
Article:
Rossiter, J.A., Haber, R. and Zabet, K. (2016) Pole-placement Predictive Functional Control for over-damped systems with real poles. ISA Transactions, 61. pp. 229-239. ISSN 0019-0578

https://doi.org/10.1016/j.isatra.2015.12.003

Pole-placement Predictive Functional Control for over-damped systems with real poles

Abstract

This paper gives new insight and design proposals for Predictive Functional Control (PFC) algorithms. Common practice and indeed a requirement of PFC is to select a coincidence horizon greater than one for high-order systems and for the link between the design parameters and the desired dynamic to be weak. Here the proposal is to use parallel first-order models to form an independent prediction model and show that with these it is possible both to use a coincidence horizon of one and moreover to obtain precisely the desired closed-loop dynamics. It is shown through analysis that the use of a coincidence horizon of one greatly simplifies coding, tuning, constraint handling and implementation. The paper derives the key results for high-order and non-minimum phase processes and also demonstrates the flexibility and potential industrial utility of the proposal.

Keywords: Predictive control; PFC; tuning; uncertainty.

1. Introduction

Predictive functional Control (PFC) has been very successful in industry (e.g. [1, 2]) and yet surprisingly received very little interest in the academic literature [3, 10, 11, 15]. A likely reason for this is that researchers in predictive control have focussed on proofs of issues such as guarantees of stability [6, 18] and feasibility, robust stability [4], parametric methods [7] and more recently robust feasibility in the presence of bounded disturbances [9]. It should be noted that PFC techniques are much simpler to code and implement than conventional predictive control methods [3, 8, 15] and thus PFC is best viewed as an alternative to PID or other low level control law where the cost per control law is necessarily small but nevertheless one may desire attributes such as systematic constraint handling. PFC allows systematic as opposed to ad hoc constraint handling and thus is often preferred to PID approaches for scenarios where constraint handling is challenging. Clearly comparisons with conventional predictive control such as Dynamic Matrix Control are not appropriate as those, by definition, can give better performance, but of course at much higher cost.

A main selling point of PFC is that the design is done by choosing a target behaviour (equivalently closed-loop pole/time constant). If there is a strong link between the user choice and the behaviour that results, this is an intuitive and easy design technique, as compared to say PID. Moreover, PFC has a critical advantage over PID approaches, that is, constraint handling can be embedded systematically and with minimum coding/computation. However, the literature has given little attention to theoretical a priori guarantees of stability, feasibility or robustness for PFC; of course some results do exist [12, 17] and industrial users always do practical assessments. This paper seeks to redress the balance slightly by demonstrating some useful new theoretical results for PFC which also extend the efficacy of tuning of the approach.

1.1. Background on PFC

A conventional PFC has two tuning parameters: the position of the coincidence point in the future (the coincidence horizon) and the desired settling time $t_{95}\%$, also called TRBF. It is implicitly assumed that the closed-loop response will become approximately first-order with the target settling time, although in fact this can only be assured if the coincidence horizon is one.

For higher-order aperiodic processes it is common to recommend the coincidence point to be near to the inflection point of the process step response

[11], which is the place of maximum value of the impulse response. The idea is based on the fact that at this point, the manipulated variable change has maximum effect. However, in such a case the actual settling time will often not match the desired settling time t_{95} so tuning becomes more challenging as the direct link between designer choice and effect is lost. One reason is that in case of higher-order processes, the closed-loop response will not approximate a first-order response closely and hence the relation to the settling time is not straightforward [16].

1.2. Paper contributions

This paper will propose an alternative way to tune PFC for systems with real poles. It will be shown that within the proposed PFC design, the closed-loop poles can be selected as free parameters and this is a major advance on conventional PFC where only the slowest closed-loop pole is selected by defining the settling time and even that requires some trial and error and has no guarantee of what is achievable. It is interesting to note that the proposed method, in the unconstrained case, has some equivalence with pole placement design, but a critical point is that it is still based on prediction and allows systematic constraint handling which is not the case for pole-placement designs! It is also noteworthy that the proposed PFC method works with a coincidence horizon of just one, which is contrary to the conventional advice with classical PFC and also enables a significant reduction in computing complexity.

Section 2 gives some background on PFC concepts, a conventional law and demonstrates the tuning challenges. Section 3 introduces the proposed pole placement PFC approach for second-order systems along with some analysis of the properties. Section 4 then generalises the approach to higher-order models and emphasises the additional degrees of freedom which enable more flexible tuning. The paper finishes with numerous examples which demonstrate the attributes of the proposed algorithm and how these compare with conventional PFC.

2. Background information on PFC

2.1. Target behaviour

PFC is based on the assumption that it is realistic to achieve closed-loop behaviour close to a first-order system with a delay τ (or h samples), time

constant T_r and unit gain, for example:

$$r^*(s) = \frac{e^{-s\tau}}{T_r s + 1} r(s); \quad r^*(z) = \frac{z^{-h}(1-\lambda)}{1-\lambda z^{-1}} r(z) \tag{1}$$

where $r(s)$ and $r^*(s)$ are Laplace representations of the reference signal and reference trajectory respectively and $r(z)$ and $r^*(z)$ the corresponding z-transform representations. In the following the reference signal is take to be a step of amplitude r. A typical desired step response, with no delay, is plotted in figure 1 where the pole is set at $\lambda = 0.8$ and the sample period is $T = 1$. Equivalently, industrial users use the notation of target closed-loop response time (CLTR), that is about 3 time constants, where $\lambda = e^{-T/T_r}$ with T the sample period and $T_r = CLTR/3$.

2.2. Coincidence point and degrees of freedom

Assuming the desired closed-loop behaviour is 'known' (as illustrated in figure 1), then the objective is for output predictions $y_p(k+i|k)$ (the predicted value of y_p at sample $k+i$ with prediction made at sample k) to follow this target exactly. Assuming a non-zero initial condition of $y_p(k)$, a first-order response with a known asymptotic value r and decay rate λ can be written down explicitly as follows:

$$y_p(k+i|k) = r - [r - y_p(k)]\lambda^i; \quad i > 0 \tag{2}$$

PFC is not able to make all future predicted output values satisfy (2) and so instead chooses a single sample instant in the future, the so called coincidence horizon n_y, and ensures that the output prediction matches the target response (2) at that point only (as illustrated in figure 1 for $n_y = 5$). Consequently the PFC control law reduces to (conceptually), enforcing the single equality:

$$y_p(k+n_y|k) = r - [r - y_p(k)]\lambda^{n_y} \tag{3}$$

In order to manipulate the predictions, some degrees of freedom (d.o.f.) are needed and these are conventionally the values of the future inputs, $u(k), u(k+1), \cdots$. Within PFC, the coding and computation requirements are deliberately very simple and thus, the predicted future input is taken to be a constant, that is:

$$u(k) = u(k+1|k) = u(k+2|k) = \cdots \tag{4}$$

Thus the only d.o.f. is the proposed value $u(k)$.

Remark 1. *The target behaviour (2) and control law requirement (3) can be coded by inspection. Where the system has a delay 'h', the target behaviour should be modified slightly to:*

$$y_p(k+n_y+h|k) = r - [r - y_p(k+h|k)]\lambda^{n_y-h}.$$

The reader will note that both output terms are based on values h samples further ahead.

A potential weakness of PFC is the simplicity of control law (3). The user needs to be sure that matching a single point implies the rest of the response is also closely matched to the target behaviour. It has been shown [11, 16] that this is the case for first-order systems only. Hence this paper proposes a modified PFC algorithm, such that the result can be extended to some higher-order systems where common understanding is that the best value for n_y can only be found by trial and error (and indeed that assumes a good choice exists). It is notable that for non-minimum phase systems it is intuitively obvious that n_y must be greater than the time of the inverse response part but how much greater is not obvious. This paper shows how such a requirement can be avoided thus enabling more systematic tuning.

