Int. J. Appl. Math. Comput. Sci., 2008, Vol. 18, No. 2, 229–239
DOI: 10.2478/v10006-008-0021-z

EXTENSION OF FIRST ORDER PREDICTIVE FUNCTIONAL CONTROLLERS TO HANDLE HIGHER ORDER INTERNAL MODELS

MOHAMED TAREK KHADIR *, JOHN V. RINGWOOD **

* Department of Computer Science
University Badji Mokhtar, Annaba, Algeria
e-mail: Khadir@lri-annaba.net

** Department of Electronic Engineering
National University of Ireland, Maynooth, Co. Kildare, Ireland
e-mail: john.ringwood@eeng.nuim.ie

Predictive Functional Control (PFC), belonging to the family of predictive control techniques, has been demonstrated as a powerful algorithm for controlling process plants. The input/output PFC formulation has been a particularly attractive paradigm for industrial processes, with a combination of simplicity and effectiveness. Though its use of a lag plus delay ARX/ARMAX model is justified in many applications, there exists a range of process types which may present difficulties, leading to chattering and/or instability. In this paper, instability of first order PFC is addressed, and solutions to handle higher order and difficult systems are proposed. The input/output PFC formulation is extended to cover the cases of internal models with zero and/or higher order pole dynamics in an ARX/ARMAX form, via a parallel and cascaded model decomposition. Finally, a generic form of PFC, based on elementary outputs, is proposed to handle a wider range of higher order oscillatory and non-minimum phase systems. The range of solutions presented are supported by appropriate examples.

Keywords: Model predictive control, predictive functional control, non-minimum phase systems, oscillatory systems.

1. Introduction

Model predictive control, an advanced control approach (Tatjewski, 2007), grew rapidly in popularity and its field of application diversified substantially since its first applications in the refining and petrochemical industry in 1980 (Cutler and Ramaker, 1980; Richalet *et al.,* 1976). It is also reported by Qin and Badgwell (2003) that MPC has been used in over 2,500 industrial applications in the chemical, pulp and paper and food processing industries, from a total of 4,500, aside from the traditional refining and petrochemical sector.

Although the principles of MPC are universal, and can be found in many textbooks (Maciejowski, 2002; Morari and Lee, 2000; Richalet, 1993), a wide range of MPC algorithm was developed, primarily to suit given types of industrial application. Among the most popular MPC algorithms one can cite:

- Model Predictive Heuristic Control (MPHC), with the original algorithm called IDCOM for identification and control, and HIECON for hierarchical control most suited for large multivariable systems (Richalet *et al.,* 1978).
- Dynamic Matrix Control (DMC), from Cutler and Ramaker (Cutler and Ramaker, 1980).
- Generalised Predictive Control (GPC), (Clarke *et al.,* 1987), and
- Predictive Functional Control (PFC), developed by Richalet and ADERSA (Richalet, 1998; Richalet, 1993).

For single-input/single-output (SISO) systems, a transfer function internal model formulation, as used in GPC and PFC, is more convenient to manipulate than the over-parameterised step response model used in DMC that requires a large number of step response parameters, often truncated for a more efficient computation time (otherwise this model would have an infinite number of parameters). Moreover, input/output representations, e.g., ARX/ARMAX, are preferred to state space formulations for SISO systems with small turnovers as they do not include the notion of state and matrix calculus. This matches the wish of many industries for a transparent and/or well understood design like PID.

PFC can use many forms of internal model, including state space (Rossiter and Richalet, 2001a), input/output (Richalet, 1998), Finite Impulse Response (FIR) (Richalet *et al.,* 1978), fuzzy rules (Skrjank and Matko, 2000), etc. This is an obvious advantage when compared with GPC, which is based on CARIMA type models, and a solution of Diophantine equations (Clarke *et al.,* 1987).

However, one of the main distinguishing features of PFC is the independent (non-realigned) nature of the internal prediction model. In this case, the predictions made depend solely on the process input and model past, present and future outputs, in contrast to other MPC approaches. For instance, GPC and its CARIMA-type model utilise past and present process outputs mixed with futures model outputs as well as process inputs in order to issue a prediction (Clarke *et al.,* 1987).

Industrial vendors ADERSA claim that input-output internal models with mixed outputs from the process and the model (or state space models with an estimator) realign the model state on noisy data (output measurements), hence often giving poor predictions as well as leading to an offset (Rossiter and Richalet, 2001a).

Another important distinguishing feature of PFC is the projection of the manipulated variable on a set of basis functions (Richalet, 1998), e.g. a step input, in the simplest case. It is claimed in (Richalet, 1998) that it is more efficient to structure the Manipulated Variable (MV) that way as

- On the one hand, we limit the number of unknown parameters N_b by projecting the future manipulated variable onto a base of functions UB_j of a smaller dimension than the prediction horizon H.
- On the other hand, the discontinuity and the control frequency range are limited, by limiting the dimension of the basis.

The future MV is then expressed in the following form:

$$u(k+i) = \sum_{j=1}^{N_b} u_j(k) UB_j(i), \qquad (1)$$

where UB_j are the basis functions.

Every input base UB_j implies an output basis SB_j known *a priori* for a given model. For example if we take a polynomial base, the first three basis functions are shown in Fig. 1, where UB_i represent the MV (input of the basis function), with its corresponding output SB_i. Usually a zero order base function (UB_0), representing a step MV change to find at each sample time, is used. This

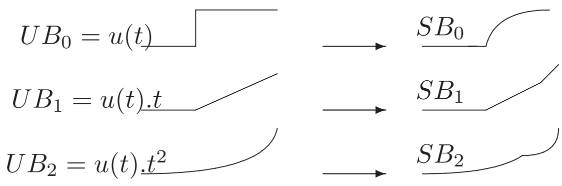


Fig. 1. Input/output basis functions.

gives an exponential (first order) output (SB_0). The concept of projecting the MV onto a functional basis can be found in (Richalet, 1998) and is given the name Predictive **Functional** Control (PFC).

