmeasured disturbance and inaccuracies due to plant-model mismatch. Since the future value of $\tilde{d}(J,z)$ is not available, an estimate is used. In the absence of any additional knowledge of $\tilde{d}(J,z)$, the predicted disturbance is assumed to be equal to that estimated at the current time $\tilde{d}(z)$. A more accurate estimate of $\tilde{d}(J,z)$ is possible if the disturbance output model and a measure of load disturbance are available. So, we use a similar equation to Eq. (6) to predict the future disturbance. The importance of the form of Eq. (5) comes also from the fact that the prediction $\hat{y}^0(J,k)$ can be updated by the current output measurement $y(k)$. This is done by substituting $\hat{y}(0,k)$ by $y(k)$, or equivalently the correction is implemented by $\tilde{d}(z)$ adding to Eq. (6), in any case we obtain

$$\hat{y}^0(J,z)=y(z)+[P(J,z)-\tilde{G}p(z)]u(z), \tag{8}$$

where $\tilde{G}p(z)$ is the plant model. This equation defines a realizable corrected *J-step ahead predictor*. Hence there are two type of predictors: one is the open-loop predictor that only depends on the values of the past inputs only [Eq. (6)] and the other is the corrected predictor (8) which receives a correction through the feedback measurement.

3. Single-prediction control

Although the idea of using only one prediction of the system output for controlling the system is not new [5–8], all the authors did not use the prediction time J as a tuning parameter. This is the case of *single-prediction controller*, whose derivation procedure is quite straightforward from the open-loop predictor. Revising the assumption used to go from Eq. (3) to Eq. (5) observe that if just the control movement $\Delta u(k)\neq 0$, then

$$\hat{y}(J,k)=\hat{y}^0(J,k)+\tilde{a}_J\Delta u(k). \tag{9}$$

This prediction can be substracted from a reference variable $r(k+J)$ to obtain the predicted error

$$\hat{e}(J,k)=\hat{e}^0(J,k)-\tilde{a}_J\Delta u(k). \tag{10}$$

The control action can be computed in the similar way as standard predictive controllers, minimizing the following performance measure,

$$f(k)=\hat{e}^2(J,k)+\rho\Delta u^2(k), \quad \rho\geqslant 0. \tag{11}$$

Then, the control action that minimizes this performance index is given by

$$\Delta u(k)=\frac{1}{K_J}\hat{e}^0(J,k), \tag{12}$$

where $\hat{e}^0(J,k)=r(k+J)-\hat{y}^0(J,k)$ and $K_J=\tilde{a}_J+\rho/\tilde{a}_J$, and the cost of controlling the system is

$$f(k)=\frac{\rho}{\tilde{a}_J^2+\rho}\hat{e}^{0^2}(J,k).$$

At this point of the work we are interested on understanding the meaning of prediction time J and its relationship with the closed-loop response. To do it, we replace $\Delta u(k)$ and K_J in the closed-loop predicted error [Eq. (10)], so we can write it as function of $\hat{e}^0(J,k)$ and ρ,

$$\hat{e}(J,k)=\frac{\rho}{\tilde{a}_J^2+\rho}\hat{e}^0(J,k).$$

Then, defining $\hat{e}(J,k)$ as a fraction of $\hat{e}^0(J,k)$,

$$\hat{e}(J,k)=\alpha\hat{e}^0(J,k), \quad |\alpha|<1,$$

where α is the remaining error after the control action $\Delta u(k)$ is applied, the control cost ρ is related with α through

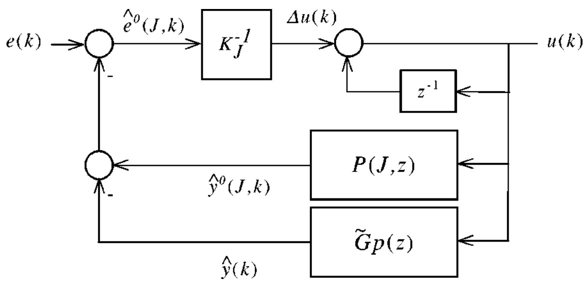

Fig. 1. Structure of the single-prediction controller.

$$\rho=\frac{\alpha}{1-\alpha}\tilde{a}_J^2.$$

This equation means that the closed-loop error will be bounded by $|e(k_0+J)|\leqslant\alpha|\hat{e}^0(J,k_0)|$, $\forall t>k_0+J$, where k_0 is time instant where the set-point changes. Therefore the prediction time J could be seen as the *closed-loop settling time* for an error $\alpha|e(k_0)|$. When the control cost ρ is set to 0, which is equivalent to $\alpha=0$, the performance measure (11) and the predicted closed-loop error $\hat{e}(J,k)$ becomes zero, and the control change is given by

$$\Delta u(k)=\frac{1}{\tilde{a}_J}\hat{e}^0(J,k). \tag{13}$$

On the other hand, when ρ is set to ∞ (which is equivalent to $\alpha=1$) no control action is taken and $\hat{e}(J,k)=\hat{e}^0(J,k_0)=e(k_0),\quad \forall k>k_0$.

Using $\mathcal{Z}$ transform on Eq. (12) shows that the single predictive control algorithm is basically an integral action applied on the predicted error

$$u(z)=\frac{1}{K_J}\frac{1}{1-z^{-1}}\hat{e}^0(J,z). \tag{14}$$

Combining Eqs. (14) and (8) the result is the controller,

$$C(z)=\frac{1}{(1-z^{-1})K_J+P(J,z)-\tilde{G}p(z)}. \tag{15}$$

Fig. 1 shows a block diagram of $C(z)$, where it is apparent that it uses the plant model to estimate the output at the present time $\hat{y}(0,k)$. This value is then compared with the actual measurement $y(k)$ to detect modeling errors and external disturbances. The global detected disturbance $\tilde{d}(z)$ is then assumed constant from every sampling instant k into the future. In other words, given all the input changes accounted for until the instant k, the single-prediction controller *observes* the value that would be reached by the system output if no future control action is taken and then $u(k)$ is computed such to the performance index (11) is minimized. Hence if $\tilde{d}(z)$ actually remains constant after the instant k and $\rho=0$, then the output reaches the reference value J sampling intervals later.

Note that controller (15) is realizable if and only if $\tilde{a}_J\neq 0$. Then, the prediction time J should be chosen such that

$$J\geqslant \operatorname{int}\left(\frac{t_d}{t_S}\right)+1,$$

where t_d is the process time delay. This fact means that the open-loop predictor $P(J,z)$ compensates the process time delay. In the following sections we assume that $\rho=0$ to simplify the expositions, however, the results that will be obtained are the same whether the control weight ρ is set to zero or not.

4. Algorithm properties

4.1. Stability analysis

To analyze the effect of the prediction time over the closed-loop stability, we substituted the controller (15) in the closed-loop characteristic equation to obtain

$$(1-z^{-1})\tilde{a}_J+P(J,z)+[Gp(z)-\tilde{G}p(z)]=0.$$

Then, combining this expression with Eqs. (A8) and (A2), and using the convolution model of the process, this equation becomes

$$\tilde{a}_J+\sum_{i=J+1}^{N}\tilde{h}_i z^{J-i}+\sum_{i=1}^{N}(h_i-\tilde{h}_i)z^{-i}+\sum_{i=N+1}^{\infty}h_i z^{-i}=0. \tag{16}$$

Using a result obtained by Desoer and Vidyasagar [10] for lineal discrete systems, we derive the following stability condition (see Appendix B):

$$\sum_{i=J+1}^{N}|\tilde{h}_i|+\sum_{i=1}^{N}|h_i-\tilde{h}_i|+\left|\sum_{i=N+1}^{\infty}h_i\right|<\tilde{a}_J. \tag{17}$$

The left side of this equation has, ordered from left to right, the following three terms: (a) the contribution of the nominal model, (b) the additive uncertainty, and (c) the effect of the truncation model error; all of them are related with the convolution length N. When there is not uncertainty in the system and if N is large enough to neglect the truncation error, Eq. (17) becomes

$$\sum_{i=J+1}^{N} |\tilde{h}_i| < \tilde{a}_{J_{stb}}. \quad (18)$$

This condition guarantees that—for any time-invariant stable plant—there is always a value of J for which the closed-loop system is asymptotically stable. We must note that if the plant has a monotone step response, the stability condition (18) can be written as

$$\tilde{a}_{J_{stb}} > \frac{1}{2} \sum_{i=1}^{\infty} \tilde{h}_i = \frac{1}{2} \tilde{K}p,$$

where $\tilde{K}p$ is the process gain.

