Monograph:

Rossiter, J.A. and Richalet, J. (2001) Predictive Functional Control of Unstable Processes. Research Report. ACSE Research Report 811 . Department of Automatic Control and Systems Engineering

Predictive functional control of unstable processes

by J A Rossiter and J Richalet

RESEARCH REPORT NO. 811

October 2001

PREDICTIVE FUNCTIONAL CONTROL OF UNSTABLE PROCESSES

J.A. Rossiter *

* *Dept. of Automatic Control & Systems Engineering, Sheffield University, Mappin Street, Sheffield, S1 3JD, email:* `J.A.Rossiter@sheffield.ac.uk`, *Tel. 44 114 2225685*

J. Richalet **

** *ADERSA, 10 Rue de la Croix Martre, 91873 PALAISEAU CEDEX, France, email:* `jacques.richalet@adersa.com`

Abstract: Application of predictive functional control (PFC) has been very succesful on many processes, however application to unstable open-loop processes has met more varied success rates. Certain processes in particular with factors of the type $(s-a)/(s-ra)$, $r > 1$ have proved very difficult to stabilise. This paper illustrates how the prestabilisation approach to prediction can be used to overcome this bottleneck and with the added bonus of retaining the intuitive tuning parameters that make PFC popular.

Keywords: Predictive functional control, stability

1. INTRODUCTION

Predictive functional control has evolved from IDCOM (Richalet *et al.*, 1978) and is a simple variant of predictive control. This simplicity is its very appeal and has allowed successful implementations in many processes where the full MPC (Model predictive control) products such as DMC (Cutler *et al.*, 1980) are too expensive and/or too complicated. Another feature of PFC is that its tuning parameters are intuitive for practising engineers and hence it is easier to understand and own than say GPC (Clarke *et al.*, 1987).

Of course the weakness of simplified control algorithms is that they are not so powerful and cannot be applied as widely. PFC as it stands does not readily extend to multivariable systems (a major selling point of MPC), however it does handle constraints which is another important feature of MPC (Garcia *et al.*, 1989). In this paper we wish to concentrate on an issue which is well studied in the literature for MPC (e.g. (Kouvaritakis *et al.*, 1992; Rawlings *et al.*, 1993; Rossiter *et al.*, 1996; Scokaert *et al.*, 1998)) but not at all for PFC. That is when does the algorithm give stabilising control ?

Although some apriori stability results are possible in PFC (as yet unpublished), the general practice in industry is not to worry about apriori results (Kouvaritakis *et al.*, 1996) and instead make use of aposteriori results. That is, during offline design and tuning, because the control law for the unconstrained case is a fixed linear feedback, it is easy to compute the implied closed-loop poles and hence chose the tuning parameters that give the most desirable closed-loop behaviour. As this process is offline and via computer simulation it is relatively cheap. It should also be noted, that for MPC at least, notional guidelines of a large prediction horizon and a small, but not too small, control horizon will nearly always give nominal stability. For PFC similar guidelines exist, for instance using a large coincidence horizon in conjunction with a stable process will give closed-loop behaviour similar to the open-loop dynamics.

However, all this changes when one encounters unstable open-loop processes. Apart from numerical conditionning issues (e.g. (Rossiter *et al.*, 1998)) normal guidelines do not always give stabilising control and this is particularly a difficultly with PFC. Hence, the aim of this paper is to develop a means whereby PFC can be given a high apriori expectation of closed-loop stability, when the current algorithm almost always fails. Such a result opens up the potential for application to many more processes. The methodology adiopted here is based on the observation in (Rossiter *et al.*, 1998) that using unstable predictions as a basis for a predictive control law design is unwise. Moroever the part of the prediction that is ignored, that is the part beyond the output horizon, is the most troublesome part because it is divergent. Hence one can not make recursive feasibility claims (Kouvaritakis *et al.*, 1996) and instability could easily be caused solely by the presence of constraints. In summary, there is a need to ensure that the prediction class is stable, even for unstable open-loop plant. In fact dual-mode strategies and recent work making use of invariant sets fulfil this need explicitly by constraining the predictions to that implied by a fixed closed-loop beyond a certain horizon.