In this paper, the goal is to extend the applicability of the intuitively attractive input/output PFC formulation to a wider range of systems. While first order (with delay) process models are widely and successfully used over a range of application areas, this paper will attempt to provide higher order control solutions while retaining the attractive simplicity of the first order one.

In fact, a number of the proposed extensions rely on the core first order solution. The paper proceeds (in Section 2) by decomposing general SISO ARX and ARMAX models (with real poles) into sets of first order subsystems, using both parallel and cascade forms. Composite PFC solutions for these decomposed systems are developed. In Section 3, some difficulties presented by zero dynamics are highlighted and solution strategies outlined. Section 4 deals with systems with complex poles and/or non-minimum phase dynamics. Two simulation studies are used to illustrate the effectiveness of the developed control solutions, and conclusions are drawn in Section 5.

2. PFC control design

PFC operates on the following four principles (Richalet,1993):

- internal model,
- reference trajectory,
- auto-compensation, and
- calculation of the manipulated variable.

In the case of a higher order process, the internal model needs (ideally) to be of the same order as the process if a plant/model mismatch is to be avoided. Observing the fact that any m-th order system can be decomposed into a *set* of first order blocks may allow a composite controller to be developed, based on a set of first order PFC controllers. Subsections 2.1 and 2.2 document two possible approaches which utilise such a philosophy.

2.1. Internal model in a parallel form. For a high order strictly proper internal process, $G_M(\mathrm{s})$ (2), the trans-

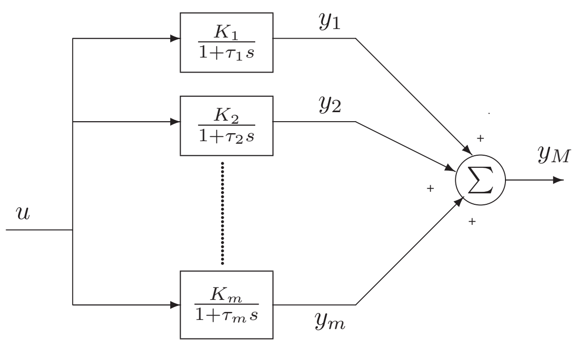


Fig. 2. m-th order parallel model.

fer function representation based on a parallel decomposition is given by

$$G_M(\mathrm{s}) = \sum_{i=1}^{m} \frac{K_i}{1+\tau_i \mathrm{s}}. \tag{2}$$

2.1.1. Output prediction. From Fig. 2, the model output $y_M(k)$ is given by

$$y_M(k) = y_1(k) + y_2(k) + \cdots + y_m(k). \tag{3}$$

The difference equation obtained from a Zero Order Hold (ZOH) equivalent of the model in (3) is given by

$$y_i(k) = \alpha_i y_i(k-1) + K_i(1-\alpha_i)u(k-1), \quad 1 \le i \le m, \tag{4}$$

where

$$\alpha_i = e^{\left(-\frac{T_s}{\tau_i}\right)}, \tag{5}$$

with T_s as the sampling period. Substituting (4) into (3) gives the following model output:

$$\begin{aligned} & y_M(k) \\ & = \alpha_1 y_1(k-1) + \alpha_2 y_2(k-1) + \cdots + \alpha_m y_m(k-1) \\ & \quad + \Big[K_1(1-\alpha_1) + K_2(1-\alpha_2) + \cdots + K_m(1-\alpha_m)\Big] \\ & \quad \times u(k-1) \end{aligned} \tag{6}$$

or, more compactly,

$$\begin{aligned} y_M(k) = & \sum_{i=1}^{m} \alpha_i y_i(k-1) \\ & + \sum_{i=1}^{m} K_i(1-\alpha_i)u(k-1). \end{aligned} \tag{7}$$

The response $y_M(k)$ (cf. (7)) may be then divided into two parts:

$$y_A(k+H) = \sum_{i=1}^{m} \alpha_i^H y_i(k) \tag{8}$$

and

$$y_F(k+H) = \sum_{i=1}^{m} K_i(1-\alpha_i^H)u(k), \tag{9}$$

where $y_A(k+H)$ is the future autoregressive prediction (free response), and $y_F(k+H)$ is the predicted forced response and H the prediction horizon. Note that this is the delay free model, any pure time delay will be compensated into the control formulation in Section 2.1.6.

Only the non-realigned nature of the internal model, inherent to PFC, permits such an easy decomposition (Rossiter, 2001).

2.1.2. Reference trajectory formulation. The future process output is specified by the reference trajectory, initialised on the real process output y_P and the desired setpoint $C(k)$. The reference trajectory used in PFC is generally in an exponential form, given by a function of the setpoint $C(k)$ and the process output $y_P(k)$ as

$$y_R(k+H) = C(k) - \lambda^H(C(k) - y_P(k)), \tag{10}$$

where λ is given by

$$\lambda = e^{\left(-\frac{T_s}{T_R}\right)}, \tag{11}$$

with T_R being the desired Closed Loop Response Time (CLRT) and T_s the sampling time. At the coincidence horizon H, the estimated process output, $\widehat{y}_P$, is set equal to the reference trajectory. We have

$$y_R(k+H) = \widehat{y}_P(k+H), \tag{12}$$

where the process output estimate $\widehat{y}_P$ at time $k+H$ is given by

$$\widehat{y}_P(k+H) = y_M(k+H) + (y_P(k) - y_M(k)). \tag{13}$$

Replacing $y_M(k+H)$ with its expression from (7), we obtain

$$\widehat{y}_P(k+H) = \sum_{i=1}^{m} y_i(k+H) + (y_P(k) - \sum_{i=1}^{m} y_i(k)). \tag{14}$$