2.3. Independent prediction model, dead-time, disturbance estimation and offset free tracking

Within PFC it is conventional to use an independent model for prediction as these are known to have low sensitivity to measurement noise. This means running a model G_m in parallel with the process G_p (see figure 2). The difference $d = y_p - y_m$ between the model output y_m and process output y_p is used as an estimate for the system disturbance and in fact can also be used to account for parameter uncertainty. Replacing process output predictions (which are not known explicitly) with unbiased model output predictions, the PFC control (3) law is restated as:

$$r - [r - y_p(k)]\lambda^{n_y} = y_p(k+n_y|k) = y_m(k+n_y|k) + d(k) \tag{5}$$

An alternative and equivalent representation of this which is slightly easier to code is:

$$[r - y_p(k)](1 - \lambda^{n_y}) = y_m(k+n_y|k) - y_m(k) \tag{6}$$

or in the case of non-zero dead-times of h samples:

$$[r - y_p(k+h|k)](1 - \lambda^{n_y}) = y_m(k+n_y|k) - y_m(k) \tag{7}$$

where one can also substitute for known values using $y_p(k+h|k) \approx y_p(k) + [y_m(k) - y_m(k-h)]$ where $y_m(k)$ is the non-delayed model output.

Remark 2. *Equation (7) shows how PFC handles dead-time processes. It can seen that the right hand side is independent of the dead time so that the model dependent part of the PFC algorithm is valid for any dead time.*

Remark 3. *Readers may like to observe that the use of either (6) or (7) ensures offset free tracking in the closed-loop, for any steady disturbance. This result is standard in the literature, assuming the closed-loop is stable, and is based on the observation that the predictions are unbiased in the steady-state and thus the predicted error converging to zero also implies the actual error must converge to zero. The result carries across throughout this paper due to the use of the same prediction structure.*

2.4. PFC for 1st-order models

A first-order model G_m is presented in discrete-time as:

$$y_m(k) = \frac{bz^{-1}}{1+az^{-1}}u(k) = G_m(z)u(k) \tag{8}$$

The predicted model output n_y steps ahead assuming a constant future $u(k)$ is:

$$y_m(k+n_y|k) = (-a)^{n_y}y_m(k) + K_m[1-(-a)^{n_y}]u(k); \quad K_m = \frac{b}{1+a} \tag{9}$$

Substituting this into control law (6) gives:

$$[r - y_p(k)](1-\lambda^{n_y}) = (-a)^{n_y}y_m(k) + K_m[1-(-a)^{n_y}]u(k) - y_m(k) \tag{10}$$

This can be re-arranged to give the control law:

$$u(k) = \frac{[r-y_p(k)](1-\lambda^{n_y}) + [1-(-a)^{n_y}]y_m(k)}{K_m[1-(-a)^{n_y}]} \tag{11}$$

Remark 4. *Earlier work [16] showed that for first-order processes a coincidence horizon of $n_y = 1$ ensures that the closed-loop dynamic is precisely as desired, that is, the closed-loop pole is exactly λ. The key steps are given*

here for completeness. Assuming nominal case $y_m = y_p = y$, the control law is summarised as:

$$u(k) = \frac{r(1-\lambda) + (1+a-1+\lambda)y(k)}{K_m(1+a)} = \frac{r(1-\lambda) + (\lambda+a)y(k)}{b} \tag{12}$$

Subsituting the $u(k)$ from (12) into the process model (8) one finds the closed-loop dynamic is given from:

$$y(k) = \frac{bz^{-1}}{1+az^{-1}} \frac{[r(1-\lambda) + (\lambda+a)y(k)]}{b} \tag{13}$$

or

$$\underbrace{(1+az^{-1} - (a+\lambda)z^{-1})}_{Closed-loop\ \ pole\ \ polynomial} y(k) = z^{-1}(1-\lambda)r \tag{14}$$

The closed-loop pole is therefore defined as: $pole = -a + (a+\lambda) = \lambda$, hence, one achieves the desired pole exactly if $n_y = 1$.

2.5. PFC for higher-order models having real roots

For higher-order systems it is recommended to use coincidence horizons larger than one [11, 16] due to the inherent lag in the response. However, one cannot easily write down a simple expression for the n_y step ahead prediction [13]. In order to maintain simple coding, PFC overcomes the complexity of prediction algebra by using partial fractions to express a model as a sum of first-order models [3, 5, 11] and hence:

$$G_m(z) = \underbrace{\frac{b_1 z^{-1}}{1+a_1 z^{-1}}}_{G_1} + \cdots + \underbrace{\frac{b_n z^{-1}}{1+a_n z^{-1}}}_{G_n} + \cdots \tag{15}$$

It is noted that in practice multiple poles can be handled as different poles lying near to each other.

Theorem 1. *Assuming the future input is constant, the predictions for models expressed in the form of (15) are trivial and can be written down explicitly.*

Proof: The model output can be constructed as:

$$y_m(k) = G_1(z)u(k) + \cdots + G_n(z)u(k) = y_m^{(1)} + \cdots + y_m^{(n)} \tag{16}$$

An equivalent block diagram is indicated in figure 3. The theorem result then follows immediately from the constructions of the predictions for a single model $y_m^{(i)}(k+1) + a_i y_m^{(i)}(k) = b_i u(k)$ are:

$$y_m^{(i)}(k+n_y|k) = (-a_i)^{n_y} y_m^{(i)}(k) + K_i[1-(-a_i)^{n_y}]u(k); \quad K_i = \frac{b_i}{1+a_i} \tag{17}$$

and $y_m(k+n_y|k) = y_m^{(1)}(k+n_y|k) + y_m^{(2)}(k+n_y|k) + \cdots$. $\square$

Corollary 1. *For a coincidence horizon $n_y = 1$, the predictions are:*

$$y_m(k+1|k) = -a_1 y_m^{(1)}(k) - \cdots - a_n y_m^{(n)} + (\sum_i^n b_i)u(k) \tag{18}$$

The use of such prediction models shows clearly how the complexity and use of PFC is distinguished from more complex strategies such as DMC and is summarised in the following theorem.

Theorem 2. *A PFC algorithm for a higher-order model using a coincidence horizon of n_y and a target pole of λ can be defined explicitly without complicated coding.*

Proof: The prediction of (17) is substituted directly into control law (6) to determine the desired input signal.

$$(1-\lambda^{n_y})(r - y_p(k)) = \textstyle\sum_i [(-a_i)^{n_y} y_m^{(i)}(k) + K_i[1-(-a_i)^{n_y}]u_k - y_m^{(i)}(k)] \tag{19}$$

Rearrange to determine the input as:

$$u(k) = \frac{(1-\lambda^{n_y})(r-y_p(k)) + \sum_i [1-(-a_i)^{n_y}] y_m^{(i)}(k)}{\sum_i K_i(1-(-a_i)^{n_y})} \tag{20}$$

It is clear that the terms in this law are simple to compute. $\square$

Corollary 2. *For a coincidence horizon $n_y = 1$, the PFC law reduces to:*

$$u(k) = \frac{(r-y_p(k))(1-\lambda) + \sum_i (1+a_i) y_m^{(i)}(k)}{\sum_i b_i} \tag{21}$$

Remark 5. *Using parallel forms for prediction requires the partial fraction expansion to be computed. However, for cases with say 2 to 3 poles, this is trivial and this minor extra computation is dwarfed by the gains in the simplification of the prediction algebra.*

2.6. The efficacy of λ as a tuning parameter with conventional PFC

It was shown in section 2.4 that for a first-order model and a choice of $n_y = 1$, then the expected closed loop pole is precisely λ. However, one can equally show [16] that for higher choices of n_y, the closed-loop is not equal to λ and this alone demonstrates that PFC tuning parameters have a weak connection with the behaviour that results. Moreover, with 2nd-order models tuning and coding is not as straightforward and some trial and error is required; indeed there is no guarantee that there exists a suitable choice of n_y such a given λ will be achieved in the closed loop and as such tuning of PFC becomes less systematic than desirable. A point of specific note is that almost always one needs to use $n_y > 1$ and implicitly this forces the behaviour towards open-loop dynamics [16].