Most articles (with the exception of (Garcia *et al.*, 1982)) on MPC make use of state realignment in prediction (Rossiter *et al.*, 2001). That is one assumes that one can initialise a model with the process measurements and then use this in prediction. However state realignment (Rossiter *et al.*, 2001) does not always give good predictions (too sensitive to measurement noise) and so many many industrial packages use FIR (Finite impulse response) or Independent Model (IM) representations (Garcia *et al.*, 1982). An IM is equivalent to a FIR but has the advantage of requiring less parameters and removing truncation errors. As a consequence, this article will adopt an IM structure rather than state realignment, although the results can be reworked easily enough. Hence, as argued above, there is a need to find a reparameterisation of the degrees of freedom to give stable predictions for an unstable process represented by an independent model. This parameterisation can then be used to develop an appropriate PFC algorithm.

The paper will be set out as follows. Section 2 will give some background on PFC, section 3 develops prediction equations and shows how prestabilisation can be achieved and then applies the prestabilised equations to PFC to give the 'new' algorithm. The paper is completed with some simulation examples and conclusions.

2. BACKGROUND ON PFC

In this paper we will adopt the notation of y, u for process outputs and inputs respectively. z^{-1} is the unit delay operator such that $z^{-1}y_k = y_{k-1}$ where y_k denotes the value of y at the kth sampling instant and $y_{k+i|k}$ denotes the predicted value of y_{k+i} computed with information available at sampling instant k.

The most common features of predictive control, shared by PFC are

(1) A system model, which is used to generate system predictions
(2) A performance specification
(3) Online selection of the controls from substituting predictions into the performance specification.

The modelling and performance specification will be discussed next and prediction left to section 3 as this is a significant part of the paper.

2.1 *Independent models of unstable processes*

In PFC it is usual to use an IM for prediction. An independent model is intended to represent the process as closely as possible so that it has matching inputs and outputs. Let y_m be the output of the independent models (IM). The norm is to simulate the IM in parallel with the process, using the same inputs u and hence in general, due to uncertainty, $y \neq y_m$.

The difficulty with an unstable process is that such a parallel simulation cannot work because the same input would not stabilise the IM and an uncertain plant. One solution (e.g. (Richalet,)) is to decompose the model into two parts as in Figure 1. Hence if the process is modelled by G, then

$$G = (I + M_2)^{-1} M_1 = \frac{n}{d} \qquad (1)$$

where both M_1 and M_2 are stable. Next note that, in the nominal case of $y = y_m$, the output of figure 2 is the same as that of figure 1, so equivalent to using the structure of Figure 1, one could use the structure of figure 2. In this case if u is stable, so is w and if y is stable so is z. Hence when the process is stabilised, so is the output of the independent model.

A convenient decomposition in the SISO case is as follows:

$$G = \frac{n_+ n_-}{d_+ d_-}; \; M_1 = \frac{n_+}{d_-}; \; M_2 = \frac{b_2}{n_-}; \; b_2 = n_- - d_+ \qquad (2)$$

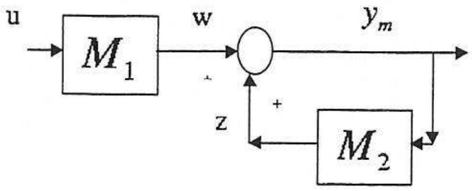

Fig. 1. Independent model used for prediction

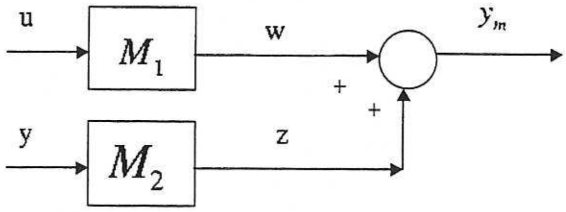

Fig. 2. Implementation of independent model for simulation

where n_+, d_+ are the numerator/denominator factors respectively containing roots outside the unit circle (unstable). It is clear that both M_1 and M_2 have stable poles.

