Processes **2015**, *3*, 25-45; doi:10.3390/pr3010025

OPEN ACCESS

processes

ISSN 2227-9717

www.mdpi.com/journal/processes

Article

The Effect of Coincidence Horizon on Predictive Functional Control

John Anthony Rossiter [1],* and Robert Haber [2]

[1] Department of Automatic Control and Systems Engineering, University of Sheffield,
Mappin Street, Sheffield S1 3JD, UK

[2] Institute of Process Engineering and Plant Design, Laboratory of Process Control,
Cologne University of Applied Sciences, Betzdorfer Str. 2, D-50679 Koln, Germany;
E-Mail: robert.haber@fh-koeln.de

* Author to whom correspondence should be addressed; E-Mail: j.a.rossiter@sheffield.ac.uk;
Tel.: +44-114-222-5685.

Academic Editor: Gabriele Pannocchia

Received: 15 October 2014 / Accepted: 12 December 2014 / Published: 8 January 2015

Abstract: This paper gives an analysis of the efficacy of PFC strategies. PFC is widely used in industry for simple loops with constraint handling, as it is very simple and cheap to implement. However, the algorithm has had very little exposure in the mainstream literature. This paper gives some insight into when a PFC approach is expected to be successful and, conversely, when one should deploy with caution.

Keywords: PFC; tuning; model predictive control

1. Introduction

Predictive control [1,2] is widely adopted in industry due to its ability to handle constraints, delays and interactivity in an intuitive, as well as effective manner. However, it is also noted that most applications of predictive control (MPC) are very expensive and, thus, restricted to relatively large or whole unit control operations. This observation is somewhat depressing, given the prevalence of conventional strategies, such as PID, even for loops where PID clearly does not work particularly well; staff tend to prefer to use what they know and understand, even when performance is relatively poor compared to what one could obtain.

Of course, there is a major exception to this, and that is predictive functional control (PFC) [3,4], which has a very significant penetration into markets where classical methods, such as PID, predominate. The success is largely due to a few significant attributes:

1. The energy and enthusiasm of key staff alongside the intuitive nature of the approach has enabled effective promotion of the benefits within industry.
2. The inclusion of PFC on core curricula [5] for potential plant operators, so that they are familiar with how to tune PFC and, thus, view it as an obvious alternative to PID.
3. PFC has costs similar to PID, whereas popular alternative MPC algorithms are neither cheap and/or not embedded in local hardware or software. Moreover, PFC handles constraints and delays much more easily than PID approaches.
4. A default PFC set-up handles ramp and parabolic targets, whereas such scenarios have not captured the attention of the more mainstream MPC literature, despite many practical scenarios having such targets.
5. Because the concept is so simple, it can easily be applied on non-linear systems, which is certainly not true for most MPC algorithms.

There appears to be little formal analysis of PFC, guarantees of performance and the like within the literature, whereas such properties can be obtained for many predictive control algorithms, in particular the popular dual-mode approaches [6,7]. Consequently, there seems to be a clear need in the literature for some analysis of PFC to state things like:

1. Can the tuning be systematic?
2. Can one give any form of guarantee or expectation of good closed-loop performance?
3. How can we make PFC robust against model parameter uncertainty?
4. Can one identify scenarios where PFC will work well and, conversely, may not?
5. Are there simple adaptations to PFC that would extend the range of scenarios for which it works well?

This paper will offer some useful insights on what might be considered the more elementary and core applications of PFC; later work will tackle more challenging application issues.

Section 2 gives a brief review of the core concepts underpinning a PFC approach and demonstrates that the coding is trivial (at least for the linear case). Section 3 focuses on first order systems and demonstrates that, for this case, one can derive some rigorous proofs of stability and performance. Section 4 considers higher order systems and shows that for a wide range of open-loop dynamics, including some challenging ones, the use of very simple figures gives an intuitive and simple insight into how PFC can be tuned and, indeed, the range of closed-loop behaviours one can reasonably expect. Section 5 then focuses on some cases where a simple PFC design does not work well and is not easy to tune systematically, or at all. Practitioners in PFC have solutions for many such cases, but the implementation is more involved and, thus, beyond the remit of this paper and, indeed, emphasises the fact that a simplified approach is not enough. The paper finishes with some conclusions and a discussion of opportunities for further studies. In order to retain a clear focus, this paper does not look explicitly at non-step targets and analysis for non-linear systems, and clearly, that is one important avenue for further work.