Generally, control engineers assume that a family $\mathcal{W}$ of M linear models is capable to capture a moderate nonlinearity. Therefore to guarantee the stability of the system we must choose a J such that it guarantees the stability of all the plants of $\mathcal{W}$. Then, the robust stability problem becomes the problem of finding a J such that Eq. (17) is satisfied for each model of $\mathcal{W}$. Using global additive uncertainty and choosing N large enough to neglect the truncation error, Eq. (17) becomes

$$\sum_{i=J+1}^{N} |\tilde{h}_i| + \sum_{i=1}^{N} \max_{l \in [1,M]} |h_i^l - \tilde{h}_i| < \tilde{a}_{J_{stb}}. \quad (19)$$

Another way to solve the robust stability problem is to find a J that satisfies simultaneously Eq. (18) for all the models of $\mathcal{W}$. In the case of a system with monotone response, the stability condition (19) can be written as

$$J_{stb} = \max(J_1; J_2; \ldots; J_M),$$

where J_l, $l = 1,2,\ldots,M$ is the prediction time for the lth model of the family $\mathcal{W}$. This expression means that we can choose a different prediction time for each model of $\mathcal{W}$, then we selected the bigger prediction time.

At this point, a remark about how to build a predictor for nonlinear system must be made. Since J_{stb} guarantees the closed-loop stability for all the models of $\mathcal{W}$, the open-loop predictor $P(J,z)$ can be directly built from the nonlinear model. This fact improves the accuracy of the open-loop prediction and the closed-loop performance. The nonlinear predictor can be built from the nonlinear model employing a numerical integration scheme or using a local model network [11].

A final remark is for recalling that the stability condition given by Desoer and Vidyasagar [10] is a sufficient one. Consequently, stability conditions (17)–(19) become conservative and impose a too high lower bound for selecting J. So, always there are prediction times lower than J_{stb}; for those the closed-loop system will be stable. The first J that guarantees the closed-loop stability can be found through a direct search, because the solution space is bounded,

$$1 \leq J \leq J_{stb}.$$

4.2. Performance analysis

The tuning of the single-prediction controller implies a discrete optimization problem having a bounded solution space

$$J \in \mathcal{N}, \quad 0 \leq J \leq N.$$

Hence, independently of the performance index being used, the general solution results from a direct search in the solution space. However, when the ISE index is used, the optimal controller results from the solution of the following problem:

$$\min_{Gc(z)} \|e(z)\|_2^2 = \min_{Gc(z)} \|\tilde{\varepsilon}(z)[d(z) - r(z)]\|_2,$$

which is equivalent to minimize the sensitivity function $\tilde{\varepsilon}(z)$ over the whole system bandwidth,

$$\min_{Gc(z)} \|\tilde{\varepsilon}(z)\|_2 = \min_{Gc(z)} \|1 - \tilde{G}p(z) Gc(z)\|_2,$$

and to guarantee the internal stability of the closed-loop system [12]. Under these design conditions the sensitivity function will be minimum when the open-loop controller $Gc(z)$ is given by

$$Gc = \tilde{G}p_+^{-1},$$

where $\tilde{G}p_+$ is the nonminimum phase portion of the plant (right-half plane zeros and time delay).

For the single-prediction controller, this condition is equivalent to choose a prediction such that the Jth coefficient of the step response $\tilde{a}_J$ has the same sign as the stationary process gain, i.e.,

$$J_{ISE}=k_i+1, \tag{20}$$

where k_i is the number of samples which covers the effect of the minimum phase portion of the plant.

When $J=J_{ISE}$, the controller gain $\tilde{a}_J^{-1}$ is the largest. Therefore it provides a vigorous control action and attempts to drive the system output to the reference in J_{ISE} time intervals. In this case, the single-prediction controller becomes a *dead-beat* (minimum-time) *controller*

$$C(z)=\frac{1}{(1-z^{-1})\tilde{a}_{J_{ISE}}+P(J_{ISE},z)-\tilde{G}p(z)}. \tag{21}$$

When J is increased the controller gain and the closed-loop performance decreases. In case of $J=N$ the controller gain is the inverse of process gain ($\tilde{a}_J=\tilde{K}p$) and the controller drives the system output to the reference in N time intervals. Thus only one significant control move is observed in absence of uncertainty (minimum-energy controller) and the single-prediction controller becomes a *predictor controller* [6],

$$C(z)=\frac{1}{\tilde{a}_N-\tilde{G}p(z)}. \tag{22}$$

To finish this analysis, we give a heuristic comparison of the closed-loop performance achieved by the single-prediction controller with that of a MPC controller. To carry out this comparison we analyze the control actions computed by both controllers. They are obtained by replacing the open-loop error by their components, assuming a step change in the setpoint, in the controller equations [4,9] and combining them with the predicted output (8). The result for the single-prediction controller is

$$u(k)=u(k-1)+K_J^{-1}e(k) -K_J^{-1}[P(J,z)-\tilde{G}p(z)]u(k-1), \tag{23}$$

and the result for the MPC is

$$u(k)=u(k-1)+\left\{\sum_{j=1}^{V}k_j\right\}e(k) -\sum_{j=1}^{V}k_j[P(j,z)-\tilde{G}p(z)]u(k-1), \tag{24}$$

where V is the prediction horizon and k_j, $j=1,2,\ldots,V$ is the jth element of the gain vector. In these equations we can see that both predictive controllers have a similar structure: the two first terms, ordered from the left, are a discrete *PI controller* and the last term is a weighing contribution of the future open-loop deviations at time $(k+j)t_S$,

$$\Delta\hat{y}^0(j,k)=\hat{y}^0(j,k)-\tilde{y}(k), \quad j=1,2,\ldots,V,J.$$

They only differ in the number of prediction gains employed. So, it is easy to see that we can choose the prediction time J such that both predictive controllers, single-prediction and DMC, have similar performances.

Example 4.1. *To analyze the sensitivity of the proposed controllers to the parameter J we consider a heat exchanger, whose hot outlet temperature is controlled by manipulating the cold stream flow rate, modeled by [13]*

$$Gp(s)=-\frac{35.41}{(4.5s+1)^5}. \tag{25}$$

The discrete model employed to build the single-prediction controller is obtained by assuming a zero-order hold at the input of continuous model (25), a sampling time $t_S=2$ and the convolution length N was fixed in 50 terms. The prediction time is chosen using the stability condition (18), so J must be

$$J\geqslant 12.$$

Fig. 2 shows the responses to a setpoint change for different values of J. As it was anticipated, small values of J give more rapid response and require large initial movements in the control variable (Fig. 3). Observe also that the closed-loop system is stable even for values of J smaller than the limit provided by Eq. (18).