2.1.3. Computation of the control law. At the coincidence point, $y_R(k+H) = \widehat{y}_P(k+H)$, and using a step input basis function (Fig. 1) as well as Eqns. (8), (9) and (14), we obtain

$$\begin{aligned} & C(k)(1-\lambda^H) - y_P(k)(1-\lambda^H) + y_1(k)(1-\alpha_1^H) \\ & \quad + y_2(k)(1-\alpha_2^H) + \cdots + y_m(k)(1-\alpha_m^H) \\ & \quad = (K_1(1-\alpha_1^H) + K_2(1-\alpha_2^H) \\ & \qquad + \cdots + K_m(1-\alpha_m^H))u(k). \end{aligned} \tag{15}$$

Solving (15), for $u(k)$ we end up with the control law

$$u(k) = \frac{(C(k) - y_P(k))(1-\lambda^H)}{\sum_{i=1}^{m} K_i(1-\alpha_i^H)} + \frac{\sum_{i=1}^{m} y_i(k)(1-\alpha_i^H)}{\sum_{i=1}^{m} K_i(1-\alpha_i^H)}. \tag{16}$$

2.1.4. Handling added disturbances. For the ARMAX case and the inclusion of a measurable, known or estimated input disturbance v, the disturbance dynamics may appear in the model in a decomposed form as

$$y(s) = \sum_{i=1}^{m} \frac{K_i}{1+\tau_i s} u(s) + \sum_{i=1}^{m} \frac{K_i'}{1+\tau_i s} v(s). \tag{17}$$

Following the developmental steps of Sections 2.1.1 to 2.1.3, the corresponding PFC control law is obtained as

$$\begin{aligned} u(k) = {} & \frac{(C(k) - y_P(k))(1-\lambda^H)}{\sum_{i=1}^{m} K_i(1-\alpha_i^H)} \\ & + \frac{\sum_{i=1}^{m} y_i(k)(1-\alpha_i^H)}{\sum_{i=1}^{m} K_i(1-\alpha_i^H)} - \frac{\sum_{i=1}^{m} K_i'(1-\alpha_i^H)}{\sum_{i=1}^{m} K_i(1-\alpha_i^H)} v(s). \end{aligned} \tag{18}$$

Specifying the input disturbance dynamics in a parallel form is not crucial for the determination of the controller solution, as long as the disturbance is subtracted in a feedforward manner (cf. (18)). However, the choice as in (17) leads to a particularly elegant control solution.

2.1.5. Proper systems. It is also possible to come across proper systems where the orders of the numerator and denominator are equal. In this case we can further decompose the system into a gain plus a sum of first order gain/pole systems:

$$G_M(s) = \frac{y_M(s)}{u(s)} = K_0 + \sum_{i=1}^{m} \frac{K_i}{1+\tau_i s}. \tag{19}$$

The difference equations based on the ZOH equivalent are

$$y_0(k) = K_0 u(k), \tag{20}$$

$$\begin{aligned} y_1(k) = {} & \sum_{i=1}^{m} \alpha_i y_i(k-1) \\ & + \sum_{i=1}^{m} \left(K_i(1-\alpha_i)\right) u(k-1), \end{aligned} \tag{21}$$

$$y_M(k) = y_0(k) + y_1(k). \tag{22}$$

Developing a PFC control law for such a system gives the following analytical solution:

$$\begin{aligned} u(k+1) = {} & \frac{(C(k) - y_P(k))(1-\lambda^H)}{K_0} \\ & + \frac{\sum_{i=1}^{m} y_i(k)(1-\alpha_i) + y_0(k)}{K_0} \\ & + \frac{\sum_{i=1}^{m} K_i(1-\alpha_i)}{K_0} u(k). \end{aligned} \tag{23}$$

The control law (23) may lead to unstable or ringing MVs, depending on the values of the dominant zero(s) of the system. This is investigated fully in Section 3.

2.1.6. Case of a process with a pure time delay. In the linear case, a process with a pure time delay can be expressed in terms of a delay free part, plus delay added at the output, as in Fig. 3. The value $y_{P_{\mathrm{delay}}}$ at time k may be measured, but not y_P. In order to take into account the delay in a control law formulation, prior knowledge of the delay value d is needed. Here y_P can be estimated as

$$y_P(k) = y_{P_{\mathrm{delay}}}(k) + y_M(k) - y_M(k-d). \tag{24}$$

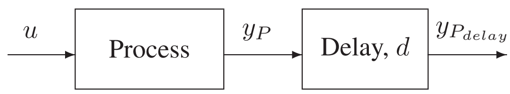


Fig. 3. Process with time delay.

The above modification of $y_P(k)$ can be applied to higher order PFC developments, see (16), (18) and (23), in order to compensate an *a-priori* known pure time delay d.

2.2. Internal model in a cascaded form. Another possibility of decomposing a given model is the one in a cascaded form, cf. Fig. 4. The ZOH equivalent of the intact m-th order $G_M(\mathrm{s})$ subsequently decomposed into cascaded first order blocks is given as (Houpis and Lamont, 1992)

$$G_M(z) = \prod_{i=1}^{m} \frac{K_i(1+\beta_i z^{-1})}{1-\alpha_i z^{-1}}. \tag{25}$$

With respect to the instant k, the model output (in the nominal case $y_m = y_M$) can be determined as

$$y_i(k) = \alpha_i y_i(k-1) + K_i y_{i-1}(k) + K_i \beta_i y_{i-1}(k-1), \quad 2 \le i \le m \tag{26}$$

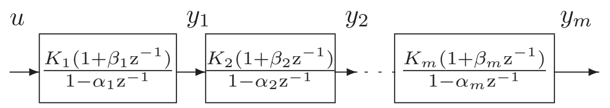


Fig. 4. m-th order cascaded model.