It is worthwhile giving some examples of this. Consider two systems H_1, H_3 and determine the closed-loop poles for different choices of λ, n_y and plot the slowest pole in figures 4 and 5 respectively.

$$H_1 = \frac{0.02(4z^{-2} - z^{-1})}{(1 - 0.5z^{-1})(1 - 0.9z^{-1})};\ H_3 = \frac{10^{-4}(3.3239z^{-1} + 0.313z^{-2} - 3.009z^{-3})}{(1 - 0.98z^{-1})(1 - 0.967z^{-1})(1 - 0.951z^{-1})} \tag{22}$$

It is immediately clear that the actual dominant closed-loop pole is well linked to the choice of λ only if n_y is small. What is less clear is that for these cases a choice of $n_y < 10$ may lead to significantly over active input signals (or even instability) and have output responses which are far from ideal. Consequently, for processes with significant lag, conventional PFC can only deliver relatively slow closed-loop dynamics with potentially poor linkage to the choice of the desired closed-loop pole λ.

2.7. Summary

The background on PFC has established a few core insights which will be used as the foundations of the proposal in this paper.

- With first-order models, the use of a horizon of one ensures a strong link between the tuning parameter λ and the closed-loop behaviour which results.
- With higher-order models, there is not expected to be a strong link between the tuning parameters and closed-loop behaviour and thus tuning is less systematic, especially if small values of n_y cannot be used, such as with non-minimum phase or lagged processes [16].

- Using parallel independent models (figure 3) allows for much simpler prediction algebra and thus simpler coding than using a higher-order model for prediction.

3. Easy tuning by using a coincidence horizon of one with 2nd-order models

There is potential to improve the tuning of PFC for higher-order models and ideally to enable a stronger link between the tuning parameter λ and the resulting closed-loop dynamics. This section uses the special case of an over-damped second-order process to show how a small change in the PFC control law can achieve this while not increasing complexity or affecting constraint handling facilities. Specifically, the core concept is to exploit results for first-order models for which the control law (12) can be written down explicitly and for which one can fix the closed-loop pole precisely with $n_y = 1$.

Remark 6. *For the purposes of closed-loop pole analysis this section will consider the nominal case of $y_m = y_p$ and $d(k) = 0$. This is useful because it indicates the efficacy of the tuning method; a formal sensitivity analysis constitutes future work although the examples demonstrate that good robustness is expected.*

3.1. Independent PFC designs with two parallel paths

The concept used here is to design separate PFC control laws for each independent model in figure 3 and then use a linear combination of these as the actual control law applied to the system. For now the focus is on models with two poles.

Lemma 3. *The contribution to the model output steady-states of each parallel model can be captured with variables $\gamma_1, \gamma_2, \quad \gamma_1 + \gamma_2 = 1$ so that:*

$$\lim_{k\to\infty} y_m = r \quad \Rightarrow \quad \begin{cases} \lim_{k\to\infty} y_m^{(1)} = \gamma_1 r \\ \lim_{k\to\infty} y_m^{(2)} = \gamma_2 r \end{cases} \tag{23}$$

Proof: Equation (23) is obvious as $\lim_{k\to\infty}(y_m^{(1)} + y_m^{(2)}) = r$. Moreover, γ can be determined by using the model steady-state gains so that:

$$g_1 = \frac{b_1}{1+a_1}; \quad g_2 = \frac{b_2}{1+a_2}; \quad \gamma_1 = \frac{g_1}{g_1+g_2}; \quad \gamma_2 = 1-\gamma_1 \tag{24}$$

Readers may note there is no need for $g_1 > 0, g_2 > 0$ nor indeed for $\gamma_i > 0$. □

Corollary 3. *A number of simple corollaries fall out from dividing the target between the two parallel paths:*

1. *Equivalent to meeting the target of $y_m \to r$, one can aim for separate targets for each independent model:*

$$y_m^{(1)} \to \gamma_1 r = r^{(1)}, \quad y_m^{(2)} \to \gamma_2 r = r^{(2)}, \tag{25}$$

2. *The impact of the disturbance estimate on the required model predictions can be divided between the subsystems using the steady-state gains so that:*

$$y_m^{(1)} \to r^{(1)} - \gamma_1 d(k); \quad y_m^{(2)} \to r^{(2)} - \gamma_2 d(k) \tag{26}$$

3. *The 'process output' can be divided in an equivalent ratio so that $y_p = y_p^{(1)} + y_p^{(2)}$ and:*

$$y_p^{(1)}(k) = y_m^{(1)}(k) + \gamma_1 d(k), \; y_p^{(2)}(k) = y_m^{(2)}(k) + \gamma_2 d(k). \tag{27}$$

Lemma 4. *Assume for now that each subsystem has an independent input and the disturbance is shared between the subsystems as above, then PFC laws (5) with a coincidence horizon $n_y = 1$ for each parallel subsystem are defined as:*

$$\begin{aligned} b_1 u^{(1)}(k) &= r^{(1)}(1-\lambda) + \lambda y_p^{(1)}(k) - \gamma_1 d(k) + a_1 y_m^{(1)}(k) \\ b_2 u^{(2)}(k) &= r^{(2)}(1-\lambda) + \lambda y_p^{(2)}(k) - \gamma_2 d(k) + a_2 y_m^{(2)}(k) \end{aligned} \tag{28}$$

Proof: This follows directly from the standard PFC law for 1st-order systems assuming the disturbance and measurement are divided between the parallel loops in ratios $\gamma_1, \; \gamma_2$. $\square$

If one were able to apply just $u^{(1)}$ to path 1 and $u^{(2)}$ to path 2, then the model would of course reach the desired steady-state and with the desired time constant. However, the obvious inconsistency is that the actual process has a single input and the independent model formulation of figure 3 assumes that the same input enters both parallel paths. Consequently, the main role of the control laws in (28) is to identify ideal choices; knowledge of these ideal choices can be used to find a compromise which balances the needs of both loops.

3.2. Proposed control law and nominal analysis

Algorithm 1. *Determine $u^{(1)}$, $u^{(2)}$ using the control law definitions of (28) and then form the actual PFC system input as a linear combination of these two.*

$$u(k) = \beta u^{(1)}(k) + (1-\beta)u^{(2)}(k); \tag{29}$$

The rationale for the choice of β will become clear next.

Lemma 5. *For the nominal case, the control laws of (28) can be expressed in z-transforms as follows:*

$$\begin{array}{rcl} u^{(1)}(z) & = & \gamma_1 \frac{1-\lambda}{b_1} r(z) + \frac{(\lambda+a_1)}{b_1} y_m^{(1)}(z) \\ u^{(2)}(z) & = & \gamma_2 \frac{1-\lambda}{b_2} r(z) + \frac{(\lambda+a_2)}{b_2} y_m^{(2)}(z) \end{array} \tag{30}$$

Proof: First replace $y_p^{(1)}(k) = y_m^{(1)}(k), y_p^{(2)}(k) = y_m^{(2)}(k)$ and $d = 0$, hence:

$$b_i u^{(i)}(k) = (1-\lambda) r^{(i)} + \lambda y_m^{(i)}(k) + a_i y_m^{(i)}(k) \tag{31}$$

Rearranging this one finds:

$$b_i u^{(i)}(k) \;=\; (\lambda + a_i) y_m^{(i)} + (1-\lambda) r^{(i)} \tag{32}$$

From which, with $r^{(1)} = \gamma_1 r$ and using a similar statement for loop 2, the result is obvious. □

Corollary 4. *The proposed PFC control law combining (29) and (30) is given as:*

$$u(z) = \beta \frac{(\lambda + a_1)}{b_1} y_m^{(1)}(z) + (1-\beta) \frac{(\lambda + a_2)}{b_2} y_m^{(2)}(z) + K r(z) \tag{33}$$

where $K = (1-\lambda)(\beta\gamma_1/b_1 + (1-\beta)\gamma_2/b_2)$.