Now, we compare the closed-loop responses provided by single-prediction controllers with that provided by a DMC controller. The DMC control-

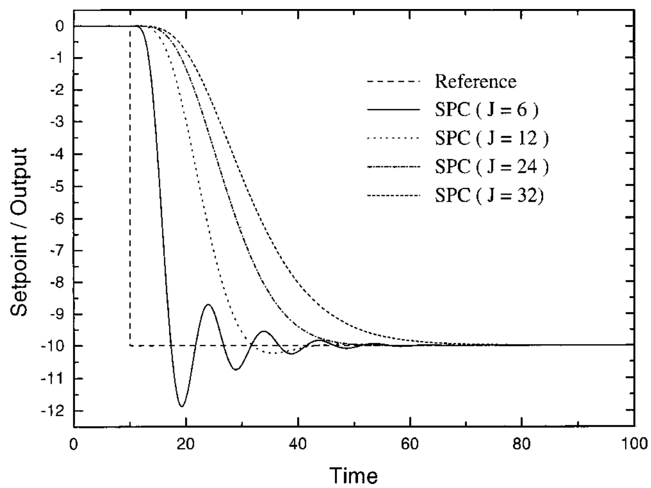

Fig. 2. Closed-loop responses of the linear system to a step change in the setpoint, showing the effect of the prediction time J.

ler was designed following the tune procedure developed by Rahul and Cooper [14]. The plant model (25) was approximated through a first-order plus time delay model,

$$Gp(s) = -35.41\frac{e^{-10.1s}}{12.04s+1},$$

which was discretized assuming a zero-order hold at the input and a sampling time $t_S=2$. *The convolution length is the same that we used to build the single-prediction controller (N=50). The prediction horizon V was set equal to the convolution length (V=N), the control horizon U was fixed to five samples, and the condition number of the controller c was fixed to 500. The control weight* λ *was computed using the following formula [14]:*

$$\lambda = \frac{U}{c}\left(3.5\frac{\tau}{t_S} + 2 - \frac{U-1}{2}\right)\tilde{K}_p = 3.73.$$

Fig. 4 shows that the single-prediction controller can provide a similar performance to that obtained by the DMC controller. This figure also shows that J can be selected such that the performance or the robustness of the system be improved, by reducing or increasing J.

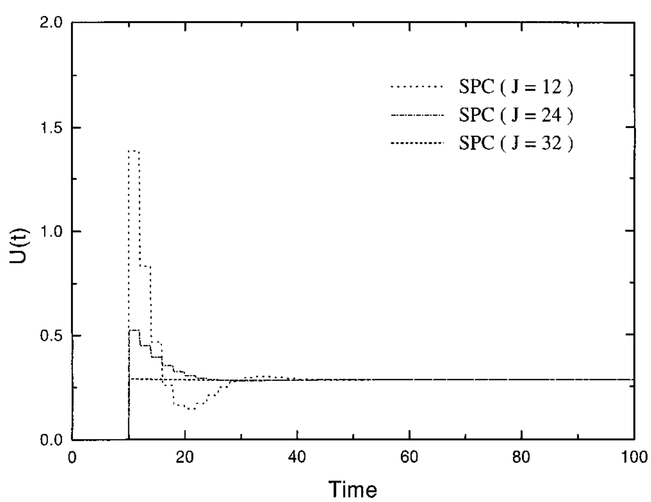

Fig. 3. Control variable correspondent to the closed-loop responses shown in Fig. 2.

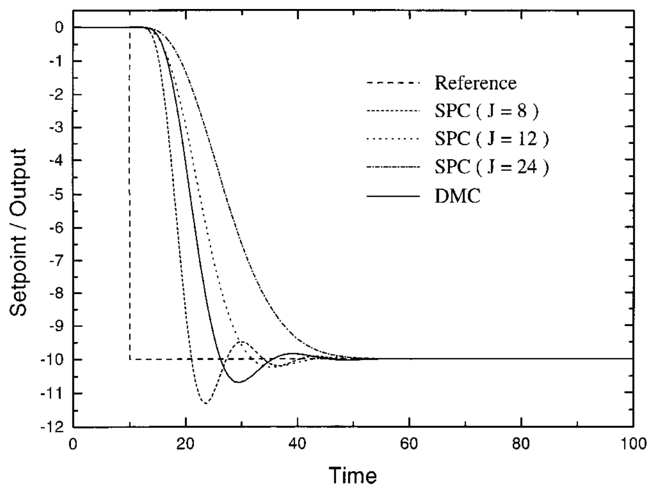

Fig. 4. Closed-loop responses of the linear system to a step change in the setpoint, comparing the response of the DMC and single-prediction controllers.

5. Predictive feedback control

From the analysis of the closed-loop performance in Section 4.2, it is clear that the single-prediction controller provides a similar closed-loop performance to a MPC controller. This fact means that a single-prediction controller shows a poor closed-loop performance when disturbances and uncertainties are present in the system, specially when they are assumed to be time invariant. This is true even when the underlying system is time invariant [15].

A way to solve this problem is to introduce a direct feedback mode in the computation of the control action. This idea can be accomplished by including a filter $F(z)$ in the single-prediction control law (14). The filter not only includes a feedback action in the predictive controller, but also introduces a new set of parameters to allow more demanding performances. Since the control law (14) includes an integral mode, $F(z)$ can take the following form:

$$F(z)=\frac{1}{\tilde{a}_J}\sum_{j=0}^{w} q_j z^{-j}=\sum_{j=0}^{w} q_j^* z^{-j}, \tag{26}$$

where $w \in Z$ is the filter order and $q_j,\ j \in [0,w]$ are the new controller parameters. Since the control law (14) employs only one prediction of the process future behavior, the delay operator z^{-j}, $j=0,1,\ldots,w$, is applied to the time instant at which the prediction is calculated (Fig. 5). Hence the control movement $\Delta u(k)$ is given by

$$\Delta u(k)=\sum_{j=0}^{w} q_j^* \hat{e}^0(J,k-j), \tag{27}$$

where $\hat{e}^0(J,k-j)$ is the **J** step ahead open-loop error computed at time $k-j$. The predictive feedback controller can be derived from Eq. (27), replacing the open-loop error $\hat{e}^0(J,k-j)$ by their components and following a similar procedure to obtain Eq. (15). The result is the controller

$$C(z)=\frac{F(z)}{(1-z^{-1})+F(z)[P(J,z)-\tilde{G}p(z)]}, \tag{28}$$

whose structure is shown in Fig. 6. We must note that $F(z)$ adds additional degrees of freedom to improve the closed-loop performance. This fact makes more difficult the controller design since now it is necessary to tune the filter parameters (the coefficients q_j^*, $j=0,1,\ldots,w$ and the order w). One way to solve this tuning problem is by using a method for a fixed-structure controller like those proposed by Abbas and Sawyer [16] or Harris and Mellichamp [17].

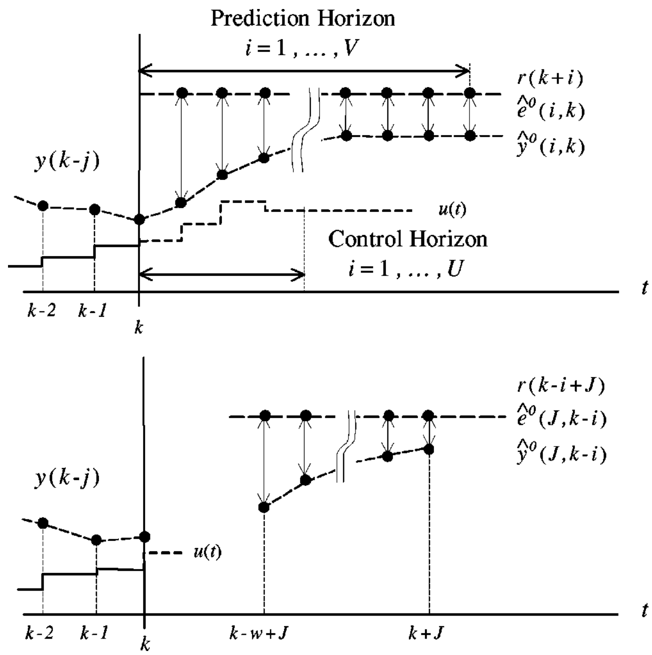

Fig. 5. General MPC and predictive feedback setups.

5.1. Relationship with other control algorithms

It is easy to see that the structure of the predictive feedback controller is a generalization of the internal model control parametrization of the feedback controllers (Fig. 6). Depending on the value of the prediction time and the parameters of the controller we have the different controllers that would be studied in the specialized literature.