2.2 *Predictive functional control online objective*

For this paper we focus on a PFC variant with just one degree of freedom. The aim here is to specify offline the desired closed-loop performance in terms of the response of a first order lag and hence the specification is defined mainly by a time constant. In PFC one chooses: (i) the lag (that is the time constant, say T_{PFC} or equivalently the target pole $e^{-\frac{T}{T_{PFC}}}$ where T is the sampling period.) and (ii) the prediction horizon say T_h. The prediction horizon is denoted the coincidence horizon. Compute the response of a target 1st order lag T_h seconds ahead. The control move is then selected as the control which will cause the predicted plant output to coincide with the desired lag T_h seconds ahead; hence the terminology *coincidence point*. In such a way one hopes that in the closed-loop the process output will behave in a similar fashion to the selected lag.

To place this algorithm in a more mathematical context (for the delay free case), let T_h seconds correspond to n_y samples (i.e $n_yT = T_h$), then the online computation reduces to solving a simple set of equalities:

$$\begin{aligned} \text{target} &= y_k + (r_{k+n_y} - y_k)(1 - e^{-\frac{T_h}{T_{PFC}}}) \\ y_{k+n_y|k} &= \text{target} \end{aligned} \qquad (3)$$

The 'target' is the *coincidence point* where the prediction is made to coincide with the target trajectory. If n_y (or equivalently T_h) is too large, then this strategy reduces to choosing the control increment which will drive the predicted output to the desired set-point in steady-state and hence will give only open-loop characteristics. So in order for the choice of lag to be effective, the value T_h must be appropriate to the desired lag time constant position T_{PFC}, for instance $e^{-\frac{T_h}{T_{PFC}}} > 0.1$.

Remark 2.1. Where the system has complex open-loop dynamics, it is often necessary to use more than one coincidence point. These processes and the solutions are not discussed here, especially because the design immediately becomes more complicated and less intuitive. Instead the aim of this paper is to see how far one can go with just one coincidence point.

2.3 *Parameterisation of the freedom in the predictions*

It is typical in MPC to use changes in control as the degrees of freedom in the predictions. In particular the use of changes (rather than absolute) allows a neat means of ensuring offset free control. In PFC a similar practice has been followed where there is one coincidence horizon, that is to use the change in control at the current time as the degree of freedom. Where more that one coincidence horizon is deployed, the 2nd degree of freedom is often a ramp rate for the input but we shall not consider those variants here. What is more important is to note that other parameterisations of the degrees of freedom could be deployed e.g. (Kouvaritakis *et al.*, 1998; Wang, 2000), perhaps to much advantage, but this has not yet been investigated. In this paper we consider one such reparameterisation.

There is a need to parameterise the degrees of freedom appropriately for the process to be controlled. For stable plant, changes in control are intuitive and simple to use and hence are appropriate, however for unstable plant this is not the case. In the first instance one must allow several d.o.f. in order to bring the unstable dynamics under control within the output horizon e.g.(Rawlings *et al.*, 1993; Rossiter *et al.*, 1996). To be more sure, it is better to place structure into the future control trajectory in such a way that the output prediction is known to be stable e.g.(Kouvaritakis *et al.*, 1992; Rawlings *et al.*, 1993; Rossiter *et al.*, 1996; Scokaert *et al.*, 1998; Kouvaritakis *et al.*, 1998; Rossiter *et al.*, 1998). The question to be answered is what structure to deploy and how to express the d.o.f. within that ? Here one solution closely related to (Rawlings *et al.*, 1993; Rossiter *et al.*, 1996) is given. Investigation of other possible solutions forms future work.

3. PREDICTION WITH AN INDEPENDENT MODEL

This section summarises the work of (Rossiter, 2001). In order to implement the PFC algorithm of eqn.(3) it is clear that a prediction is needed and within that a clear separation of the degrees of freedom to ease solution of (3). When using an independent model for prediction, future y are unknown. Hence, we use partial state realignment, that is return to figure 1 and realign the loop variable y_m on the process output measurement, then use this as a basis for prediction, where now the only unknown is the future input variable u; these values of course are the usual degrees of freedom.