The following theorem now summarises a key analytical result and contribution of this paper.

Theorem 6. *The proposed PFC law of (33) achieves the target closed-loop pole of λ and thus is an effective design tool for achieving the desired closed-loop dynamics, irrespective of the choice of β.*

Proof: The first thing to note is that, direct from figure 3, one can deduce the following z-transform relationships:

$$y_m^{(i)}(z) = \frac{b_i z^{-1}}{1 + a_i z^{-1}} u(z) \tag{34}$$

Substitute from (34) into (33) and one finds:

$$u(z) = \beta \frac{(\lambda + a_1)}{b_1} \frac{b_1 z^{-1}}{1 + a_1 z^{-1}} u(z) + (1 - \beta) \frac{(\lambda + a_2)}{b_2} \frac{b_2 z^{-1}}{1 + a_2 z^{-1}} u(z) + Kr(z) \tag{35}$$

Next, combine all terms containing $u(z)$:

$$\begin{array}{c} p_c(z)u(z) = a(z)Kr(z) \\ p_c(z) = [(1 + a_1 z^{-1})(1 + a_2 z^{-1}) - \beta(1 + a_2 z^{-1})(\lambda + a_1) z^{-1} \\ -(1 - \beta)(1 + a_1 z^{-1})(\lambda + a_2) z^{-1}] \\ a(z) = (1 + a_1 z^{-1})(1 + a_2 z^{-1}) \end{array} \tag{36}$$

Validate the closed-loop pole polynomial has a root at λ by showing that $p_c(\lambda) = 0$. $\square$

3.3. Exploiting the flexibility in β

The previous section proved that the proposed control law guarantees that one of the closed-loop poles will be at λ. This section investigates how alternative choices for β impact on the remaining poles and thus gives some insights into how this parameter could be chosen. The key result is summarised in the following theorem.

Theorem 7. *The root of $p(z)$ given in (36) which is not at λ, is independent of the choice of λ and depends solely on the model parameters and the choice of β.*

Proof: Assume that the 2nd root is given as δ, then:

$$p_c(z) = (1 - \lambda z^{-1})(1 - \delta z^{-1}) = 1 - (\lambda + \delta) z^{-1} + \lambda \delta z^{-2} \tag{37}$$

Using eqn.(36) to define the coefficients of $p(z)$ and match up the coefficients of z^{-1}, z^{-2} with (37).

$$\begin{array}{rcl} a_1 + a_2 - \beta(\lambda + a_1) - (1 - \beta)(\lambda + a_2) & = & -\lambda - \delta \\ a_1 a_2 - \beta a_2(\lambda + a_1) - (1 - \beta) a_1(\lambda + a_2) & = & \lambda \delta \end{array} \tag{38}$$

Eliminating and simplifying these both reduce to:

$$(1-\beta)a_1 + \beta a_2 = -\delta \tag{39}$$

That is, δ has an explicit dependence upon β; conceptually δ is a linear combination of the open-loop poles and one can define β from a desired δ or vice versa. □

Remark 7. *In practice the choice of δ will be such that λ is the dominant closed-loop pole.*

3.4. Interim Summary

This section has shown how a slight change to the formulation of PFC has enabled the tuning parameter of λ to be more reliable in that one can guarantee this mode exists in the closed-loop dynamics, which is not the case for a conventional PFC algorithm. Of course, the designer still needs to engage in a full design which considers trade-offs between the target closed-loop dynamic and other criteria such as input activity.

4. Generalisation to higher-order models: Pole placement PFC

The previous section used a second-order example to demonstrate a simple principle, that a PFC approach could be used to give precise closed-loop poles akin to a pole placement design while retaining core PFC properties which enable systematic constraint handling. This section shows how this approach can be extended to higher-order examples.

Lemma 8. *For multiple models in parallel such as in figure 3, the contribution to the overall model G output steady-states of each parallel model can be captured with variables γ_i, so that:*

$$\lim_{k\to\infty} y_m = r \quad \Rightarrow \quad \begin{cases} \lim_{k\to\infty} y_m^{(i)} = \gamma_i r \\ \gamma_i = G_i(1)/G(1) \end{cases} \tag{40}$$

The proof is obvious. Note that $\sum_i \gamma_i = 1$ as $\sum_i G_i(1) = G(1)$.

Algorithm 2. *[Pole placement PFC] The proposed PFC control law (nominal case) combining (29) and (30) for multiple parallel models is given as:*

$$u(k) = \sum_i \beta_i \frac{(\lambda + a_1)}{b_1} y_m^{(i)}(k) + Kr \tag{41}$$

for suitable K and β_i variables to be chosen subject to $\sum_i \beta_i = 1$.

Lemma 9. *The closed-loop pole polynomial for Algorithm 2 is given from:*

$$p_c(z) = 1 - \sum_i \frac{\beta_i(\lambda + a_i)z^{-1}}{1 + a_i z^{-1}} \tag{42}$$

Proof: Substitute the independent model equations $y_m^{(i)} = [b_i z^{-1}/(1+a_i z^{-1})]u(k)$ into the control law of (33). Hence:

$$u(z) = \sum_i \beta_i \frac{(\lambda + a_i)}{b_i} \frac{b_i z^{-1}}{1 + a_i z^{-i}} u(z) + Kr(z) \tag{43}$$

The definition of the closed-loop pole polynomial is now obvious. □

Corollary 5. *The closed-loop pole polynomial of (42) has at least one root at λ as long as $\sum_i \beta_i = 1$. This follows easily by substituting $z = \lambda$:*

$$p_c(\lambda) = 1 - \sum_i \frac{\beta_i(\lambda + a_i)}{\lambda + a_i} = 1 - \sum \beta_i \tag{44}$$

Hence $p_c(\lambda) = 0$ if $1 = \sum_i \beta_i$.

Theorem 10. *The degrees of freedom within β_i are sufficient to place all the poles of $p_c(z)$.*

Proof: Let the desired poles be $\rho_i, i = 1, 2, ...$ and let $\rho_1 = \lambda$. Then, these are closed-loop poles iff $p_c(z)$ can be expanded as follows:

$$1 - \sum_i \beta_i \frac{(\lambda + a_i)}{b_i} \frac{b_i z^{-1}}{1 + a_i z^{-i}} = \prod_i \frac{1 - \rho_i z^{-1}}{1 + a_i z^{-1}} = 1 + \sum_i \frac{\eta_i z^{-1}}{1 + a_i z^{-1}} \tag{45}$$

Using a cover-up rule to determine suitable values for η_i gives:

$$\eta_j = \frac{\prod_i(-a_j - \rho_i)}{\prod_{i,i\neq j}(-a_j + a_i)} = -\beta_j(\rho_1 + a_j) \tag{46}$$

The key point here is that values of η_i can always be computed and there is a simple dependence of β_i upon ρ_i, that is one can choose ρ_i and then find the required β_i. □

Corollary 6. *If a desired pole ρ_i is specified as equal to an open-loop pole $-a_i$, then this implies $\beta_i = 0$. This is obvious from (46) and also makes good intuitive sense - the corresponding open-loop dynamic is in effect uncontrolled if the independent model output $y_m^{(i)}$ plays no part in the control law.*

5. Numerical examples

This section gives some simulation examples of the proposed pole placement PFC law [denoted PP-PFC] (33) and compare with conventional PFC [denoted CPFC]. These examples demonstrate that:

- the closed-loop dynamic of PP-PFC is directly affected by the choice of λ whereas this is difficult with CPFC.
- the PP-PFC method is robust to non-zero disturbances and some parameter uncertainty.
- the PP-PFC method allows systematic constraint handling.
- the PP-PFC method can deal with non-minimum phase processes easily with $n_y = 1$ (CPFC is far less flexible).