When $J=1$ the open-loop predictor $P(J,z)$ becomes the system model $\tilde{G}p(z)$, and the predictive feedback controller (28) is a *reduced order controller* [6],

$$C(z)=\frac{1}{(1-z^{-1})}\sum_{j=0}^{w} q_j^* z^{-j}. \tag{29}$$

When the prediction time is set equal to the time delay ($J_d t_S = t_d$,) the open-loop predictor $P(J,z)$ becomes the system model without time delay, and the predictive feedback controller (28) is the *Smith predictor* of the reduced order controller (29),

$$C(z)=\frac{F(z)}{(1-z^{-1})+F(z)[1-z^{-J_d}]\tilde{G}p(z)}. \tag{30}$$

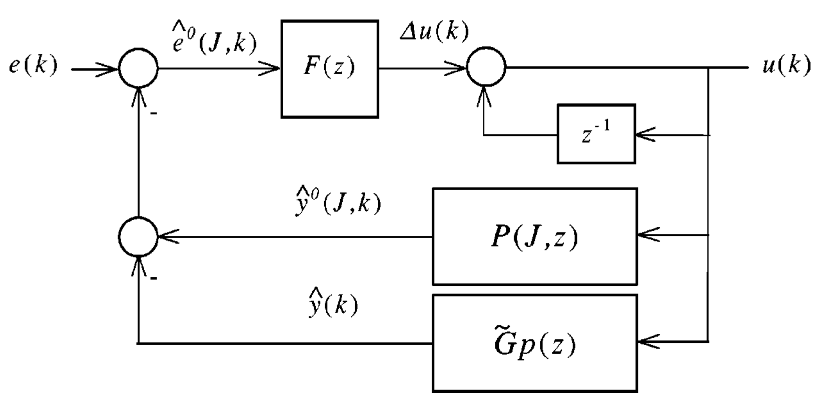

Fig. 6. Structure of the predictive feedback controller.

In this case, the open-loop predictor only compensates the time delay present in the system. In the case of $J = J_d + 1$ and the parameters of the filter are free to be tuned, the predictive feedback controller becomes the *analytical predictor algorithm* [5].

When the prediction time is greater than the time delay ($J_d < J < N$), $w = 0$, and $q_0^* = \tilde{a}_J^{-1}$ the resulting controller is the *single-prediction controller*,

$$C(z) = \frac{1}{\tilde{a}_J + P(J,z) - \tilde{G}p(z)}, \tag{31}$$

which was studied in the previous sections. The character of this controller is governed by the prediction time J, which directly influences the speed of closed-loop response. For the particular choice of the prediction time $J = N$, we can derive a family of predictive controllers whose main characteristic is to obtain a closed-loop response that is at least as good as the normalized open-loop response. If no other design condition is demanded, the controller (23) becomes the *predictor controller* [6],

$$C(z) = \frac{1}{\tilde{a}_N - \tilde{G}p(z)}. \tag{32}$$

Fixing the parameter of the controller $q_0 = \alpha \tilde{a}_N^{-1}$, $\alpha \geqslant 1$, we obtain *simplified model predictive controller* [7],

$$C(z) = \frac{\alpha}{\tilde{a}_N - \alpha \tilde{G}p(z)}. \tag{33}$$

The parameter α provides a way to speed up the closed-loop response and build a dead-time compensation in the controller, but it does not give offset-free responses in the presence of modeling errors. To solve this problem, Chawla *et al.* [18] proposed the inclusion of a first-order filter into the control law such that the resultant controller is the *conservative model based controller*,

$$C(z) = \frac{(1-\beta z^{-1})}{(1-\beta)\tilde{a}_N - (1-\beta z^{-1})\tilde{G}p(z)}, \quad 0 \leqslant \beta < 1. \tag{34}$$

This controller is derived from the predictive feedback controller (28) by fixing its parameters to

$$w = 1, \;\; q_0 = \frac{1}{1-\beta}\tilde{a}_N^{-1}, \;\; \text{and} \;\; q_1 = \frac{\beta}{1-\beta}\tilde{a}_N^{-1}.$$

Observe that varying the parameters α and β governs the character of these controllers influencing the speed of closed-loop response. When the parameters are in the lower limit ($\beta = 0$ and $\alpha = 1$), the controllers (33) and (34) become the predictor controller (32), obtaining the open-loop response. In the other case ($\beta = 1$ and $\alpha = \infty$) the controllers (34) and (33) become the inverse of the system model, obtaining the minimum time response. Varying the parameters between these limits we modify the characteristics of the closed-loop response, speeding up or slowing down the response, in similar way as the single-prediction controller with the prediction time J.

Finally, we can see that predictive feedback control has a strong connection and significant difference from VHP [8]. The approach employed by both frameworks is similar. They employ one prediction of the system output, which can be freely chosen, and use the predicted error as inputs into the controller. However, VHP by itself is a predictive structure, not a controller, that can be added to another control algorithm (for example PI, PID, or simplified predictive control). It is employed to compensates time delay and interactions and provide a built-in feedforward scheme. In this way, VHP is similar to a single-prediction controller. Furthermore, VHP computes the whole trajectory, like standard predictive control, and then selects one prediction.

6. Predictive feedback properties

6.1. Stability analysis

Now, we study the effect of the filter and prediction time on the closed-loop stability. Then, we substitute the predictive feedback controller (28) in the characteristic closed-loop equation, which becomes

$$T(z^{-1}) = (1 - z^{-1}) + F(z^{-1})P(J,z^{-1}) + F(z^{-1})[Gp(z^{-1}) - \tilde{G}p(z^{-1})]. \tag{35}$$

Combining the transfer function of the predictor (A8) and using the discrete convolution, the characteristic equation $T(z^{-1})$ can be written as follows:

$$(1-z^{-1})+\sum_{j=0}^{w} q_j^* \tilde{a}_J z^{-j-1} + \sum_{j=0}^{w} q_j^* \sum_{i=J+1}^{N} \tilde{h}_i z^{J-i-j} + \sum_{j=0}^{w} q_j^* \sum_{i=1}^{N} (h_i-\tilde{h}_i) z^{-i-j} + \sum_{j=0}^{w} q_j^* \sum_{i=N+1}^{\infty} h_i z^{-i-j}=0. \quad (36)$$

The stability of the closed-loop system depends on both the prediction time J and the filter parameters. So, it may be tested by any usual stability criteria. Using the same procedure as in Appendix B, we can derive the following stability condition:

$$\sum_{i=J+1}^{N} |\tilde{h}_i| + \sum_{i=1}^{N} |h_i-\tilde{h}_i| + \left| \sum_{i=N+1}^{\infty} h_i \right| < \tilde{a}_J .$$

Note that this condition is the same as that derived for the single-prediction controller [Eq. (17)]. However, we should note that the parameters of the filter q_j^*, $j=0,1,\ldots,w$ affect the closed-loop stability. These facts look contradictory, because it is clear from the characteristic Eq. (35) that the closed-loop stability depends simultaneously on both. However, these results can be interpreted as follows: the closed-loop stability for the predictive feedback controller is obtained by the independent selection of the prediction time J and the parameters of the filter, such that they independently guarantee the closed-loop stability. These facts mean that the prediction time J should be selected like the single-prediction controller [Eqs. (17) to (19)], and the filter must be tuned as there is no time delay in the system, because it has been compensated by the open-loop predictor.

Since the prediction time J can be fixed independently of the filter's parameters, we can vary it such that the closed-loop performance is improved. Varying J we modify the closed-loop settling time, speeding up or slowing down the system response. So, if we have to control a nonlinear system we can choose a different J for each operating region so that we obtain a similar closed-loop response for each one of them. Then, during the operation, we vary J according to the operating region controlled at each sample.

6.2. Performance analysis

Finally, we analyze the effect of the filter over the overall closed-loop performance and compare the predictive feedback controller with a standard MPC controller, such as the DMC. To carry out this analysis we compare the control action computed by both predictive controllers.

The control actions generated by the predictive feedback control law (27) are obtained by replacing the open-loop error $\hat{e}^0(J,k-j)$ by their components, assuming a step change in the setpoint, and combining with the predicted output (8), the result is

$$u(k)=u(k-1)+\sum_{j=0}^{w} q_j^* e(k-j) + \sum_{j=0}^{w} q_j^* [P(J,z)-\tilde{G}p(z)]u(k-j). \quad (37)$$

In this equation we must observe that the last term is a weighing contribution of the future open-loop deviations at time $(k-j+J)T_S$,

$$\Delta\hat{y}^0(J,k-j)=\hat{y}^0(J,k-j)-\tilde{y}(k-j).$$

It only depends on the past control actions and the system model, so it states the effect of the past control actions on the future behavior of the system. This fact implies that it has significant influence on the closed-loop performance when we have to track the setpoint. However, this term has a negligible influence when we have to reject a disturbance, because it has little information about the disturbance. The two first terms of Eq. (37), ordered from the left, are the time implementation of a *reduced order controller* of order w [6], and commands the system behavior during the disturbance rejection. It is clear now that the control law (37) includes a direct feedback action based on measured error.