2. Basic Concepts of PFC

This section reviews the main parts of PFC and how such an algorithm could be coded. For now, the emphasis is on PFC for tracking step targets; later work will consider the additional issues linked to tracking ramps and parabola. Furthermore, for convenience, as this does not impact on fundamental concepts, tuning or analysis, this paper excludes the algebraic details required to ensure offset correction, although those are included in all of the simulation examples. It is known that PFC automatically includes integral action, and this enables offset free tracking in the presence of small levels of uncertainty. Uncertainty is usually accounted for by adjusting predictions using a term d or disturbance estimate, which is the difference between a model output and the process.

2.1. Target Behaviour or Reference Trajectory

A key concept underlying PFC is that one would like to achieve closed-loop behaviour close to a first order system with a delay t_d seconds (or m samples), and thus, a key requirement is to identify an input sequence that will achieve this. The target closed-loop behaviour (or reference trajectory) can be expressed as:

$$r^{tr}(s) = \frac{e^{-st_d}}{T_r s + 1} R; \quad r^{tr}(z) = \frac{z^{-m}(1-\rho)}{1-\rho z^{-1}} R \tag{1}$$

where $r^{tr}(s)$ represents the continuous time or Laplace representation and $r^{tr}(z)$ the discrete time or z-transform representation; R is taken to be the actual steady-state target. A typical desired step response, with no delay, is plotted in Figure 1, where, here, the discrete pole is set at $\rho = 0.8$.

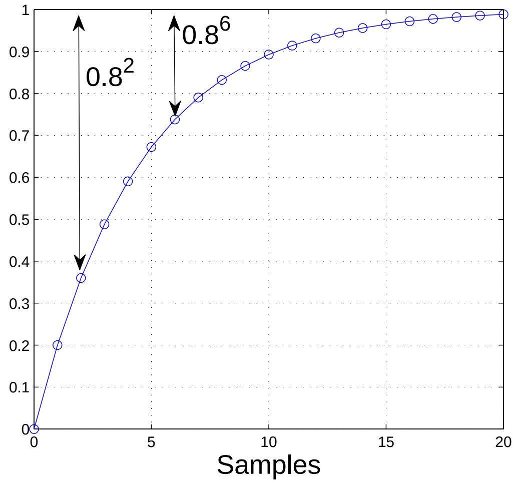

Figure 1. Target closed-loop step response with a pole of 0.8 and no delay.

The discrete pole ρ and target time constant T_r are related through the sampling time T_s as $\rho = e^{-\frac{T_s}{T_r}}$. Some authors prefer to define a target settling time, such as the time it takes to get within 5% of the target (denoted closed-loop time response (CLTR)) and note that $T_r \approx CLTR/3$.

2.2. Coincidence Point

The objective is to make the process behave like a pre-specified first order system. This means the desired behaviour is "known". For example, assuming no delay to simplify the algebra for now and taking y_k to be the current output and R to be the steady-state target, one wants the future outputs to follow a first order response from the current value to the target, in other words, to take the following values:

$$y_{k+i} = R - [R - y_k]\rho^i; \quad i > 0 \tag{2}$$

For step targets, PFC chooses a single sample instant in the future, the so-called coincidence horizon n_y, and ensures that the actual output prediction matches the target response at that point only. Consequently, the PFC control law reduces to (conceptually), enforcing the equality:

$$y_{k+n_y|k} = R - [R - y_k]\rho^{n_y} \tag{3}$$

where the notation $y_{k+n|k}$ means the predicted value of y at sample $k + n$, where the prediction is made at sample k.

Figure 2 illustrates conceptually what is going on. The target trajectory r^{tr} is marked with a dotted line. One forces the output prediction (solid line) to "coincide" at a specified horizon (here, 10 samples into the future). The computation is updated at every sample instant, and this introduces feedback.

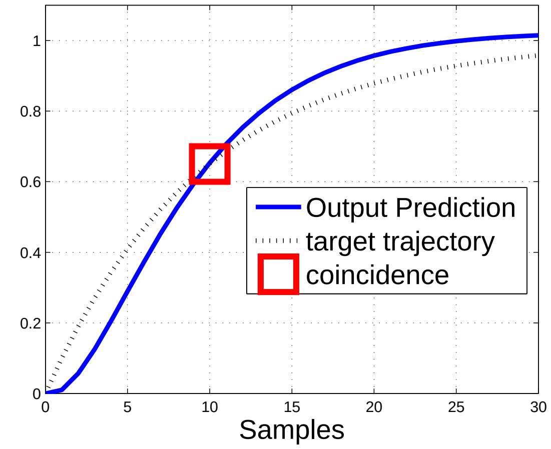

Figure 2. Conceptual illustration of PFC control law definition matching prediction to target trajectory at a single point.