and

$$y_1(k) = \alpha_1 y_1(k-1) + K_1 u(k) + K_1\beta_1 u(k-1). \quad (27)$$

The free (auto-regressive) and the forced responses are given respectively by

$$\begin{aligned} y_A(k+H) = \alpha_m^H y_m(k) &+ K_m\beta_m y_{m-1}(k-1) \\ &+ K_m\Big[\alpha_{m-1}^H y_{m-1}(k) \\ &+ K_{m-1}\beta_{m-1}y_{m-2}(k-1)(k) \\ &+ K_{m-1}[\dots[\alpha_1^H y_1(k) \\ &+ K_1 u(k) + K_1\beta_1 u(k-1)]\dots]\Big] \end{aligned} \quad (28)$$

and

$$y_F(k+H) = K_m K_{m-1}\cdots K_1 u(k). \quad (29)$$

The control law for a cascaded internal model can be obtained following the development steps giving an FPC formulation with a parallel internal model, cf. Sections 2.1.1 to 2.1.3, and is derived as

$$\begin{aligned} u(k) &= \frac{(C(k)-y_P(k))(1-\lambda^H)}{K_m K_{m-1}\cdots K_1} \\ &+ \frac{y_m(k)(1-\alpha_m^{H-1})}{K_m K_{m-1}\cdots K_1} \\ &- \frac{(K_m\alpha_{m-1}^H + K_m\beta_m)y_{m-1}(k)}{K_m K_{m-1}\cdots K_1} \\ &- \frac{(K_m K_{m-1}\alpha_{m-2}^H + K_m K_{m-1}\beta_{m-1})y_{m-2}(k)}{K_m K_{m-1}\dots K_1} \\ &- \dots \\ &- \frac{(K_m K_{m-1}\dots K_2\alpha_1^H + K_m K_{m-1}\dots K_2\beta_2)y_1(k)}{K_m K_{m-1}\cdots K_1} \\ &- \beta_1 u(k-1). \end{aligned} \quad (30)$$

For any added measured input disturbances, a procedure similar to that in Section 2.1.4 is followed, though the resultant MV expression is rather more cumbersome and is omitted here for brevity.

2.3. Tuning, constraint handling and time delay compensation in PFC. As the primary goal of this paper is to develop a generic higher order PFC controller based on an ARX/ARMAX representation, tuning techniques, constraint handling and time delay compensation approaches are not dealt with in detail. Indeed, no significant modifications in the matter have been made since the original form (Richalet, 1993; 1998), except for the work by Rossiter and Richalet (2002), investigating unstable systems. The issue of tuning for other predictive control algorithms is not discussed in this paper, as it focuses on PFC only. However, tuning issues in other MPC algorithms, for instance GPC and DMC, may be found in (Clarke *et al.*, 1987; Królikowski and Jerzy, 2001; Kowalczuk and Suchomski, 1999; Garcia and Morshedi, 1987).

A brief idea about how PFC deals with these issues is given in what follows:

Tuning: An exponential reference trajectory is often chosen along with a zeroth order basis function (a step function), cf. Fig. 1. A default choice of $H = 1$ for the co-incidence point is appropriate for first order or well-behaved systems, while a larger value can be chosen for more emphasis on a smooth MV, which is a common requirement in many industrial systems. However, choosing $H = 1$ is unsuitable for non-minimum phase or oscillatory processes as it may lead to instability, and therefore a co-incidence point beyond the inflection points of the transient response should be chosen. These cases will be investigated in Section 3.2. Such choices of tuning parameters (a reference trajectory, a basis function and a co-incidence point) result in particularly straightforward control calculations, which are attractive from an intuitive viewpoint. In PFC, the desired response is normally specified as

$$R_r = \frac{OLRT}{CLRT}, \quad (31)$$

which defines the ratio of the Open Loop Response Time (OLRT, the time to 90% of the final value) to the Closed Loop Response Time (CLRT), T_r, defined in (11). For slow processes, e.g. heat exchange systems, a ratio of 3 is found most suitable (Khadir, 2002). Tuning becomes a one-degree-of-freedom operation, and T_r can be tuned in much the same way as a gain in PID design (Rossiter and Richalet, 2001a).

Constraint handling: PFC uses a simple (but non-optimal) solution to handle constraints. For input constraints, the model is simply given an constrained input value, rather than the manipulated variable calculated by the PFC algorithm (Richalet, 1998). However, for open loop unstable systems with a factor of the form $(s-a)/(s-ra)$, $r > 1$, the original constraint handling scheme may lead to instability. Rossiter and Richalet (2002) proposed a modification to the original approach to ensure stability when controlling such systems, keeping the algorithm simple. Constraints on the Controlled Variable (CV) are handled using a controller override technique, where a separate controller calculates an MV based on a set-point on the actual CV constraints. This MV is used only if the online controller leads the CV beyond the constraints boundaries (Richalet *et al.,* 1978).

Time delay compensation: The delay is still compensated in the same manner as shown in Section 2.1.6.

Example 1. Interleaved system
Consider the third order interleaved system

$$G(s) = \frac{(1+5s)(1+s)}{(1+10s)(1+2s)(1+0.5s)}. \quad (32)$$

A simplified model can be obtained using a balanced realisation transformation followed by order reduction (Moore, 1981). The simplest reduced model can then be obtained in the form of a first order system and is given by

$$G'(s) = \frac{0.99}{(1+8s)}. \quad (33)$$

If the sampling period is chosen to be $T_s = 0.1$, the control results given by PFC controllers, using the full and simplified models as internal prediction models, are given in Fig. 5. The improvement of using a full internal model over a simplified one is clearly noticed, despite the good results obtained with the latter. Tuned to give roughly the same CLRT, the PFC using a full model gives a much faster control response. The first order PFC can only achieve such a speed at the expense of a much more aggressive MV which may violate constraints.