Now, we compare the performance achieved by the predictive feedback with that of an MPC con-

Table 1
Predictive controller parameters and results.

Controller parameters	f_{Set}	f_{Dist}
$N=60$, $V=54$, $U=4$, $\lambda=0.14$	7.7164	3.0323
$w=1$; $q=[0.7280\ \ 0.0987]$	7.7061	2.5029
$w=2$; $q=[0.7468\ \ 0.0825\ \ 0.0902]$	7.6938	2.4595
$w=3$; $q=[0.7614\ \ 0.0979\ \ 0.0792\ \ 0.0816]$	7.7005	2.4229

troller. Recalling Eq. (24) we have the control action generated by a standard MPC controller,

$$u(k)=u(k-1)+\left\{\sum_{j=1}^{V} k_j\right\}e(k) - \sum_{j=1}^{V} k_j[P(j,z)-\tilde{G}p(z)]u(k-1). \tag{38}$$

In this equation we can see that MPC controllers only use the last measured error and the two first terms, ordered from the left, are a discrete PI controller. Like the predictive feedback controller, the last term is a weighing contribution of the future open-loop deviations at time $(k+j)t_S$, $j=1,2,\ldots,V$.

Comparing Eqs. (37) and (38) we can see that predictive feedback controller uses more feedback information than standard predictive controllers to compute the control actions. Therefore the predictive feedback controller reduces the effect of disturbances more aggressively than any standard MPC controller, and has better performance than MPC, especially for disturbance rejection problem or important uncertainties present in the system.

A final comment about the predictive feedback can be made. From Eq. (37) it is easy to see that a predictive feedback controller combines the capacity of the predictive control algorithm for good setpoint tracking and time delay compensation, with the classical use of the feedback information to improve the disturbance rejection. Furthermore, for controlling a nonlinear system the prediction time J can be varied to improve the closed-loop performance by modifying the closed-loop settling time.

Example 6.1. *In this example we compare the performances achieved by a predictive feedback controller and a standard MPC controller for setpoint tracking and disturbance rejection. To evaluate the closed-loop responses we consider the following linear plant*:

$$Gp(s)=\frac{e^{-50s}}{(150s+1)(25s+1)},$$

which was used by Rahul and Cooper to evaluate a tuning procedure for a DMC controller. The discrete transfer function is obtained by assuming a zero-order hold at the input and a sampling time $t_S=16$. *The tuning parameters for the DMC controller are same as those using by Rahul and Cooper in his work [14] (see Table 1).*

The predictive feedback controller was designed by solving the basic tuning problem proposed by Giovanini and Marchetti [19] with the objective function

$$f=\sum_{k=1}^{N} e^2(k)+\lambda\Delta u^2(k), \tag{39}$$

for given filter orders and a prediction time ($J=5$). *The order of the filter as chosen such that the resulting controllers include the predictive version of popular PI and PID controllers* ($w=1,2,3$). *The problem described to this point has a fast solution using an algorithm based on the descendent gradient method. The filter parameters obtained for each controller are shown in Table 1.*

The results obtained in the simulations are summarized in Figs. 7 and 8. They show the performance achieved by the predictive controllers normalized to DMC performance. It has been computed as

$$\rho_x=100\,\frac{f_{DMC\,x}-f_{PF\,x}}{f_{DMC\,x}}, \quad x=Set,Dist,$$

where $f_{DMC\,x}, f_{PF,x}$ *is the performance achieved by DMC and predictive feedback controllers for*

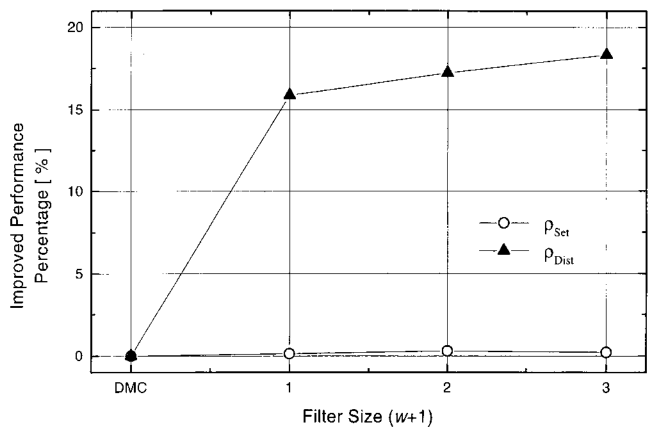

Fig. 7. Performance comparison between DMC and predictive feedback, using index (39).

setpoint tracking and disturbance rejection. Fig. 7 shows the normalized performance measured with Eq. (39), while Fig. 8 shows the normalized performance measured with the ISE index. The predictive feedback controller exhibits a similar performance to the DMC controller (Fig. 7) for the setpoint tracking. Although a better response is obtained by the predictive feedback controller (Fig. 8), it uses more control energy than DMC (Fig. 7). On the other hand, the predictive feedback controller shows an important improvement in the closed-loop performance for disturbance rejection. A better response is again obtained by predictive feedback controller (Fig. 8) and the controller uses a similar control energy than the DMC (Fig. 7).

7. Simulations and results

Let us consider the problem of controlling a continuously stirred tank reactor (CSTR) in which an irreversible exothermic reaction is carried out at constant volume (Appendix C). This is a nonlinear system previously used by Morningred *et al.* [20] for testing predictive control algorithms. Fig. 9 shows the dynamic responses to the following sequence of changes in the manipulated variable q_c: $+10$, -10, -10, and $+10\ \mathrm{l\,min^{-1}}$, where the nonlinear nature of the system is apparent.

Four continuous linear models are determined using least-squared procedures to adjust the com-

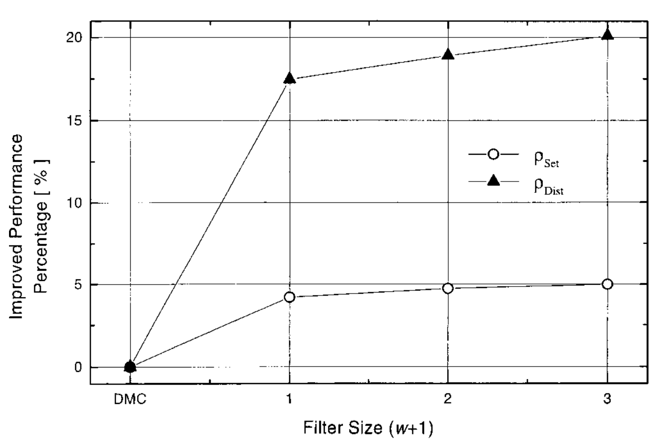

Fig. 8. Performance comparison between DMC and predictive feedback using the ISE index.

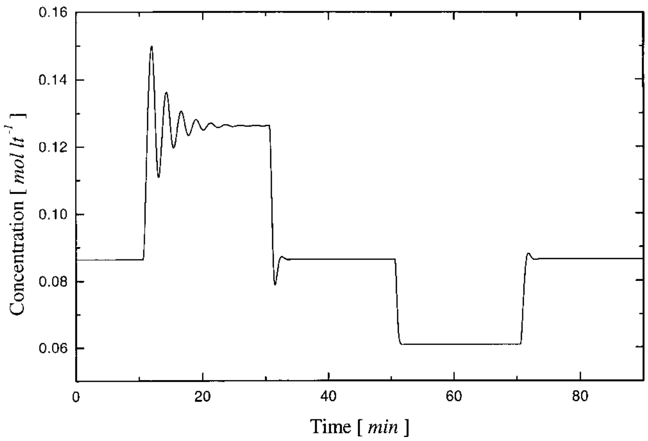

Fig. 9. Open-loop responses of the CSTR concentration to step changes in the coolant flow rate $q_C(t)$.

position responses to the above four step changes in the manipulated variable. Notice that those changes imply three different operating points corresponding to the following stationary manipulated flow rates: 100, 110, and 90 $\mathrm{l\,min^{-1}}$. Table 2 shows the four process transfer functions obtained,

$$\frac{Ca(s)}{q_C(s)} = G_{P_l}(s), \quad l = 1-4.$$

They define the polytopic model associated to the nonlinear behavior in the operating region being considered.