Remark 1. *The target behaviour Equation (2) and control law requirement Equation (3) can be written down by inspection. Clearly, Equation (2) is simple. Where the system has a delay of "m" samples, the target behaviour should be modified slightly, as discussed in Section 2.4.*

Remark 2. *One can argue that PFC is intuitive in that one in essence proposes a control move that gets the system output closer (than the current output) to the desired steady state some time in the near future and then continually update this, so in effect, one expects to always be getting closer and closer to the target. However, the main weakness of PFC is the same as its strength. Basing control decision making on a single point will work well if matching that single point implies the rest of the response is also fairly closely matched, as this will imply consistent decision making at each sample. However, conversely, if matching a single point does not imply matching of other parts of the response, then the implied decision making could be seriously flawed, as it could lead to inconsistent decisions from one sample to the next. In simple terms, one needs to be sure that the part of the prediction beyond the coincidence point moves smoothly and monotonically to the steady state.*

2.3. Prediction and the PFC Control Law

This section presents PFC for linear discrete models (single input and single output). The decision variable is taken to be the value of the future system input. Moreover, for the purposes of prediction, the system input is assumed to be constant in the future:

$$u_k = u_{k+1|k} = u_{k+2|k} = \cdots \tag{4}$$

so implicitly, within the predictions, one is "expecting" or utilising behaviour close to open-loop dynamics.

2.3.1. First Order Models with Open-Loop Dynamics Equal to Target Dynamics

For a first order model and constant input, prediction is very simple.

$$\left.\begin{array}{c}(1+a_1z^{-1})y_k = b_1z^{-1}u_k \\ u_k = u_{k+1} = \cdots\end{array}\right\} \quad \Rightarrow \quad y_{k+i|k} = \frac{b_1}{1+a_1}(1-(-a_1)^i)u_k + (-a_1)^i y_k \tag{5}$$

For the first order case only, we write the denominator as $1+a_1z^{-1} = 1-az^{-1}$, that is $a = -a_1$, as this makes the prediction algebra of Equation (5) easier and more transparent; also, for simplicity, $b = b_1$. Combining this prediction with the coincidence requirement of Equation (3) results in the following equality or essence, the control law.

$$R-(R-y_k)\rho^{n_y} = \frac{b}{1-a}(1-a^{n_y})u_k + a^{n_y}y_k \quad \Rightarrow \quad u_k = \frac{(R-(R-y_k)\rho^{n_y} - a^{n_y}y_k)(1-a)}{b(1-a^{n_y})} \tag{6}$$

Theorem 1. *If the target pole ρ is chosen equal to the open-loop pole a, that is $a = \rho$, then the closed-loop pole will match the open-loop pole for any choice of n_y (in the nominal case).*

Proof. The PFC control Equation (6) is summarised as follows when $\rho = a$:

$$u_k = \frac{[R(1-\rho^{n_y}) + (\rho^{n_y}-\rho^{n_y})y_k](1-a)}{b(1-\rho^{n_y})} = \frac{R(1-\rho^{n_y})(1-a)}{(1-\rho^{n_y})b} = \frac{1-a}{b}R = \frac{R}{K} \tag{7}$$

where, in effect, K represents the system steady-state gain. As expected, this suggests that input is moved directly to the expected steady-state and is not dependent on the current output. This observation is irrespective of the choice of coincidence horizon n_y. Clearly if the input selection is a constant, open-loop dynamics will follow. □

2.3.2. First Order Models with Open-Loop Dynamics Different from Target Closed-Loop Dynamics

As the prediction of the output at the coincidence point is based on an assumption of constant input, for large coincidence horizons, it is not intuitively reasonable to request closed-loop behaviours much different from open-loop behaviours, as this will give a significant inconsistency between the assumed input predictions and the actual desired input behaviour to give the required response; in other words, faster responses need over-actuation in transients. Nevertheless, a basic premise of PFC is that one can choose a target pole, so it is worth doing some analysis of the extent to which this really can work effectively. Specifically, we are interested in the extent to which the user selection of ρ is meaningful in that the resulting closed-loop dynamic is close to ρ.