♦

3. Appearance of undesirable controller poles

In the case of a proper system, i.e. where the numerator and the denominator are of the same order, the control law must be modified. A controller pole appears, depending on the values of the process zeros, cf. (23). For simplicity, consider a PFC development for a first order proper system of the form

$$G_M(s) = \frac{K(1+as)}{1+\tau s} = \frac{y_M(s)}{u(s)}. \quad (34)$$

Proceeding as in Sections 2.1.1 to 2.1.3, the following control law can be easily obtained:

$$\begin{aligned} u(k+1) = {} & \frac{\tau(C(k)-y_P(k))(1-\lambda^H)}{Ka} \\ & + \frac{\tau y_M(k)(1-\alpha^H)}{Ka} \\ & - \Big(\frac{\tau}{a}(1-\alpha^H)-1\Big)u(k). \end{aligned} \quad (35)$$

3.1. Stability analysis. Observe that the control law (35) can be represented in the z-domain by

$$u(z) = \frac{N(z)}{z-1+\frac{\tau}{a}(1-\alpha^H)}, \quad (36)$$

where $N(z)$ depends on the particular internal model formulation. It can be seen that the controller contains a pole given by

$$z = 1 - \frac{\tau}{a}(1-\alpha^H). \quad (37)$$

We have the following cases:

1. If $a > \tau(1-\alpha^H)$, then $0 < z < 1$, which gives a stable manipulated variable with no ringing.
2. If $a < \tau(1-\alpha^H)$, then $z < 0$ and $u(k)$ will oscillate with period $2T_s$.
3. If $a < \frac{\tau}{2}(1-\alpha^H)$, including $a < 0$, then $z < -1$, and thus the controller is unstable.

Clearly, an unstable or oscillatory manipulated variable is undesirable and some modification in the PFC algorithm in (35) is required. One possible solution is to decompose the system in (34) as

$$G_M(s) = K_0 + \frac{K_1}{1+\tau s} = \frac{K(1+as)}{1+\tau s}, \quad (38)$$

where

$$K_0 = \frac{Ka}{\tau}, \qquad K_1 = K - \frac{Ka}{\tau}. \quad (39)$$

Since neither individual systems contain a zero, we can employ the control solution for parallel subsystems, as in (16). However, it is found that such a formulation still results in a controller pole, as in (37), appearing *implicitly* in the overall control calculation.

Nevertheless, such an approach can lead to an improvement if a minor adjustment in the process model is allowed. Consider the approximation to the ZOH equivalent of (38) as

$$y_0(k) = K_0 u(k-1), \quad (40)$$

$$y_1(k) = \alpha y_1(k-1) + K_1(1-\alpha)u(k-1), \quad (41)$$

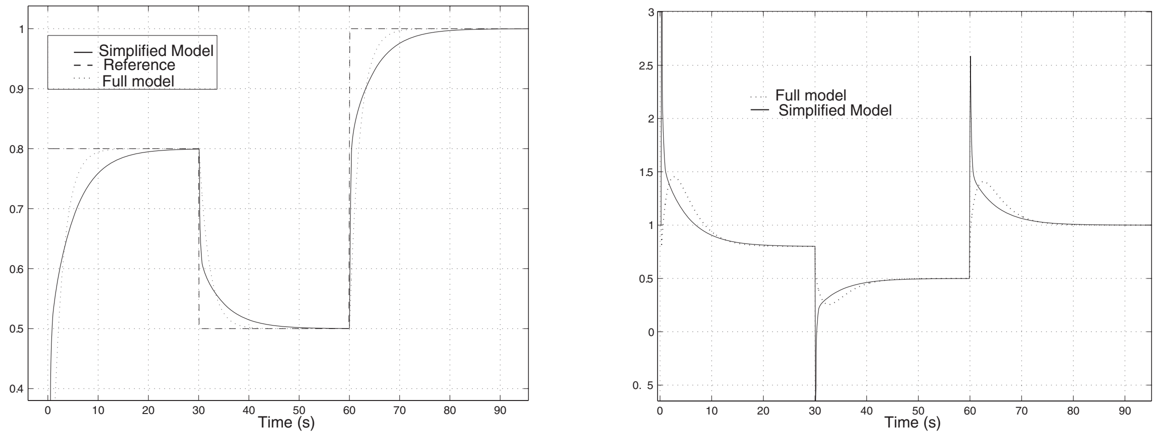
$$y_M(k) = y_0(k) + y_1(k), \quad (42)$$

with an introduction of a one-step (extra) delay into the pure gain term in (40). The control then amounts to steering a sum of two systems:

- a gain/delay system, $y_0(s) = K_0 e^{T_s s} u(s)$, and
- a gain/pole system, $y_1(s) = \frac{K_1}{1+\tau s} u(s)$,

where the composite prediction model autoregressive and forced responses are given respectively by

$$y_A(k+H) = \alpha^H y_1(k) \quad (43)$$



(a) (b)

Fig. 5. PFC response for a third-order interleaved system: (a) controlled variable, (b) manipulated variable.

and

$$y_F(k+H) = (K_0 + K_1(1-\alpha^H))u(k). \tag{44}$$

Proceeding as in Sections 2.1.1 to 2.1.3, the final control law is given by

$$u(k) = \frac{(C(k) - y_P(k))(1-\lambda^H)}{K_1(1-\alpha^H)+K_0} + \frac{y_1(k)(1-\alpha^H)+y_0(k)}{K_1(1-\alpha^H)+K_0}. \tag{45}$$

Equation (45) can be recast to show the controller pole by explicitly writing y_0 and y_1 in terms of $u(k)$ to give

$$u(k) = \frac{(C(k) - y_P(k))(1-\lambda^H)}{K_1(1-\alpha^H)+K_0} + \frac{\alpha y_1(k-1)(1-\alpha^H)}{K_1(1-\alpha^H)+K_0} + \frac{K_1(1-\alpha^H)(1-\alpha)+K_0}{K_1(1-\alpha^H)+K_0}u(k-1), \tag{46}$$

with the controller pole identified as

$$z = \frac{K_1(1-\alpha^H)(1-\alpha)+K_0}{K_1(1-\alpha^H)+K_0}. \tag{47}$$

Using the definitions of α, K_0 and K_1 from (5) and (39), we get the following:

1. If $a > 0$, then $0 < z < 1 \;\forall\; |a|$, which gives a stable manipulated variable with no ringing.

2. If $a < 0$ (a non-minimum phase zero), then $z > 1$, and the controller is unstable. This case is discussed in Section 3.2.