3.1 *Notation*

Define vectors of future (arrow pointing right) and past values (arrow pointing left)

$$\Delta \underset{\rightarrow}{u} = \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+n_u-1} \end{bmatrix}; \quad \underset{\rightarrow}{y} = \begin{bmatrix} y_{k+1} \\ y_{k+2} \\ \vdots \\ y_{k+n_y} \end{bmatrix}$$

$$\Delta \underset{\leftarrow}{u} = \begin{bmatrix} \Delta u_{k-1} \\ \Delta u_{k-2} \\ \vdots \end{bmatrix}; \quad \underset{\leftarrow}{y} = \begin{bmatrix} y_k \\ y_{k-1} \\ \vdots \end{bmatrix}$$

In general the vector of future values can be any length, but the lengths n_y will correspond to the coincidence horizon and n_u the input horizon (taken to be one here). For computing predictions, we will use the Toeplitz/Hankel notation for a given polynomial $n(z) = n_0 + n_1 z^{-1} + \ldots$, then

$$C_n = \begin{bmatrix} n_0 & 0 & 0 & \ldots \\ n_1 & n_0 & 0 & \ldots \\ n_2 & n_1 & n_0 & \ldots \\ \vdots & \vdots & \vdots & \vdots \\ n_m & n_{m-1} & n_{m-2} & \vdots \end{bmatrix} \tag{4}$$

$$H_n = \begin{bmatrix} n_1 & \ldots & n_{m-1} & n_m \\ n_2 & \ldots & n_m & 0 \\ \vdots & \vdots & \vdots & \vdots \\ n_m & 0 & \vdots & 0 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

Also note: (i) the definition that Γ_n is a tall and thin variant on the square matrix C_n so that $[1, z^{-1}, z^{-2} \ldots]\Gamma_n b = n(z)[1, z^{-1}, \ldots]b$; (ii) the commutative and inverse properties $C_a C_b = C_b C_a$; $C_a^{-1} = C_{1/a}$ and (iii) throughout this paper dimensions are flexible to fit the context.

3.2 *Prediction without prestabilisation*

Prediction reduces to the solution of some simultaneous equations. These come from setting up consistency conditions around M_1 and M_2 (from figure 1 but with initial conditions taken from figure 2) at each future time instance, e.g.

$$\begin{aligned} C_{d_-} \underset{\rightarrow}{w} &= C_{n_+} \underset{\rightarrow}{u} + H_{n_+} \underset{\leftarrow}{u} - H_{d_-} \underset{\leftarrow}{w} \\ C_{n_-} \underset{\rightarrow}{z} &= C_{b_2}[\underset{\rightarrow}{y}_m + \hat{d}] + H_{b_2} \underset{\leftarrow}{y} - H_{n_-} \underset{\leftarrow}{z} \\ \underset{\rightarrow}{y}_m &= \underset{\rightarrow}{z} + \underset{\rightarrow}{w} \\ \underset{\rightarrow}{y} &= \underset{\rightarrow}{y}_m + \hat{d} \\ \hat{d} &= L(\underset{\leftarrow}{y} - \underset{\leftarrow}{z} - \underset{\leftarrow}{w}) \\ \underset{\rightarrow}{u} &= E\Delta \underset{\rightarrow}{u} + L \underset{\leftarrow}{u} \end{aligned} \tag{5}$$

where $\hat{d}$ represents a correction for offset (common in DMC), L is a vector of ones and E is a lower triangular matrix of ones. The solution of (5) takes the form (details are provided in (Rossiter, 2001)):

$$\begin{aligned} \underset{\rightarrow}{y} &= \underset{\rightarrow}{y}_m + L(\underset{\leftarrow}{y} - \underset{\leftarrow}{z} - \underset{\leftarrow}{w}) \\ \underset{\rightarrow}{y}_m &= C_{d_-}^{-1} C_{\Delta}^{-1} C_{d_+}^{-1} \{C_n \Delta \underset{\rightarrow}{u} + M_s v\} \end{aligned} \tag{6}$$

where

$$\begin{aligned} M_s &= [K_u, K_w, K_y, K_z]; \\ K_u &= C_{\Delta}(C_n L + C_{n_-} H_{n_+}) \\ K_w &= -C_{\Delta}(C_{n_-} H_{d_-} + C_{d_-} C_{b_2} L); \quad v = \begin{bmatrix} \underset{\leftarrow}{u} \\ \underset{\leftarrow}{w} \\ \underset{\leftarrow}{y} \\ \underset{\leftarrow}{z} \end{bmatrix} \\ K_y &= C_{\Delta} C_{d_-}(C_{b_2} L + H_{b_2}) \\ K_z &= -C_{\Delta} C_{d_-}(C_{b_2} L + H_{n_-}) \end{aligned}$$