5.1. Illustrations of the impact of λ on closed-loop behaviour

This section will consider second and third-order examples H_2, H_3 respectively.

$$H_2 = \frac{0.1z^{-1} + 0.4z^{-2}}{(1 - 0.5z^{-1})(1 - 0.9z^{-1})}; \quad H_3 = \frac{10^{-4}(3.3239z^{-1} + 0.313z^{-2} - 3.009z^{-3})}{(1 - 0.98z^{-1})(1 - 0.967z^{-1})(1 - 0.951z^{-1})} \tag{47}$$

The figures will compare the closed-loop behaviour with the conventional PFC control law of (20) with the proposed pole placement PFC law of (41). The simulations include an additive output disturbance (around sample 31 in figure 6) to demonstrate that both methods are robust to disturbances.

5.1.1. Example H_2

A challenging scenario would be a substantial speed up of the process to give a closed-loop pole at $\rho = 0.3$. With CPFC a search of various possible tuning parameters demonstrates that although one can speed the process up, the actual closed-loop poles cannot be selected precisely, even with trial and error search over λ, n_y as seen in section 2 and here:

$$\begin{array}{lcl} \{\lambda = 0.3, n_y = 2\} & \Rightarrow & \rho_{1,2} = 0.25, 0.06 \\ \{\lambda = 0.3, n_y = 3\} & \Rightarrow & \rho_{1,2} = 0.42 \pm j0.16 \\ \{\lambda = 0.2, n_y = 3\} & \Rightarrow & \rho_{1,2} = 0.42 \pm j0.018 \end{array} \tag{48}$$

Clearly however PP-PFC can deliver the desired pole precisely - in this case the user choices are:

$$\rho_{1,2} = 0.3, 0.3; \quad a_1 = 0.9, \quad a_2 = 0.5, \quad \beta_1 = 1.5, \quad \beta_2 = -0.5 \tag{49}$$

The closed-loop simulations are given in figure 6 and demonstrate that although the responses are similar, PP-PFC has given behaviour much closer to the target dynamic. A further demonstration of the efficacy of PP-PFC is given in figure 7 which shows that the algorithm reliably gives the desired dynamic as $\rho_1 = \rho_2 = \rho$ are changed.

5.1.2. Example H_3

In this case the aim is to place a closed-loop pole at $\rho = 0.8$, that is to roughly double the speed. With CPFC a search of various possible tuning parameters demonstrates that although one can speed the process up, the *dominant* closed-loop poles cannot be selected precisely, even with trial and error search over λ, n_y as seen in section 2 and here:

$$\begin{array}{lcl} \{\lambda = 0.8, n_y = 1\} & \Rightarrow & \rho_{1,2,3} = 0.8, 0.9, 0.999 \\ \{\lambda = 0.8, n_y = 3\} & \Rightarrow & \rho_{1,2,3} = 0.5, 0.8, 0.9 \\ \{\lambda = 0.8, n_y = 8\} & \Rightarrow & \rho_{1,2,3} = 0.87 \pm j0.1, 0.91 \end{array} \tag{50}$$

Clearly however PP-PFC can deliver the desired three equal poles precisely. The closed-loop simulations are given in figure 8 and demonstrate that although the responses are similar, PP-PFC has given behaviour much closer to the target dynamic. Clearly nothing is for free and the speed up is paid for by a relatively aggressive input.

5.2. Dealing with parameter uncertainty

This section demonstrates that the PP-PFC algorithm is robust to some parameter uncertainty - a detailed sensitivity analysis forms future work. Take system H_2 as the system model G_m and the real process G_p as follows:

$$G_p = \frac{0.12z^{-1} + 0.37z^{-2}}{(1 - 0.4z^{-1})(1 - 0.92z^{-1})} \tag{51}$$

The closed loop behaviour for two choices of λ are given in figures 9 and 10; a disturbance d is also added at sample 40. The figures show clearly that the large difference between the model output y_m and actual process output y_p does not affect the offset free tracking property. These figures demonstrate clearly the robust behaviour of the algorithm and the continuing efficacy of λ as a tuning parameter.

5.3. Non-minimum phase systems and delays

The proposed PP-PFC technique works for non-minimum phase systems (here $H_4(z)$ given in (52)) whereas a CPFC approach requires $n_y \gg 1$ and λ is less effective. Figure 11 shows the closed-loop PP-PFC behaviour matches the target pole λ despite the request for a significant speed up. CPFC has to use a large coincidence horizon for this case and hence can only be used to obtain poles close to the open-loop dynamics.

$$H_4 = \frac{-0.2z^{-1} + 0.6z^{-2}}{(1 - 0.5z^{-1})(1 - 0.9z^{-1})} \tag{52}$$

For completeness, it is noted that non-zero delays can be included in a standard fashion as indicated in eq.(7). Figure 12 gives the same example as figure 11 but where model H_4 now has a dead-time of 4 samples. It is noted that the proposed PP-PFC algorithm is equally effective.

5.4. Constraint handling

A major reason for preferring PFC to PID, even with low-order systems where achievable closed-loop behaviour may be similar, is the constraint handling facility which is cumbersome to code and test with PID but elementary with PFC.

The proposed control law for PP-PFC given in (41) has a simple facility for including constraint handling summarised as follows:

1. Test whether the proposed input satisfies input absolute and rate constraints. If not, modify $u(k)$ to ensure both using saturation.
2. To ensure satisfaction of output/state constraints one must form the implied predictions of (16) over a sensible but large horizon and modify $u(k)$ as required to ensure satisfaction. This reduces to a simple *for loop* which ensures, for example, that $\max_i y_p(k+i|k)$ is within limits; as y_p has a linear dependence on $u(k)$ this is easy to do.

Figure 13 shows the PP-PFC algorithm successfully integrates input constraints of $-2 < u(k) < 2$ and continues to deliver good performance for system H_3.

Remark 8. *It is non-trivial to guarantee the satisfaction of state constraints in general, especially in the presence of disturbances and parameter uncertainty and algorithms for achieving this have substantial coding requirements*

and complexity and thus are inconsistent with the context of a PFC approach. Nevertheless, the PFC approach guarantees recursive feasibility of state constraints for the nominal case which is still a strong result.

6. Summary of coding requirements and applications on hardware

For completeness this section looks at how the PP-PFC algorithm would be coded and demonstrates efficacy on real hardware.

The coding complexity of the proposed algorithm is simple compared to common alternatives, and can be implemented in just a few (5 to 10) lines of code.

OFFLINE COMPUTATIONS

1. Find partial fraction expansion of (15); this is trivial using the cover-up rule[1].
2. Choose desired values for ρ_i and use (46) to determine β_i.
3. Determine parameters γ_i as in (40).

ONLINE COMPUTATIONS (each sample)

1. Update model states: $y_m^{(i)}(k+1) = -a_i y_m^{(i)}(k) + b_i u(k)$.
2. Update $d(k)$ estimate using: $d(k) = y_p(k) - \sum_i y_m^{(i)}(k)$.
3. Calculate $u(k)$ using (41).
4. Ensure $u(k)$ satisfies any relevant constraints before implementing.

7. Real time experiment

7.1. Description of the pilot plant

This section illustrates the efficacy of the proposed method on a hot air blower pilot plant (AMIRA LTR-701) (see figure 14). The air is sucked in through a radial fan and flows through a throttle valve for heating. The air is heated and then flows through a tube. The input signal is the heating power and the output signal is the temperature at the end of the tube. The signals ranges are 0 to 10 V DC or 4 to 20 mA.