Like in Morningred's work, the sampling time period was fixed at 0.1 min, which gives about four sampled-data points in the dominant time constant when the reactor is operating in the high concentration region. Then, four discrete linear models are used for representing the nonlinear reactor using the polytope idea. These models are obtained by $\mathcal{Z}$ transforming the continuous transfer functions and assuming a zero-order-hold device is included. This representation should be associated to the M vertex models in the tuning problem formulation.

In this application we stress the fact that the reactor operation becomes very sensitive once the manipulation exceeds 113 $\mathrm{l\,min^{-1}}$. Hence, assuming that a hard constraint is physically used on the coolant flow rate at 110 $\mathrm{l\,min^{-1}}$, an additional restriction for the more sensitive model (model 1 in Table 2) must be considered for the deviation variable $u(k)$:

$$u_1(k+i) \leqslant 10, \quad 0 \leqslant i \leqslant V. \tag{40}$$

Besides, a zero-offset steady-state response is demanded, then we include the following constraint:

Table 2
Vertices of the polytope model.

Change	Model obtained
Model 1 $q_C = 100,\ \Delta q_C = 10$	$G_{P_1}(s) = \dfrac{-0.0008t^3 + 0.033\ s^2 - 0.018s + 0.67}{s^4 + 1.92\ s^3 + 30.35\ s^2 + 21.49\ s + 153.7} e^{-0.5\ s}$
Model 2 $q_C = 110,\ \Delta q_C = -10$	$G_{P_2}(s) = \dfrac{-1.3\ 10^{-5}\ s^3 + 0.0065\ s^2 + 0.354\ s + 3.35}{s^4 + 10.5\ s^3 + 101.37\ s^2 + 334.89\ s + 834.6} e^{-0.5\ s}$
Model 3 $q_C = 100,\ \Delta q_C = -10$	$G_{P_3}(s) = \dfrac{6.7\ 10^{-6}\ s^3 - 0.0055\ s^2 + 0.652\ s + 9.35}{s^4 + 28.45\ s^3 + 324.67\ s^2 + 1737.15\ s + 3718.6} e^{-0.5\ s}$
Model 4 $q_C = 90,\ \Delta q_C = 10$	$G_{P_4}(s) = \dfrac{-1.07\ 10^{-4}\ s^3 + 0.0256\ s^2 + 0.143\ s + 0.457}{s^4 + 9.58\ s^3 + 49.69\ s^2 + 128.05\ s + 178.4} e^{-0.5\ s}$

Table 3
Nominal CSTR parameter values.

Parameter	Nomenclature	Value
Measured concentration	Ca	0.1 mol l^{-1}
Reactor temperature	T	438.5 K
Coolant flow rate	q_C	103.41 l min^{-1}
Process flow rate	q	100 l min^{-1}
Feed concentration	Ca_O	1 mol l^{-1}
Feed temperature	T_O	350 K
Inlet coolant temperature	T_{CO}	350 K
CSTR volume	V	100 l
Heat-tranfer term	hA	7.0×10^5 cal min^{-1} K^{-1}
Reaction rate constant	k_0	7.2×10^{10} min^{-1}
Activation energy	E/R	1.0×10^4 K^{-1}
Heat of reaction	ΔH	-2.0×10^5 cal mol^{-1}
Liquid densities	ρ,ρ_c	1.0×10^3 g l^{-1}
Specific heats	C_p, C_{pc}	1.0 cal g K^{-1}

$$\begin{aligned} y(k) &\leqslant 1.05\, r(k), \quad 0 \leqslant k \leqslant N, \\ |e(k)| &\leqslant 5.0\, 10^{-4}, \quad 50 \leqslant k \leqslant N. \end{aligned} \tag{41}$$

This assumes that the nominal absolute value for the manipulation is around 100 l min^{-1} and that the operation is kept inside the polytope whose vertices are defined by the linear models.

Now, we define the parameters of the predictive feedback controller to tune the filter parameters using the method proposed by Giovanini and Marchetti [19]. The filter size w is arbitrarily adopted $(w=3)$ and the predictor of the controller $P(J,z)$ is built using the nonlinear model (C1), assuming the nominal parameters value (Table 3). An adaptive Runge-Kutta integration scheme is used to compute the system output prediction, with the initial state given by $X(k)=[Ca(k)\ T(k-5)]$ at each sample. Since the inlet coolant temperature $T_{CO}(t)$ is measurable, we include it in the predictor to improve the rejection to this disturbance.

Because the controller predictor is built using the nonlinear model, it is reasonable to assume that there are no uncertainties. Therefore we build the open-loop predictor of the tuning problem [19] using the same model as that used to simulated the system output. Finally, we choose the prediction time J. It is fixed such that the robust stability of the system is guaranteed. The prediction time satisfies

$$J_{stb} = \max(J_1; J_2; J_3; J_4) = 10. \tag{42}$$

Notice in this case that it is the polytope that must be shaped along the time being considered. Hence the objective function necessary for driving the adjustment must consider all the linear models simultaneously. At a given time instant and operating point, there is no clear information about which model is the convenient one for representing the process. This is because it depends not only on the operating point but also on which direction the manipulated variable is going to move. The simpler way to solve this by proposing

$$f = \sum_{l=1}^{M} \sum_{k=1}^{V} \gamma_l [e_l(k)^2 + \rho \Delta u_l(k)^2], \tag{43}$$

where the time span is defined by $V=200$. The control weight ρ was fixed in a value such that the control energy has a similar effect as errors in the tuning process $(\rho=0.01)$. Since in this application we found no reason to differentiate the models, we adopt $\gamma_l=1,\ l\in[1,M]$. The problem described to this point has a rapid numerical solution using an algorithm based on the gradient method. The parameters obtained are the following:

$$\begin{aligned} &q_0 = 0.2313; \quad q_1 = -0.1550; \\ &q_2 = -0.2531; \quad q_3 = 0.2157. \end{aligned} \tag{44}$$

Morningred *et al.* [20] have previously worked with this reactor model for testing different alternatives of predictive controllers and confronted the results with the responses obtained using a PI controller whose parameters were adjusted by the ITAE criterion; thus we used the same settings: the gain value 52 l^2 mol^{-1} min^{-1} and the integration time constant 0.46 min. The simulation tests are also similar to Morningred's work and consists of a sequence of step changes in the reference value and a sequence of load changes in the feed stream concentration and the refrigerant inlet temperature.

Fig. 10 shows the results obtained when comparing the discrete controller with the mentioned PI. The setpoint was changed in intervals of 10 min. from 0.09 to 0.125 mol l^{-1}, returns to 0.09, then steps to 0.055 and returns to 0.09 mol l^{-1}.

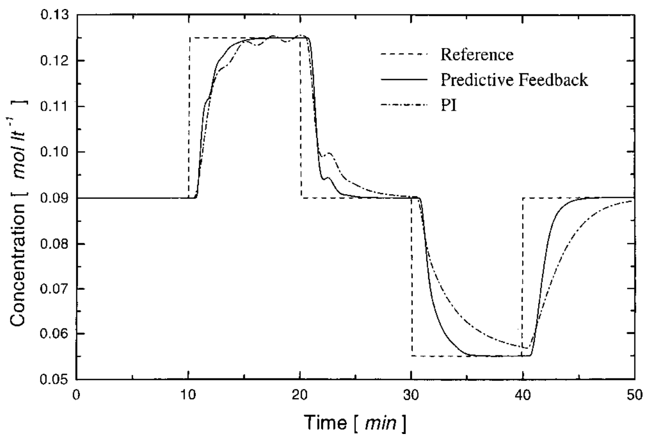

Fig. 10. Closed-loop responses of the CSTR concentration to a sequence of step changes in the setpoint using the predictive controller with one-side constraint in the manipulation and a PI controller adjusted with ITAE.