It is clear that one can easily decouple the degrees of freedom ($\Delta \underset{\rightarrow}{u}$) and the notional free response (depending on measured data), but this is omitted to save space as it will not be used here. Moreover it is evident that these predictions diverge due to the term $C_{d_+}^{-1}$.

Remark 3.1. As noted in (Rossiter *et al.*, 1998; Rossiter, 2001), and any simple examples, control laws based on predictions (6) can easily fail due to:

(1) Numerical ill-conditionning (divergent predictions cause numerical difficulties in computing $\underset{\rightarrow}{y}_m$ and hence the control law accurately)
(2) With a small number of degrees of freedom it is difficult to make the output predictions near stable. Hence predicted errors beyond this horizon rapidly diverge.
(3) As one increases n_u to overcome this, one must increase n_y (guidelines given $n_y \gg n_u$) and then numerical ill-conditionning may occur.

3.3 *Stabilising the predictions from a realigned model*

Given the analytic nature of the predictions in (6), it is straightforward to form a parameterisation of future inputs that stabilises the predictions. Rewrite eqn.(6) in terms of z-transforms

$$\underset{\rightarrow m}{y}(z) = \frac{n(z)\Delta \underset{\rightarrow}{u}(z) + x(z)}{d_-(z)d_+(z)\Delta(z)}; \; x(z) = [1, z^{-1}, ...]M_s v \quad (7)$$

The output predictions $\underset{\rightarrow m}{y}$ are stable iff the numerator term contains a factor $d_+(z)$ which implies the following constraint on $\Delta \underset{\rightarrow}{u}(z)$:

$$n(z)\Delta \underset{\rightarrow}{u}(z) + x(z) = d_+(z)\gamma(z) \quad (8)$$

where $\gamma(z)$ is a degree of freedom (assumed here to be a polynomial). Constraint (8) can be written as a matrix equation

$$[\Gamma_n, -\Gamma_{d_+}] \begin{bmatrix} \Delta \underset{\rightarrow}{u} \\ \gamma \end{bmatrix} = M_s v \quad (9)$$

which has a solution of the following form:

$$\begin{aligned} \Delta \underset{\rightarrow}{u} &= K_1 v + \Gamma_{d_+} \underset{\rightarrow}{c} \\ \gamma &= K_2 v \end{aligned} \quad (10)$$

where matrices K_1, K_2 are suitable block matrices from $[\Gamma_n, -\Gamma_{d_+}]^{-1} M_s$. The corresponding output prediction is

$$\underset{\rightarrow m}{y} = [C_\Delta C_{d_-}]^{-1}[K_2 v + \Gamma_n \underset{\rightarrow}{c}] \quad (11)$$

Hence, the process output predictions ($\underset{\rightarrow}{y} = \underset{\rightarrow m}{y} + \hat{d}$) are given as

$$\begin{aligned} &\underset{\rightarrow}{y} = H_1 \underset{\rightarrow}{c} + M_y v \\ &H_1 = [C_\Delta C_{d_-}]^{-1}\Gamma_n \\ &M_y \equiv [C_\Delta C_{d_-}]^{-1} K_2 + [0, -L, L, -L] \end{aligned} \quad (12)$$