Theorem 2. *For first order models, the link between the desired pole ρ and the actual closed-loop pole is weak and is effective only when $n_y = 1$.*

Proof. The PFC control and system is summarised as follows when $\rho \neq a$:

$$u_k = \frac{(R(1-\rho^{n_y}) + \rho^{n_y} y_k - a^{n_y} y_k)(1-a)}{b(1-a^{n_y})}; \quad (1 - az^{-1})y_k = bz^{-1}u_k \tag{8}$$

Eliminating u_k, one finds that the closed-loop dynamic is given as:

$$(1-az^{-1})y_k = bz^{-1}u_k \quad \Rightarrow \quad \underbrace{(1 - az^{-1} + \frac{(a^{n_y} - \rho^{n_y})(1-a)b}{b(1-a^{n_y})})}_{Closed-loop\ pole\ polynomial} y_k = \gamma R \tag{9}$$

where γ is a scalar.

The closed-loop pole is therefore defined as:

$$pole = a - \frac{(a^{n_y} - \rho^{n_y})(1-a)}{(1-a^{n_y})} \tag{10}$$

It is obvious that, as n_y increases, the deviation of the pole from a reduces. However, for a coincidence horizon of one, the resulting pole is given as:

$$pole = a - \frac{(a-\rho)(1-a)}{(1-a)} = \rho \tag{11}$$

Hence, one can achieve the desired pole exactly if $n_y = 1$. □

For completeness, Figure 3 shows how the closed-loop pole position varies with the coincidence horizon for various choices of ρ, assuming that $a = 0.8$; the figure makes it clear that the closed-loop poles tend to a for large n_y, but are close to ρ iff n_y is small.

Corollary 1. *The coincidence horizon and parameter ρ only have a noticeable impact on the control law if the coincidence horizon is small. Irrespective of the choice of ρ, the closed-loop pole tends to a as the coincidence horizon tends to infinity. This is a natural consequence of Equation (10), because:*

$$\lim_{i\to\infty} \frac{(a^i - \rho^i)(1-a)}{(1-a^i)} = 0 \tag{12}$$

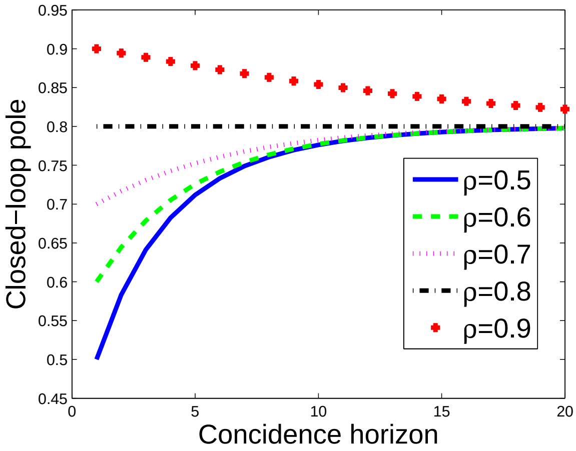

Figure 3. Closed-loop pole dependence upon coincidence horizon and ρ for $a = 0.8$.

2.4. First Order Model with Delay

The modification to deal with delays is subtle, but simple, so just a passing comment is given here; with this minor edit, the results of this section all carry across. Clearly, if a system includes a delay, then a reasonable target to follow is a delayed first order response. For a delay of m samples, the target trajectory becomes:

$$y_{k+n_y} = R - [R - y_{k+m|k}]\rho^{n_y - m}; \quad n_y > m \tag{13}$$

There are two key changes here.

1. The replacement of ρ^{n_y} by ρ^{n_y-m} to take into account the fact that the response cannot begin until after m samples.
2. The replacement of process measurement y_k by a predicted value $y_{k+m|k}$; this second change is essential as the target first order response is expected to begin m samples ahead, so the initial error from the target must be based on that sample.

2.5. PFC for Higher Order Models

In the case of higher order models, one cannot easily write down a simple expression for the n_y step ahead prediction [8]. It can be shown that for a model of the form (usually, we assume $a_0 = 1$):

$$[a_0 + a_1 z^{-1} + \cdots + a_n z^{-n}]y_k = [b_1 z^{-1} + \cdots + b_n z^{-n}]u_k \tag{14}$$

then the n_y step ahead prediction, with the future inputs fixed, takes the form:

$$y_{k+n_y} = h_{n_y} u_k + q_{n_y} \underbrace{[y_k, y_{k-1} \cdots, y_{k-n}]^T}_{\underset{\leftarrow}{y}} + p_{n_y} \underbrace{[u_{k-1}, \cdots, u_{k-n}]^T}_{\underset{\leftarrow}{u}} \tag{15}$$

for suitable $h_{n_y}, p_{n_y}, q_{n_y}$. Consequently, the PFC control law reduces to:

$$R - [R - y_k]\rho^{n_y} = h_{n_y} u_k + q_{n_y} \underset{\leftarrow}{y} + p_n \underset{\leftarrow}{u} \quad \Rightarrow \quad u_k = \frac{R - [R - y_k]\rho^{n_y} - q_{n_y} \underset{\leftarrow}{y} - p_{n_y} \underset{\leftarrow}{u}}{h_{n_y}} \tag{16}$$