The superior performance of the discrete controller is obtained through a vigorous initial movement in the manipulated variable, which, however, does not overcome the 110 $\mathrm{l\,min^{-1}}$ limit as shown in Fig. 11, but shows more movements than the PI.

Fig. 12 shows the results obtained when comparing the discrete controller with the mentioned PI under load changes. For testing the disturbance rejection the following sequence of changes are made: first the feed stream concentration $Ca_O(t)$ changes from 1 to 1.05 $\mathrm{mol\,l^{-1}}$ and 10 min later the refrigerant temperature $T_{CO}(t)$ goes down 10 °C; then $Ca_O(t)$ and $T_{CO}(t)$ return to the original values, with a 10-min difference between them. A better disturbance rejection capability is observed in the discrete controller suggested in this paper.

Fig. 13 shows the manipulated movements corresponding to the repsonses in Fig. 12. In this figure we see that the excursion of $q_C(t)$ is more important in the case of the PI, but smoother.

8. Conclusions

A new method to design predictive controllers for linear SISO systems has been presented in this work. It uses only one prediction of the system output J time intervals ahead to compute the correspondent future error. Then, the predictive feedback controller is defined by introducing a filter

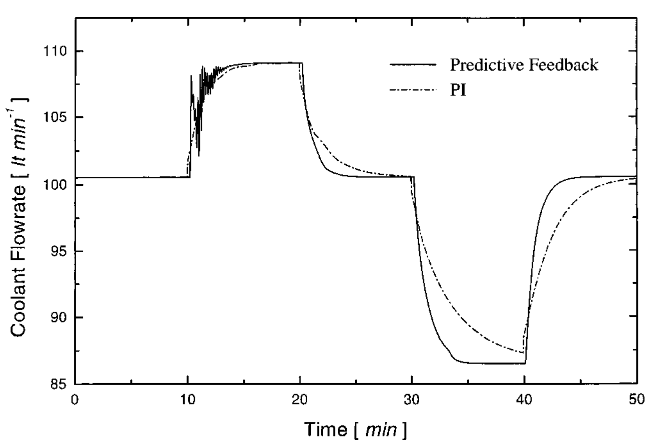

Fig. 11. Manipulated movements corresponding to the responses in Fig. 10.

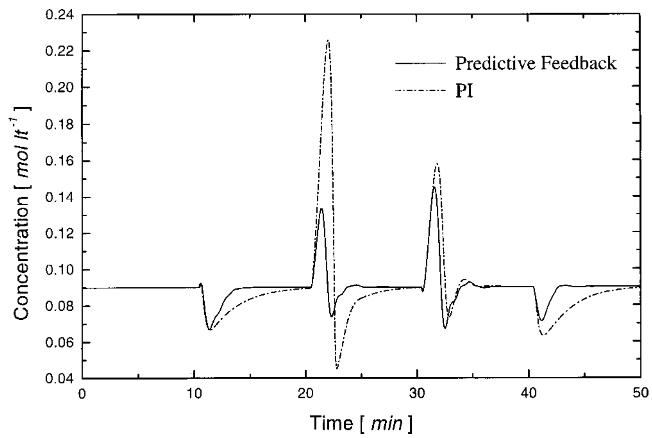

Fig. 12. Closed-loop responses of the CSTR concentration to a sequence of load changes using the predictive controllers.

which weights the last predicted errors. In this way, the resulting control action is computed by observing the system future behavior and the present and past errors. These features enable the predictive feedback controller to combine the capacity of predictive control algorithm for good set-point tracking and time delay compensation, with the classical use of the feedback information to improve the disturbance rejection.

The character of these controllers is governed by one parameter, the prediction time, which directly influences the speed of closed-loop response. Some simple criteria for its selection are provided: they guarantee the robust stability of the closed-loop system. The predictive feedback controller has additional tuning parameters: the parameters of the filter. Robust stability and closed-loop performance issues of these controllers have been analyzed. An extensive analysis of closed-loop performance, compared with standard MPC controllers, have been also carried out.

In spite of the results obtained in his work, several questions about extension to multivariable systems and how we can address on-line constraints in the input and output variables still remain open as future research topics. A future work can also include an on-line tuning of the predictive feedback parameters such that the performance remains optimal and the constraints would be fullfilled for every sample.

Appendix A: Predictor transfer function

Eq. (5) defines a realizable J-step ahead predictor in the discrete time domain. The predictor is

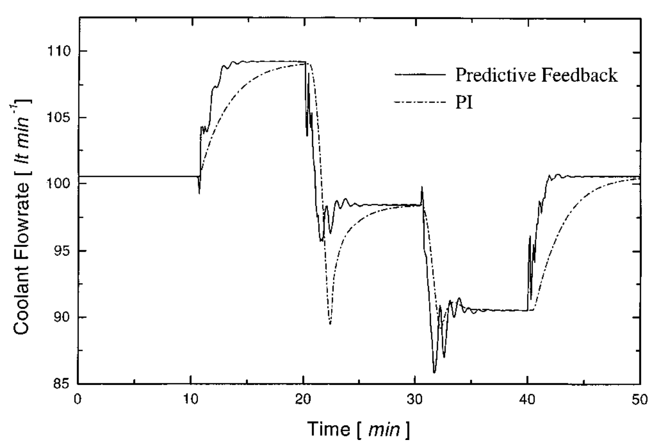

Fig. 13. Manipulated movements corresponding to the responses in Fig. 12.

realizable since only past inputs to the system are used to computed the future behavior. Expanding this equation results in

$$\hat{y}^0(J,k)=\hat{y}(0,k)+\sum_{i=2}^{N}\tilde{h}_i\Delta u(k+1-i)+\cdots+\sum_{i=J+1}^{N}\tilde{h}_i\Delta u(k+J-i),$$

and taking the $\mathcal{Z}$ transform gives

$$\hat{y}^0(J,z)=\hat{y}(z)+\left[\sum_{i=2}^{N}\tilde{h}_i z^{1-i}+\cdots+\sum_{i=J+1}^{N}\tilde{h}_i z^{J-i}\right]\times(1-z^{-1})u(z). \quad (A1)$$

Defining the following function:

$$H(J,z)=\begin{cases}\sum_{i=J+1}^{N}\tilde{h}_i z^{J-i}, & 0\leqslant J\leqslant N-1\\ 0, & J=N,\end{cases} \quad (A2)$$

Eq. (A1) can be written as

$$\hat{y}^0(J,z)=\hat{y}(z)+[H(1,z)+\cdots+H(J,z)]\times(1-z^{-1})u(z). \quad (A3)$$

Note that there is a recursive relationship,

$$H(m,z)=\tilde{h}_{m+1}z^{-1}+H(m+1,z)z^{-1}. \quad (A4)$$

Then, combining Eqs. (A3) and (A4) and rearranging gives

$$\hat{y}^0(J,z)=\hat{y}(z)+\left[\sum_{i=2}^{N}\tilde{h}_i z^{-1}+H(J,z)-H(1,z)z^{-1}\right]u(z). \quad (A5)$$

Adding and substracting $\tilde{h}_1 z^{-1}$ and operating gives

$$\hat{y}^0(J,z)=\hat{y}(z)+\left[\tilde{h}_1 z^{-1}+\sum_{i=2}^{J}\tilde{h}_i z^{-1}+H(J,z)-(\tilde{h}_1 z^{-1}+H(1,z)z^{-1})\right]u(z), \quad (A6)$$

where we can identify the **J**th coefficient of step response,

$$\tilde{a}_J z^{-1}=\tilde{h}_1 z^{-1}+\sum_{i=2}^{J}\tilde{h}_i z^{-1},$$

and the plant model

$$H(0,z)=\tilde{h}_1 z^{-1}+H(1,z)z^{-1}=\tilde{G}p(z).$$

Hence the expression (A6) can be written

$$\hat{y}^0(J,z)=\hat{y}(z)+[\tilde{a}_J z^{-1}+H(J,z)-\tilde{G}p(z)]u(z), \quad (A7)$$

and, since

$$\hat{y}(z)=\tilde{G}p(z)u(z),$$

the $\mathcal{Z}$ transform of the **J**-step ahead single predictor is given by

$$P(J,z)=\tilde{a}_J z^{-1}+H(J,z), \quad (A8)$$

i.e.,

$$\hat{y}^0(J,z)=P(J,z)u(z). \quad (A9)$$

Appendix B: Robust stability condition for single-predictive controller

The characteristic closed-loop equation $T(z^{-1})$ for the single-prediction controller is given by

$$T(z^{-1})=\tilde{a}_J+H(J,z^{-1})+Gp(z^{-1})-\tilde{G}p(z^{-1}). \quad (B1)$$

Using the discrete convolution and Eq. (A2), the last expression can be written in the following way:

$$T(z^{-1})=\tilde{a}_J+\sum_{i=J+1}^{N}\tilde{h}_i z^{J-i}+\sum_{i=1}^{N}(h_i-\tilde{h}_i)z^{-i}+\sum_{i=N+1}^{\infty}h_i z^{-i}. \tag{B2}$$

The stability of the closed-loop system depends on the prediction time J and may be tested by any usual stability criteria. First, the following lemma is introduced.