3.4 *The PFC algorithm with prestabilised predictions*

Using the algorithm of (3) as a basis, the description of a PFC algorithm based on predictions (12) is elementary. Choose $\underset{\rightarrow}{c}$ to have just one variable c_k and define $\mathbf{e}_{n_y}^T$ to be the n_y^{th} standard basis vector. Then

$$y_{k+n_y|k} = \mathbf{e}_{n_y}^T [H_1 c_k + M_y v] \quad (13)$$

Solving for coincidence (3) gives

$$c_k = \frac{y_k + (r_{k+n_y} - y_k)(1 - e^{-\frac{T_h}{T_{PFC}}}) - \mathbf{e}_{n_y}^T M_y v}{\mathbf{e}_{n_y}^T H_1} \quad (14)$$

One can then substitute (14) into (10) to find the new control increment.

$$\Delta u_k = \mathbf{e}_1^T [K_1 v + \Gamma_{d_+} c_k] \quad (15)$$

Remark 3.2. The use of the IM (Figure 2) and control law (15) can be represented as a fixed term control law if desired for closed-loop analysis. The implementation would be via (14,15).

4. EXAMPLES

In this section the efficacy of the PFC algorithm based on prestabilised equations is demonstrated. The following unstable process was used

$$G = \frac{s-1}{s^2 - 1.5s - 1} \quad (16)$$

This was sampled at a rate of 0.2 sec giving discrete poles at approx 1.5, 0.9.

Remark 4.1. PFC based on the (unstable) prediction equations of (6) with Δu_k as the degree of freedom, did not give a stable closed-loop. However, it is obvious that ensuring closed-loop stability is straightforward with the prestabilised predictions, even with just a single coincidence horizon.

4.1 *Simulation 1*

First we illustrate, apart from stability which is apparent, that the PFC tuning parameters can still be used in the usual intuitive way. The following (see table 1) pairs of coincidence horizons and discrete 'target' poles are selected. In terms of time constants recall that target pole p is given from

$$p = e^{-\frac{T}{T_{PFC}}} \quad (17)$$

To simplify comparison, the coincidence horizon is taken as a default value whereby $p_y^n \approx 0.1$ where n_y is the coincidence horizon. Clearly other choices are possible. Moreover it is noted that $n_y \geq 6$ or an unstable closed-loop results; this is to be expected as lower bounds on the prediction horizon are also required when MPC is applied to unstable processes. The corresponding simulations are displayed in figure 3 where the x-axis is in sampling instants.

	Pair 1	Pair 2	Pair 3	Pair 4
n_y	6	8	10	20
Target pole	0.5	0.6	0.8	0.9
Line	Solid	Dashed	Dotted	Dash-dot

Table 1. Tuning parameters and legend for fig. 3

It is clear that in accordance with expectations, the closed-loop behaviour has a strong correlation with the selected target pole. That is as the target pole is speeded up so does the closed-loop response with a corresponding increase in input activity.

4.2 *Simulation 2*

The second simulation is to illustrate that the controller is robust. Hence for the choice of pair 3 (from table 1) a further closed-loop simulation is performed as seen in figure 4. This time a step