The temperature change of the tube is neglected. Then the air temperature $\vartheta(t)$ depends on the heating power$P(t)$, the mass flow $\dot{m}_L(t)$, the

[1]We exclude discussion of systems with repeated poles for brevity. In practice, following an identification, real systems can always be modelled with slightly different poles anyway.

stationary air mass M in the tube, and the specific heat capacity of air c_{pL} according to

$$\dot{m}(t)c_p\Delta\vartheta(t) + Mc_p\frac{d\vartheta(t)}{dt} = P(t) \tag{53}$$

This equation represents a proportional first-order model without time delay as

$$\frac{d\Delta\vartheta(s)}{P(s)} = \frac{K}{1+sT} \tag{54}$$

The proportional gain K and the time constant T depend on the flow

$$K = \frac{1}{\dot{m}(t)c_p}; \quad T = \frac{M}{\dot{m}(t)} \tag{55}$$

The linearized model was assumed for small changes around an operating point.

7.2. Process identification

As the parameters of the model above cannot be determined precisely, the model structure is used alongside a standard identification algorithm to identify representative model parameters between the manipulated variable of the heater u_T and the temperature signal y_T at the end of the tube were identified at a sampling time of 2s. The flow of the air was kept constant by the manipulated variable of the radial fan with $u_F = 3V$, and the heating power u_T was changed stepwise from 1 to 3 V. The step-response of the process is plotted in figure 15.

No dead time was observed during the identification process. The identified model parameters using the least squares method of the measured data filtered by a low-pass filter with time constant $T_f = 10s$ were

$$y_m(z) = \left[\frac{0.23z^{-1}}{1-0.71z^{-1}} + \frac{0.0186z^{-1}}{1-0.986z^{-1}}\right]u(z) \tag{56}$$

7.3. Real-time control

The proposed PFC is designed uses poles $\rho_{1,2} = 0.98, 0.9$. For the real-time control (figure 16), the reference is changed stepwise from 2.9V to 6V and a disturbance was introduced around 1300s by increasing the flow rate substantially between 3 and 7 V. The proposed PFC control law has been effective at obtaining the desired response characteristics and has robust behaviour.

8. Conclusions and future work.

Although PFC is successful in many scenarios, the underlying requirement of making a higher-order system take on output predictions close to a first-order system can give rise to inconsistencies where there are significant lag or non-minimum phase characteristics and thus tuning by trial and error is difficult and indeed the achievable results are limited.

This paper has proposed a novel PFC law which exploits the individual dynamics making up the system model and thus is able to use a coincidence horizon of just one. The proposed PP-PFC approach is effective and simple to tune on high-order systems with simple poles, even though this is in contradiction to the normal guidance of large coincidence horizons being essential. This result is significant because of two main factors:

1. The use of a coincidence horizon of one means that the coding and prediction is trivial; implementation requires just 5 to 10 simple instructions.
2. Control tuning for conventional PFC requires trial and error and often cannot deliver the desired dynamics due to the limited flexibility of the approach. Conversely, the proposed pole placement PFC algorithm has more flexibility with no overall increase in coding complexity, combined with a more intuitive design procedure, as no trial and error is required.
3. It is notable that the proposed approach is able to deal with non-minimum phase characteristics much better than CPFC and thus has far more flexibility in tuning the resultant closed loop.

The proposed approach retains the important attributes of robustness to uncertainty and in particular systematic constraint handling and thus is more flexible than the obvious alternatives of PID or pole-placement for which systematic constraint handling is much more difficult. Efficacy has been demonstrated on numerous examples and laboratory hardware. It would be interesting to pursue in more detail a study of the sensitivity of this proposed approach and how it compares to alternatives such as conventional PFC and PID and this constitutes immediate future work although the authors are cautious about putting too much reliance on the multitude of robust approaches in the more general MPC literature as these methods largely conflict with the requirement for computational simplicity required for low level control.

Future work will consider the extension of this concept to systems with more demanding dynamics. Preliminary work indicates that the extensions

to deal with complex poles are straightforward with almost no changes to the work in this paper. However, more careful consideration is needed to deal with open-loop unstable systems as a simple internal model cannot be used in this case. For example, conventional PFC requires changes in the underlying loop structure and prediction models to cater for unstable open-loop plants. Within the broader MPC literature ideas such as prestabilisation [14, 17] are commonplace. There us also minor interest in what happens when targets take the form of ramps rather than steps; again this requires a significant change in the underlying approach as the use of two integrators within the loop is now implicit in order to achieve no offset and this radically changes the performance that is achievable, as is well known from the mainstream literature.

Finally there is a need to consider the extent to which one can exploit the additional flexibility available within the choice of the other closed-loop poles; as these positions come from simple linear combinations of existing poles it is expected that a common sense default will be possible.

References

[1] Changenet, C., Charver, J.N., Gehin, D., Sicard, F. and Charmel, B., Predictive Functional Control of an expansion valve for controlling the evaporator superheat, IMechE Journal of Systems and Control Engineering, 222 (6), pp571–582, 2008.

[2] Fallasohi, H., Ligeret, C. and Lin-shi, X., Predictive Functional Control of an expansion valve for minimizing the superheat of an evaporator, International Journal of Refrigeration, 33(2), pp409–418, 2010.

[3] Haber, R., Bars, R. and Schmitz, U., Predictive Control in Process Engineering: From the Basics to the Applications, Chapter 11: Predictive Functional Control, Wiley-VCH, Weinheim, Germany, 2011.

[4] Kerrigan, E.C., Robust Constraint Satisfaction: Invariant Sets and Predictive Control. PhD thesis, University of Cambridge, UK, 2000.

[5] Khadir, M.T., and Ringwood, J.V., Extension of first order predictive functional controllers to handle higher order internal models, Int. J. Appl. Math. Comput. Sci., 18(2), pp229-239, 2008.

[6] Kouvaritakis, B., Rossiter, J.A., and Chang, A.O.T., Stable generalized predictive control: an algorithm with guaranteed stability, Proc IEE, 139(4), pp349–362, 1992.

[7] Kvasnica, M. (2009). Real-time model predictive control via multi-parametric programming: theory and tools. VDM-Verlag.

[8] Maciejowski, J., Predictive control with constraints, Prentice Hall, 2001.

[9] Mayne, D.Q., Seron, M.M. and Rakovic, S.V., Robust model predictive control of constrained linear systems with bounded disturbances, Automatica, 41 (2), pp219-224, 2005.

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

[11] Richalet J. and ODonovan, D., Predictive functional control Principles and industrial applications, Springer-Verlag, 2009, London, England.

[12] Richalet, J., Lavielle, G. and Mallet, J., Commande Predictive, Eyrolles, 2004.

[13] Rossiter, J.A., Notes on multi-step ahead prediction based on the principle of concatenation, Proceedings IMechE., 207, pp261–263, 1993.

[14] Rossiter, J.A.; Rice, M.J.; Kouvaritakis, B. A numerically robust state-space approach to stable predictive control strategies, Automatica, 1998, 34, pp65–73.

[15] Rossiter, J.A., Model predictive control: a practical approach, CRC press, 2003.

[16] Rossiter, J.A. and Haber, R., The effect of coincidence horizon on predictive functional control, Special issue on Process Control: Current Trends and Future Challenges, Processes, 2015, 3(1), pp25–45, DOI: 10.3390/pr3010025

[17] Rossiter, J.A., Input shaping for PFC: how and why? Journal of control and decision, DOI: 10.1080/23307706.2015.1083408, 2015.