It is clear that such a formulation shifts the controller pole to the positive real axis, which solves the ringing problem. This is illustrated with Example 2.

Note that the stability issue discussed here is limited to this particular case and to this PFC formulation. A more comprehensive and general study of MPC design can be found in (Scattolini *et. al.*, 1999).

Another possible way to eliminate the difficulties caused by the introduction of a controller pole is to perform a factorization of the process zero polynomial, as is commonly done in other control formulations, such as pole placement (Åstrom and Wittenmark, 1997). In this philosophy, the zeros which cause the controller instability or ringing are separated from the plant zero polynomial and can, if desired, be put into the reference model (as is done in (Åstrom and Wittenmark, 1997)). In our case, such zeros are simply discarded (with the preservation of the DC gain). However, although PFC has been shown to be relatively robust to a plant/model mismatch (Khadir, 2002), it was noted in (Khadir and Ringwood, 2003) that this mismatch may become significant as a gets larger, possibly affecting the controller accuracy. Therefore, this model simplification is very much restricted to well behaved processes and will not be investigated further in this paper.

Example 2. Pole/zero system
Consider the system

$$y(s) = \frac{K(1+as)e^{-ds}}{1+\tau s}u(s) + \frac{K_2}{1+\tau s}v(s), \tag{48}$$

with the parameter values given in Table 1.

An exact, but delay free, internal model is given by

$$y_{M1}(s) = \frac{K(1+as)}{1+\tau s}u(s) + \frac{K_2}{1+\tau s}v(s). \tag{49}$$

Table 1. Parameter values of Example 2 .

Parameter	K	a	K_2	τ	d
	1	0.5	2	30	10

Further decomposing the pole/zero system, the internal model can be approximated as

$$y_{M2}(s) = \left(\frac{Ka}{\tau}e^{-T_s s} + \frac{K - \frac{Ka}{\tau}}{1+\tau s}\right)u(s) + \frac{K_2}{1+\tau s}v(s). \tag{50}$$

The control performance of PFC controllers, based on two different internal models (M1 and M2, as given by (49) and (50), respectively) and including time delay compensation as per (24), are given in Fig. 6 for $a = 0.5$.

From Fig. 6 it can be seen that although both controllers produce satisfactory control, the manipulated variable provided by the controller based on M1 sustains heavy ringing. This is caused by the presence of a controller pole between 0 and -1 (see Section 3.1).

3.2. Extension to the higher order case. A higher order proper system in a parallel form can be given by

$$G_M(s) = K_0 + \sum_{i=1}^{m} \frac{K_i}{1+\tau_i s}. \tag{51}$$

If desired, the internal model delay modification, as in (40), can be made to (20) to avoid potential ringing on the MV, with the modified controller calculation of

$$u(k) = \frac{(C(k) - y_P(k))(1-\lambda^H)}{\sum_{i=1}^{m} K_i(1-\alpha_i) + K_0} + \frac{\sum_{i=1}^{m} y_i(k)(1-\alpha_i) + y_0(k)}{\sum_{i=1}^{m} K_i(1-\alpha_i) + K_0}. \tag{52}$$

Note that any input disturbances can always be handled as in (18).

4. Non-minimum phase and complex pole systems

The generic control formulations, derived in (16), (18) and (23) for the higher order case, cover a relatively wide range of process behavior. They can, however, be less effective for unstable processes, processes with complex poles and non-minimum phase dynamics. This point introduces instability into the control formulation (as shown in Section 3.1), and the complex case requires some modifications to avoid using complex numbers to calculate a "real" MV.

Although PFC can deal successfully with unstable systems, this is not the main focus of the paper. The original approach, when dealing with such systems, was to decompose the initial unstable system into two stable systems (Richalet *et al.*, 1978). However, such an approach was found to be limited to systems with factors of the form $(s-a)/(s-ra), r > 1$, when the presence of constraints (Rossiter and Richalet, 2001a) is considered. This problem was addressed by the work (Rossiter and Richalet, 2002) using stabilised prediction (Rossiter, 2001).

A popular solution, implemented on many industrial processes, is to leave the process model in its original unfactored ARMAX form as

$$y_M(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_m z^{-m}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n}} u(z) + \frac{c_0 + c_1 z^{-1} + c_2 z^{-2} + \dots + c_q z^{-m}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n}} v(z) \tag{53}$$

leading to the following difference equation:

$$y_M(k) = -\sum_{i=1}^{n} a_i y_M(k-i) + \sum_{i=0}^{m} b_i u(k-i) + \sum_{i=0}^{q} c_i v(k-i), \tag{54}$$

where u is the MV and v is the input disturbance. Any complex poles will be induced in the values of the denominator a_n.