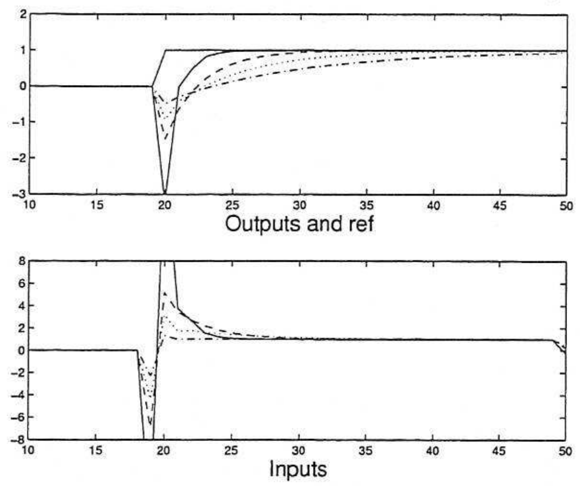

Fig. 3. Simulations for tuning pairs in table 1

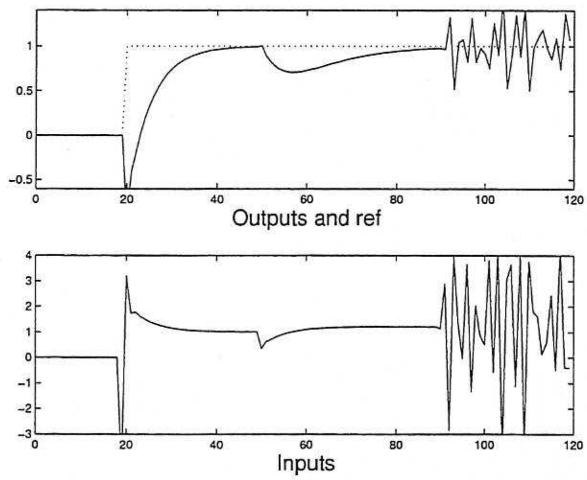

Fig. 4. Simulations with disturbances and noise

output disturbance is introduced at the 50th sampling instant and measurement noise is introduced at the 90th sampling instant. It is clear that the the disturbance is rejected effectively with zero offset and so is the noise [1].

5. CONCLUSION

This paper has shown how the method of prestabilisation can be developed for IM models and hence used to extend the applicability of PFC to open-loop unstable processes. The efficacy of the proposed algorithm was illustrated by an example. Future work will consider extensions to more than one coincidence point, constraint handling and other parameterisations of stabilising predictions.

[1] From the work of (Rossiter, 2001) on stable models, it is expected that an IM will give better noise rejection than a realigned model. A comparison for unstable models constitutes future work

6. REFERENCES

Clarke, D.W., C. Mohtadi and P.S. Tuffs (1987). Generalised predictive control, Parts 1 and 2, *Automatica*, **23**, pp. 137-160

Cutler, C.R. and B.L. Ramaker (1980), Dynamic matrix control - a computer control algorithm, Proc. ACC, San Fransisco

Garcia, C.E. and M. Morari (1982), Internal Model control 1. A unifying review and some new results, I&EC Process Design and Development, 21, pp308-323

Garcia, C.E. , D.M. Prett and M. Morari (1989), Model predictive control: theory and practice, a survey, Automatica, 25, p335-348

Kouvaritakis, B., J.A. Rossiter and A.O.T.Chang (1992), Stable Generalized predictive control: an algorithm with guaranteed stability, Proc IEE, 139, No.4, pp349-362

Kouvaritakis, B., J.R. Gossner and J.A. Rossiter (1996), Apriori stability condition for an arbitrary number of unstable poles, Automatica, Vol. 32, No.10, pp. 1441-1446

Kouvaritakis, B., J.A. Rossiter and M. Cannon (1998), Linear quadratic feasible predictive control, Automatica, 34, 12, pp1583-1592

Rawlings, J.B. and K.R. Muske (1993), The stability of constrained receding horizon control, Trans IEEE AC, 38, pp1512-1516

Richalet, J., A. Rault, J.L. Testud and J. Papon (1978), Model predictive heuristic control: applications to industrial processes, Automatica, 14, 5, pp413-428

Richalet, J., Commande predictive, R 7 423

Rossiter, J.A. (1993), Notes on multi-step ahead prediction based on the principle of concatenation, Proc. IMechE, Vol.207, pp261-263

Rossiter, J.A., B.Kouvaritakis and J.R. Gossner (1995), Feasibility and stability results for constrained stable generalised predictive control, Automatica, 31,6, pp863-877

Rossiter, J.A., J.R. Gossner and B. Kouvaritakis (1996) Infinite horizon stable predictive control, Trans. IEEE AC, 41, 10, pp1522-1527.

Rossiter, J.A., M.J. Rice and B.Kouvaritakis (1998), A numerically robust state-space approach to stable predictive control strategies, *Automatica*, 38, 1, 65-73

Rossiter, J.A. and J. Richalet (2001), Re-aligned models for prediction in MPC: a good thing or not ? APC6 (York)

Rossiter, J.A., Stable prediction for unstable independent models, submitted

Scokaert, P.O.M. and J. B. Rawlings (1998), Constrained linear quadratic regulation, IEEE Trans AC, 43, 8, pp1163-1168

Yoon, T-W., and D.W. Clarke (1995), Observer design in receding horizon predictive control, IJC, 61, pp171-191

Wang, L. (2000), UKACC Control 2000, CD-ROM