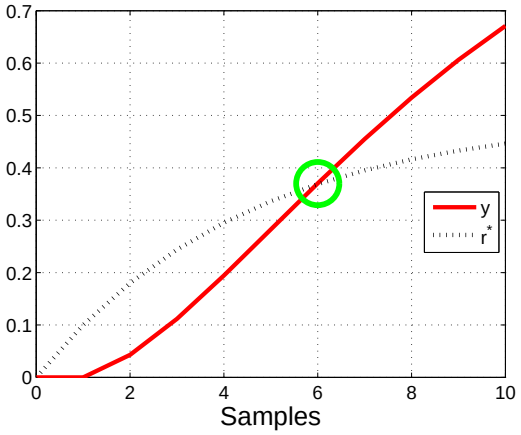

Figure 1: Target step response $r^*(z)$ with pole of 0.8 and illustration of coincidence with output prediction y at a coincidence horizon of 6.

[18] Scokaert, P.O. and Rawlings, J.B., Constrained linear quadratic regulation. Automatic Control, IEEE Trans. on AC, 43(8), 1163–1169, 1998.

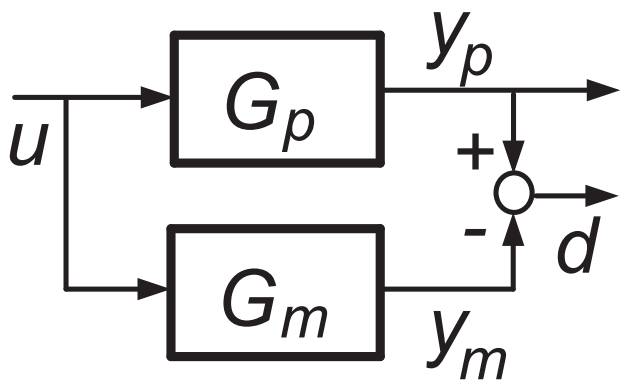

Figure 2: Independent model simulation structure.

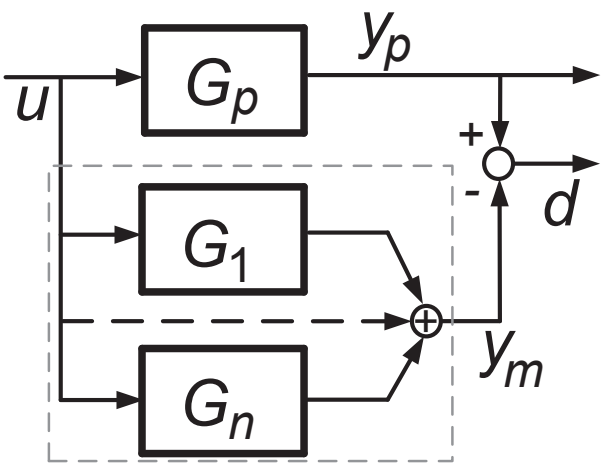

Figure 3: Independent model structure with partial fraction expansion.

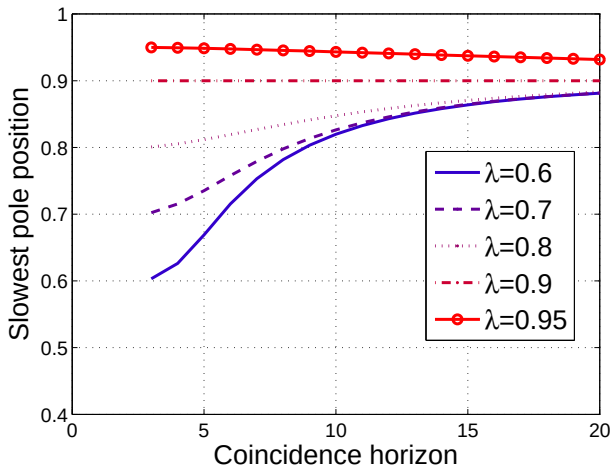

Figure 4: Relationship between tuning parameters and closed-loop poles for conventional PFC on H_1

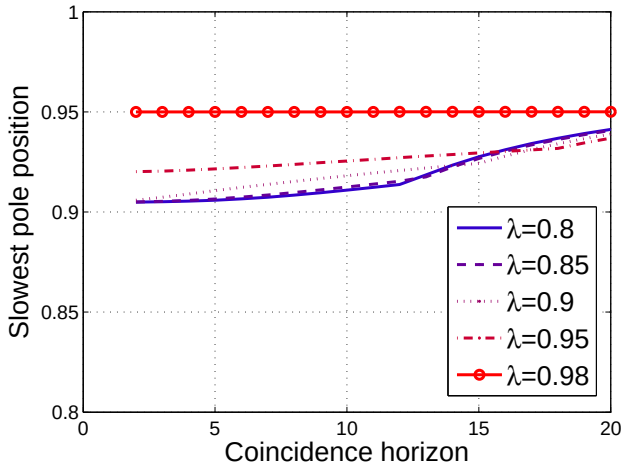

Figure 5: Relationship between tuning parameters and closed-loop poles for conventional PFC on third-order model H_3.

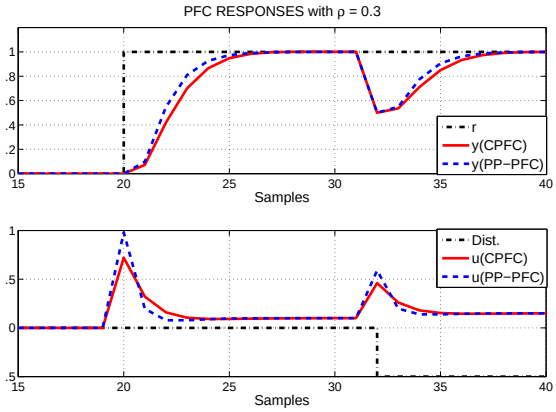

Figure 6: Simulation of model H_2 with a target pole of $\lambda = 0.3$ using CPFC ($n_y = 3$) and PP-PFC ($\rho_1 = \rho_2 = \lambda$).

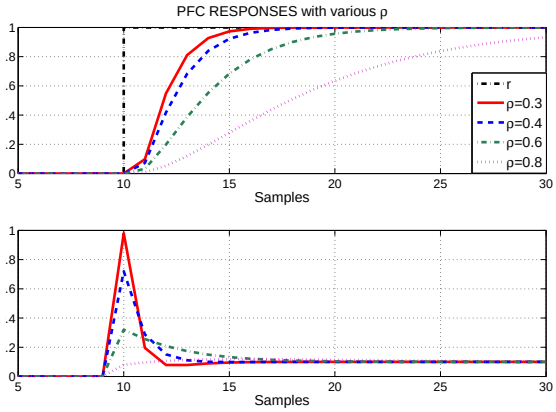

Figure 7: Simulation of model H_2 with PP-PFC using different double target poles.

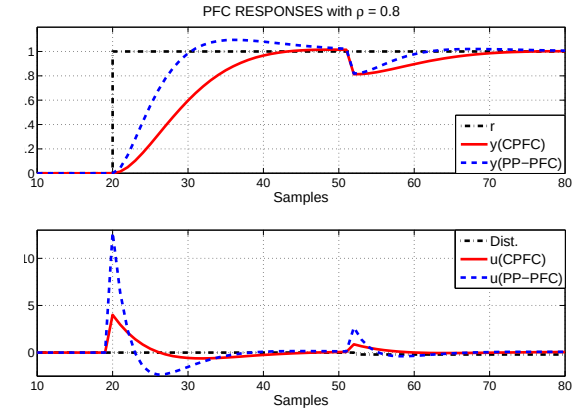

Figure 8: Simulation of model H_3 with a target pole of $\lambda = 0.8$ using CPFC ($n_y = 8$) and PP-PFC ($\rho_1 = \rho_2 = \lambda$).