Lemma B.1. *If the polynomial* $T(z^{-1})=\sum_{i=0}^{\infty}t_i z^{-i}$ *has the property that*

$$\inf_{|z|\geqslant 1}|T(z^{-1})|>0,$$

then the related closed-loop system will be asymptoptically stable [18].

Hence

$$|T(z^{-1})|\geqslant|\tilde{a}_J|-\left|\sum_{i=J+1}^{N}\tilde{h}_i z^{J-i}\right|-\left|\sum_{i=1}^{N}(h_i-\tilde{h}_i)z^{-i}\right|-\left|\sum_{i=N+1}^{\infty}h_i z^{-i}\right| \tag{B3a}$$

$$\geqslant|\tilde{a}_J|-\sum_{i=J+1}^{N}|\tilde{h}_i z^{J-i}|-\sum_{i=1}^{N}|(h_i-\tilde{h}_i)z^{-i}|-\left|\sum_{i=N+1}^{\infty}h_i z^{-i}\right|, \tag{B3b}$$

and using lemma 1 gives

$$\inf_{|z|\geqslant 1}|T(z^{-1})|\geqslant\inf_{|z|\geqslant 1}\left\{|\tilde{a}_J|-\sum_{i=J+1}^{N}|\tilde{h}_i z^{J-i}|-\sum_{i=1}^{N}|(h_i-\tilde{h}_i)z^{-i}|-\left|\sum_{i=N+1}^{\infty}h_i z^{-i}\right|\right\}>0. \tag{B4}$$

The worst case happens when $z=1$, thus

$$|\tilde{a}_J|-\sum_{i=J+1}^{N}|\tilde{h}_i|-\sum_{i=1}^{N}|h_i-\tilde{h}_i|-\left|\sum_{i=N+1}^{\infty}h_i\right|>0, \tag{B5}$$

which is equivalent to

$$\sum_{i=J+1}^{N}|\tilde{h}_i|+\sum_{i=1}^{N}|h_i-\tilde{h}_i|+\left|\sum_{i=N+1}^{\infty}h_i\right|<|\tilde{a}_J|. \tag{B6}$$

Appendix C: Nonlinear reactor model

The model of a continuous reactor where an irreversible exothermic reaction takes place has been selected for testing the proposed controller. The reaction is

$$A\rightarrow B,$$

and occurs in a constant volume reactor cooled by a single coolant stream. The operation is modeled by the following equations:

$$\frac{dCa(t+t_d)}{dt}=\frac{q(t)}{V}[Ca_0(t)-Ca(t+k_d)]-k_0Ca(t+t_d)\exp\left[\frac{-E}{RT(t)}\right],$$

$$\frac{dT(t)}{dt}=\frac{q(t)}{V}[T_0(t)-T(t)]-\frac{k_0\Delta H}{\rho c_p}Ca(t+t_d)\exp\left[\frac{-E}{RT(t)}\right]+\frac{\rho_c c_{pc}}{\rho c_p V}q_c(t)\left\{1-\exp\left[\frac{-hA}{q_c(t)\rho_c c_{pc}}\right]\right\}\times[T_{CO}(t)-T(t)].$$

The nominal parameter values for this model appear in Table 3. The objective is to control the measured concentration of reactive A at the outlet stream Ca by manipulating the coolant flow rate $q_C(t)$. The concentration has a measured time delay of $t_d=0.5$ min. The nonlinear characteristics are clearly appreciated in Fig. 9 where the responses to equal amplitude step changes are shown.

References

[1] Clarke, C., Mothadi, D., and Tuffs, P., Generalized predictive control: The basic algorithm. Automatica **24**, 137–160 (1987).

[2] Cutler, C. and Ramaker, B., Dynamic matrix control—A computer algorithm. In Proc. Join Autom. Control Conf., San Francisco, 1980.

[3] Richalet, A., Rault, J., Testud, J., and Papon, J., Model predictive heuristic control: Applications to industrial processes. Automatica **14**, 413–428 (1978).

[4] Shah, S., Model-Based Predictive Control: Theory and Implementation Issues. ADCHEM'94, Kyoto Research Park, Kyoto, Japan, 1994.

[5] Moore, C., Selected problems in the design and implementation of direct digital control, Ph.D. thesis, Department of Chemical Engineering, Louisiana State University, 1969.

[6] Isermann, R., Digital Control Systems. Springer-Verlag, Berlin, 1981.

[7] Arulalan, G. and Deshpande, P., Simplified model predictive control. Ind. Eng. Chem. Res. **26**, 347–356 (1987).

[8] Wojsznis, W., Variable Horizon Predictor. Proceedings of the 33rd Conference on Decision and Control, Lake Buena Vista, Florida, December, 1994.

[9] García, C., Prett, D., and Morari, M., Model predictive control: Theory and practice—a survey. Automatica **25**, 335–348 (1989).

[10] Desoer, C. and Vidyasagar, M., Feedback System: Input-Output Properties. Academic Press, New York, 1975.

[11] Prasad, G., Swidenbank, E., and Hogg, B., A local model network based multivariable long-range predictive control strategy for thermal power plants. Automatica **34**, 1185–1204 (1998).

[12] Morari, M. and Zafiriou, E., Robust Process Control, Englewood Cliffs, NJ, 1989.

[13] Giovanini, L., Robust predictive control with constrains. Ph.D. thesis, School of Water Resource Engineering, Universidad Nacional del Litoral, Argentina, 2000.

[14] Rahul, S. and Cooper, D., A tuning strategy for unconstrained SISO model predictive control. Ind. Eng. Chem. Res. **36**, 729–746 (1997).

[15] Lee, J. and Yu, Z., Worst-case formulation of model predictive control for system with bounded parameters. Automatica **33**, 763–781 (1997).

[16] Abbas, A. and Sawyer, P., A multiobjective design algorithm: Application to the design of SISO control systems. Comput. Chem. Eng. **19**, 241–248 (1995).

[17] Harris, S. and Mellichamp, D., Controller tuning using optimization to meet multiple closed-loop criteria. AIChE J. **31**, 484–487 (1985).

[18] Chawla, V., Prasad, P., and Deshpande, P., New digital control algorithm applied to SISO systems. Hydrocarbon Process. **8**, 59–63 (1989).

[19] Giovanini, L. and Marchetti, J., Shaping time-domain response with discrete controllers. Ind. Eng. Chem. Res. **38**, 4777–4789 (1999).

[20] Morningred, J., Paden, B., Seborg, D., and Mellichamp, D., An adaptive nonlinear predictive controller. Chem. Eng. Sci. **47**, 755–765 (1992).

Leonardo L. Giovanini was born in Argentina in 1969. He received the B.S. degree in electronic engineering from the Universidad Tecnologica Nacional (UTN), Regional School at Villa Maria in 1995, and the Ph.D. degree in engineering science from Universidad Nacional del Litoral, School of Water Resource in 2000. Since then he has been a lecturer in the Department of Electronic Engineering at UTN, Regional School at Villa Maria. Actually, he is working as a research fellow in the Industrial Control Center, University of Strathclyde. His research interests include predictive control for nonlinear, fault tolerant, and distributed control systems.