It is immediately clear that in this case, the algebra gives very little insight into the expected control law or closed-loop behaviour and, indeed, the sensitivity to the choice of n_y.

Theorem 3. *As the coincidence horizon approaches infinity, the PFC control law will reduce to giving open-loop characteristics for a stable open-loop process, irrespective of the choice of* ρ*.*

Proof. For a stable open-loop process, it is easy to show that the prediction parameters of Equation (15) have the following asymptotic properties:

$$\lim_{n_y \to \infty} \quad \{h_{n_y} \to \frac{\sum_{i=1}^{n} b_i}{\sum_{i=0}^{n} a_i}; \quad p_{n_y} \to 0; \quad q_{n_y} \to 0\} \tag{17}$$

Consequently, the control law becomes:

$$\lim_{n_y \to \infty} \frac{R - [R - y_k]\rho^{n_y} - q_{n_y} \underleftarrow{y} - p_{n_y} \underleftarrow{u}}{h_{n_y}} = \frac{\sum_{i=0}^{n} a_i}{\sum_{i=1}^{n} b_i} R = \frac{R}{K} \tag{18}$$

In other words, the selection of input is the steady-state value required to give no offset and is independent of the current state. □

Corollary 2. *The PFC code for large coincidence horizons can be reduced to a few trivial lines. This is self-evident from Equation (18).*

More general guidance on PFC tuning for higher order models will be discussed in Section 4.

2.6. Model Uncertainty, Offset Free Tracking and Disturbances

It is known that PFC provides robust offset free tracking in the presence of some model uncertainty and disturbances. The detailed algebra is not included in this paper, as it is not a primary focus. Moreover, these details have no impact on the discussions and analysis in this paper, so they are omitted to enable easier reading of the core concepts, although the reader will note that they are included within several of the simulation examples. Therefore, a brief comment only is given here for completeness.

Within PFC, it is common practice to simulate a model (say G_m) in parallel with the process G_p; the model is used for prediction. The difference $d = y_p - y_m$ between the model output y_m and process output y_p is taken as an estimate of the system disturbance, although, in fact, it can also correct the offset for parameter uncertainty. The PFC control law that ensures offset free tracking reduces to:

$$R - [R - y_p(k)]\rho^{n_y} = y_p(k + n_y|k) = y_m(k + n_y|k) + d(k) \tag{19}$$

Remark 3. *Where a system has real roots only, several PFC practitioners (e.g., [4,9,10]) favour using partial fractions to write the model* G_m *as a sum of first order models. This has the advantage that the prediction can be written down explicitly in forms like Equation (5) and, thus, with much simpler coding. However, this issue is not central to the current paper.*

2.7. Constraint Handling

Constraint handling is a core attribute of PFC; however, the techniques for doing this do not form part of the core or novel contribution of this paper, and thus, the reader is referred to the references for a more detailed discussion [10]. It should be noted of course that that the proposals given in this paper fit within a conventional PFC approach, and thus, standard PFC constraint handling procedures can be deployed.

2.8. Summary of PFC Tuning

It is clear that for first order systems, the PFC approach will work well as a means of designing a control law that will give dynamics close to the open-loop behaviour and offset free tracking. Moreover, one can use PFC as an effective way of changing the dynamics, as long as the coincidence horizon is small, preferably one. Section 3 will give some numerical examples to reinforce this.

For systems with more complex dynamics, it is less transparent whether PFC will work well or not. Given the complexity of Equation (16), existing analysis does not give any useful generic results beyond Theorem 3. Consequently, there is a need for some better insight into when PFC is appropriate and how it might be tuned. Section 4 will present some proposals along with numerical illustrations.

3. Numerical Illustrations for First Order Systems

This section will look at the tuning of PFC through a number of first order examples and demonstrate the impact of changing the parameters $\rho,\ n_y$.