The free (autoregressive) response of the system is given by y_A, when $u(k) = v(k) = 0$, as

$$y_A(k) = -\sum_{i=1}^{n} a_i y_A(k-i). \tag{55}$$

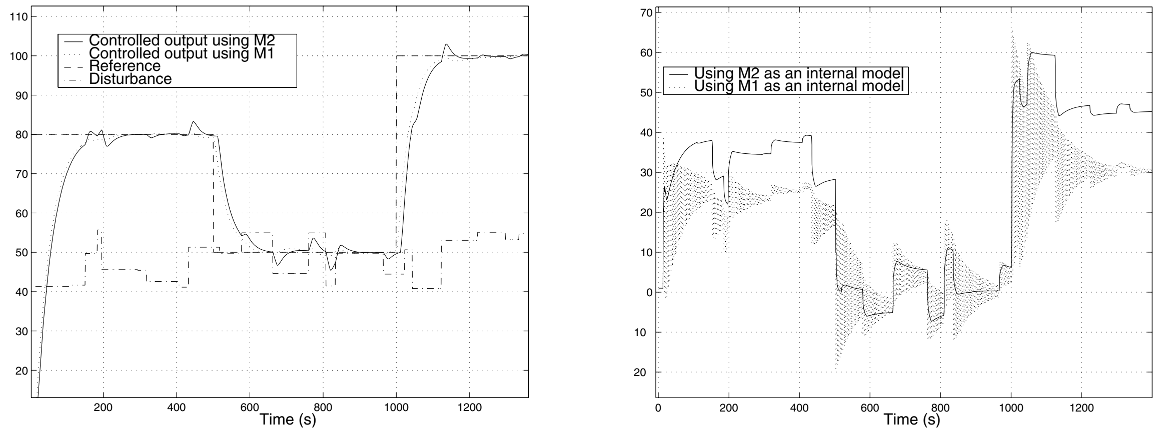
The future free response of a linear system, at $k+1$, can also be given using the future N elementary outputs $y_1(k+1)$, $y_2(k+1)$ to $y_N(k+1)$ as

$$y_A(k+1) = y_1(k+1)y_M(k) + y_2(k+1)y_M(k-1) + \dots + y_N(k+1)y_M(k-N+1), \tag{56}$$

where $y_i(k+1)$ are obtained from (54) setting all initial conditions to zero except the one set to unity successively as

$$\begin{aligned} y_1(k+1) &= -a_1 y_1(k), & y_1(k) &= 1,\\ y_2(k+1) &= -a_1 0 - a_2 y_2(k-1), & y_2(k-1) &= 1,\\ y_N(k+1) &= -a_1 0 - a_2 0 - \cdots \\ &\quad - a_n y_N(k-n+1), & y_N(k-n+1) &= 1. \end{aligned}$$

In practice, we require to iterate the elementary outputs $y_i(k+H)$ N times, starting at time $k=1$. This is



(a) (b)

Fig. 6. PFC performance using y_{M1} and y_{M2} as internal models (with a=0.5): (a) controlled variable, (b) manipulated variable.

done only once at the start of the real time calculations of the MV, with all initial conditions being zero except one until reaching the coincidence point H, thus obtaining a set of values $y_i(k+h)$. These values are stored in computer memory and used to obtain the free response (57) at the coincidence point h as

$$\begin{aligned} y_A(k+H) &= y_1(k+H)y_M(k)+y_2(k+H)y_M(k-1) \\ &\quad +\dots+y_N(k+H)y(k-N+1). \end{aligned} \tag{57}$$

Note that the formulation of the free response, as in (56), is only possible when using independent models as all future and past values of $y(k)$ are known, which is a unique attribute of the independent model formulation (Rossiter, 2001).

The forced response is calculated going from zero initial conditions and applying a step input. The final control law, obtained following the steps taken in Sections 2.1.1 to 2.1.3, is then given by Eqn. (58), where $p_1 = y_1(k+H), p_2 = y_2(k+H), \dots, p_N = y_N(k+H)$, and K is the overall gain of the system.

For the ARX case, the control law is identical to (58) setting $c_i = 0$. Some guidance points in the determination of the co-incidence horizon are as follows:

Complex: Choose H to be, at least, one period of the open-loop complex response, and

Non-mimimum phase: Longer than the inverse 'dip' in the step response.

Example 3. Third order non-minimum phase with a pure time delay and complex poles
Consider the non-minimum phase third order system with complex poles and a pure time delay:

$$G_M(z) = \frac{(b_0 + b_1 s + b_2 s^2)e^{-20s}}{1 + a_1 s + a_2 s^2 + a_3 s^3}, \tag{59}$$

with the model parameters given in Table 2. The system has complex poles at$-0.033 \pm 0.451j$ and a zero at -2.139, which introduces non-minimum phase behavior.

Table 2. Parameter values for Example 3.

	Index i		
Parameter	1	2	3
a_i	-2.016	-1.160	0
b_i	10.327	8.151	48.899
Delay	20		

T_r was chosen to be 100 s, and the coincidence point is $H = 22$, i.e. beyond the dip. Applying the control algorithm in (58), we obtained the results of Fig. 7. To show the power of the simple PFC algorithm, the free response of the system is also shown in Fig. 7. It can be seen that the control is effective and overcomes the oscillatory, non-minimum phase dynamics and the pure time delay of the system.

5. Conclusion

This paper has developed higher order solutions to SISO processes, based on ARX/ARMAX input/output process descriptions, retaining the intuitive appeal of such PFC formulations. Combining the decomposition techniques of Section 2 with the further developments in Sections 3 and 4, most SISO industrial processes can be handled

$$u(k) = \frac{(C(k) - y_P(k))(1 - \lambda^H) + y_M(k)\sum_{i=1}^{N} p_i y_M(k-i)}{K\left(1 - \sum_{i=1}^{N} p_i\right)} - \frac{\sum_{i=0}^{q} c_i v(k-i)}{K\left(1 - \sum_{i=1}^{N} p_i\right)} \tag{58}$$

with one, or at most two parameters, to tune, i.e. T_r and H. The choice of H is process and MV/CV trade-off dependent, but it still results in a choice of a single parameter. This makes PFC easier to tune than PID, qualifying it as an ideal candidate for industrial use where good dynamic performance and intuitive appeal are paramount. It is clear that a more complex MPC formulation, such as GPC, especially with the solution of the Diophantine equation, will produce similar results as the PFC formulations presented in this paper. However, the presented PFC formulation is much simpler, inducing a straightforward formulation better suited for practitioners.