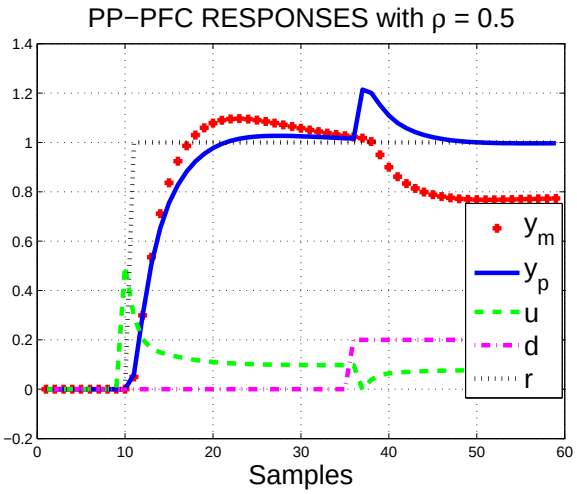

Figure 9: Simulation of model H_2 and with parameter uncertainty (real process given in (51)) and a double target pole of $\rho = 0.5$.

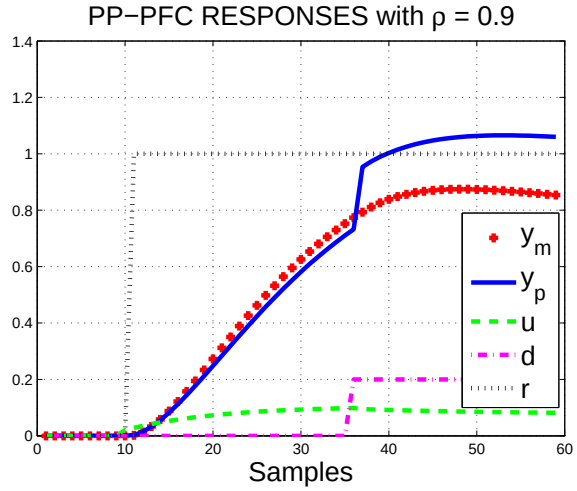

Figure 10: Simulation of model H_2 and with parameter uncertainty (real process given in (51)) and a double target pole of $\rho = 0.9$.

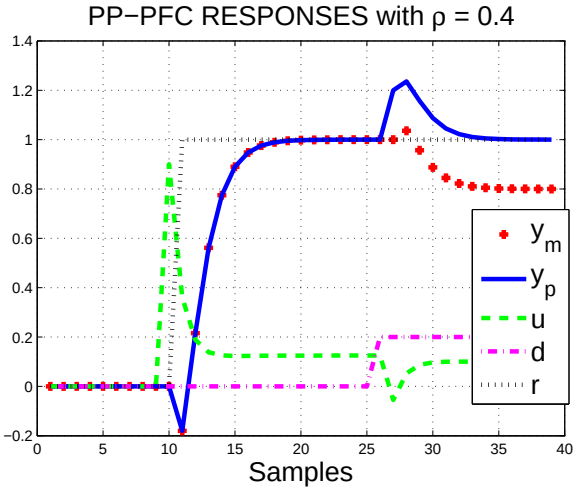

Figure 11: Simulation of non-minimum phase system H_4 with a double target pole of $\rho = 0.4$.

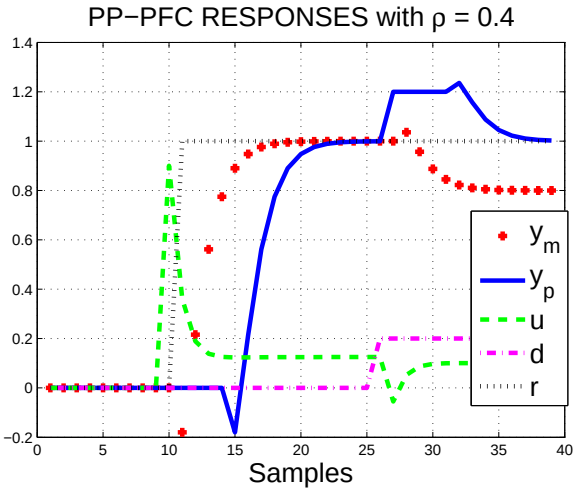

Figure 12: Simulation of non-minimum phase system H_4 with a double target pole of $\rho = 0.4$ and the addition of a significant dead-time to the process.

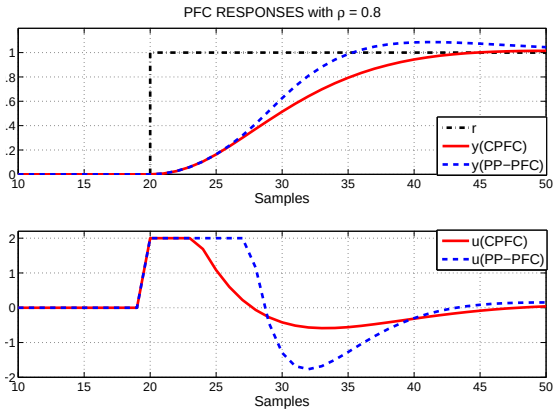

Figure 13: Simulation of system H_3 with input constraints and a triple target pole of $\rho = 0.8$.

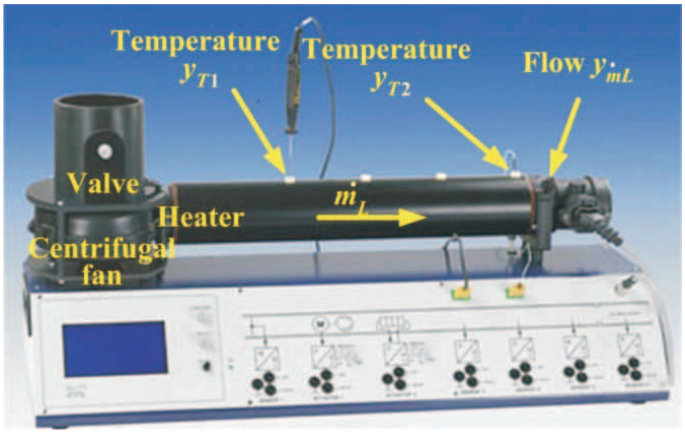

Figure 14: Hot air blower AMIRA LTR-701.

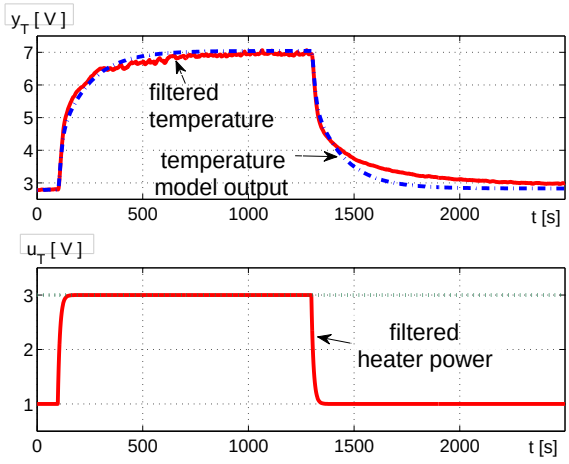

Figure 15: Pilot plant identification

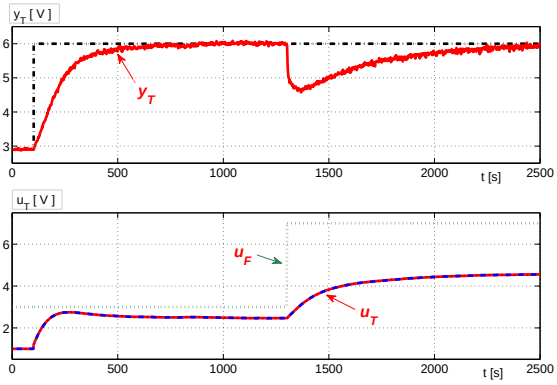

Figure 16: Temperature control