Though there is a direct equivalence between parallel and cascade decompositions, the higher order PFC using an internal model formulation based on a parallel internal model form is found to be less computationally expensive, and more elegant, than that using a cascaded form, and thus the authors strongly recommend using a parallel decomposition of the internal model.

However, such a compact form is not possible with oscillatory or non-minimum phase systems, since the control parameters have to be calculated from the system's free response utilising a form of the unfactored process difference equation.

Though extra computational expense is incurred in higher order controllers, this is not problematic due to cheap computational power, and the performance advantage is demonstrated clearly in our illustrative examples. Most importantly, the computation is minimised by using an input/output formulation (since matrix computations, often containing zero elements, are avoided), and the simplicity and intuitive appeal are maximised.

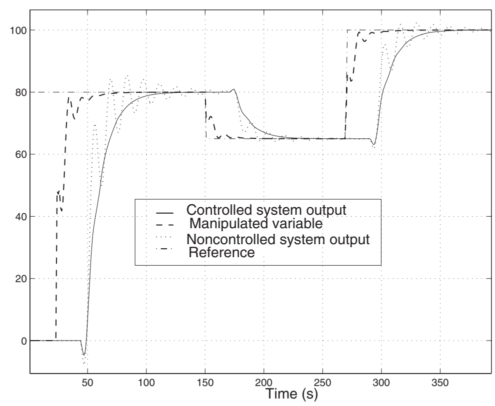


Fig. 7. PFC performance for a third order non-minimum phase complex system with a pure time delay.

References

Åstrom K.J. and Wittenmark B. (1997). *Computer-Controlled Systems: Theory and Design, 3rd Ed.*, Prentice-Hall, Englewood Cliffs, NJ.

Clarke D.W., Mohtadi C. and Tuffs P.S. (1987). Generalised predictive control – Part I. The basic algorithm, *Automatica* **23**(2): 137–148.

Cutler C.R., and Ramaker P.S. (1980). Dynamic matrix control — A computer algorithm, *Proceedings of the Joint Automatic Control Conference*, San Francisco, CA, USA, paper No. WP5-B.

Houpis C.H and Lamont G.B. (1992). *Digital Control Systems*, McGraw-Hill, New York.

Khadir M.T. (2002). *Modelling and predictive control of a milk pasteurisation plant*, Ph.D. thesis, Department of Electronic Engineering, NUI, Maynooth, Ireland.

Khadir M.T. and Ringwood J. (2003). Higher order predictive functional control, *Internal report EE/JVR/01, Department of Electronic Engineering, NUI Maynooth*.

Królikowski A. and Jerzy D. (2001). Self tuning generalised predictive control with input constraints, *International Journal of Applied Mathematics and Computer Science* **11**(2): 459-479.

Kowalczuk Z. and Suchomski P. (1999). Analytical design of stable continuous time generalised predictive control, *International Journal of Applied Mathematics and Computer Science* **9**(1): 53–100.

Maciejowski J.M. (2002). *Predictive Control with Constraints*, Prentice Hall, London.

Moore B.C. (1981). Principal component analysis in linear systems: Controlability, observability, and model reduction, *IEEE Transactions on Automatic Control* **26**(1): 17–32.

Morari M. and Lee J.H. (2000). Model predictive control: Past, present and future, *Computers and Chemical Engineering* **23**(4): 667–682.

Richalet J., Raul A., Testud J.L., and Papo J. (1976). Algorithmic control of industrial processes, *Proceedings of the 4-th IFAC Symposium on Identification and System Parameters Estimation, Tbilissi, URSS*, pp. 1119–1167.

Richalet J., Raul A., Testud J.L. and Papon J. (1978). Model predictive heuristic control: Application to industrial processes, *Automatica* **14**(5): 413–428.

Richalet J. (1993). *Pratique de la commande predictive*, Hermès, Paris.

Richalet, J. (1998). La commande predictive, *Techniques de l'Ingenieur Traite Mesure et Control,* R7 423, pp. 1–17.

Rossiter J.A. and Richalet J. (2001a). Predictive functional control of unstable processes, *Internal report 807, Department of Automatic Control and Systems Engineering, Sheffield University, UK*.

Rossiter J.A. and Richalet J. (2001b). Realigned models for prediction in MPC: A good thing or not? *Proceedings of Advanced Process Control 6,* New York, UK, pp. 63-70.

Rossiter J.A. (2001). Stable predictive for unstable independent models, *Internal report 812, Department of Automatic Control and Systems Engineering, Sheffield University, UK*.

Rossiter, J.A., and Richalet, J. (2002). Handling constraints with predictive functional control of unstable processes, *Proceedings of the American Control Conference*, Anchorage, Alaska, AK, pp. 4746–4751.

Scattolini R., De Nicolao G. and Magni L. (1999). Some issues in the design of predictive control, *International Journal of Applied Mathematics and Computer Science* **9**(1): pp. 9-24.

Skrjank I. and Matko D. (2000). Predictive functional control based on fuzzy model for heat-exchanger pilot plant, *IEEE Transactions on Fuzzy Systems* **8**(6): 705–712.

Tatjewski P. (2007). *Advanced Control of Industrial Processes: Structures and Algorithms*, Springer, London.

Qin S.J. and Badgwell T.A. (2003). A survey of industrial model predictive control technology, *Control Engineering Practice* **11**(7): 733–767.

Received: 8 August 2007
Revised: 2 December 2007