3.1. Concepts of Well-Posed Decision Making

Within predictive control, the concept of well-posed decision making is based on the consistency in subsequent samples of any decisions made. In other words, if one determines a strategy now that is supposedly good for the long term, then one would expect that at the next sample, the updated strategy would be very similar. Substantial changes in strategy from one sample to the next are an indicator of inconsistency, which could lead to chaos and most certainly are not a solid foundation for a control law.

In this section, the figures presented will give two line plots:

1. The implied PFC prediction at a given sample (denoted up, yp for the input and output, respectively).
2. The actual closed-loop behaviour that results (denoted us, ys for the input and output, respectively).
3. The signal r in Figures 4, 5, 8, 9 and 12–14 is used to denote the target r^{tr} that arises from Equation (1), rather than the steady-state target value R.

The indicator we are looking at for well-posed decision making is that prediction and behaviour are similar, and most certainly, the change from one sample to the next should be small.

3.2. PFC Responses with Target Behaviour Equal to Open-Loop Behaviour

Take a first order model $(1 - az^{-1})y_k = bz^{-1}u_k$; without loss of generality, we will use $a = 0.8$, $b = 0.25$ for the simulation, as the observations here are based around selecting $\rho = a$. Plot the closed-loop responses for various choices of coincidence horizon and overlay the predictions made at the first sample. The results are given in Figure 4. What is immediately clear is that the predictions and closed-loop behaviour overlap ($ys = yp$) precisely (as expected from Theorem 1) in the nominal case, so in this case, PFC gives well-posed decision making.

For completeness, a Figure 5 shows responses for an equivalent system with a delay; the same observations follow as expected.

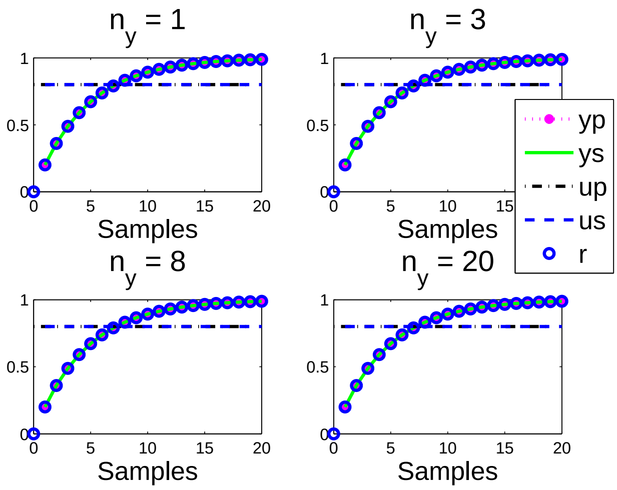

Figure 4. Closed-loop PFC input and output us, ys responses overlaid with predictions up, yp made at the first sample for $a = \rho$; here, $ys = yp$, $us = up$.

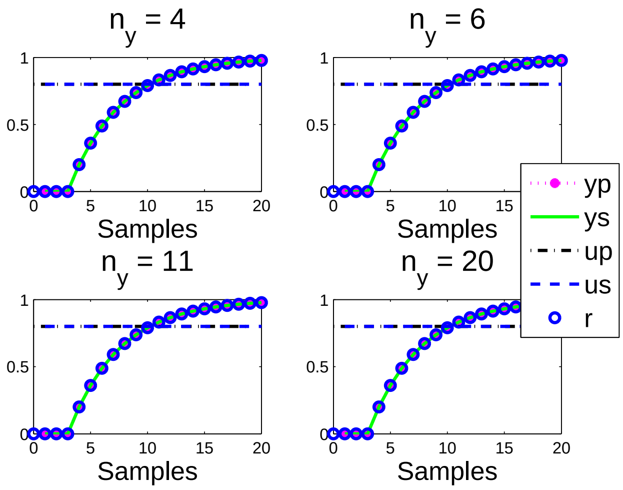

Figure 5. Closed-loop PFC input and output us, ys responses overlaid with predictions up, yp made at the first sample for $a = \rho$; here, $ys = yp$, $us = up$ for system with a three-sample delay.

3.3. PFC Responses with Target Behaviour Different from Open-Loop Behaviour

Take the same model as in the previous section, but use time constants a little faster $\rho < a$ (Figure 6) and a little slower $\rho > a$ (Figure 7) than the open-loop dynamics. Again, investigate various choices of the coincidence horizon. The actual closed-loop dynamic behaviour is strongly linked to the choice of n_y, as discussed in Section 2.3.2, but here, we are also interested in the consistency of the decision making, so the figures overlay the implied predictions utilised by PFC and, of course, the target dynamic.

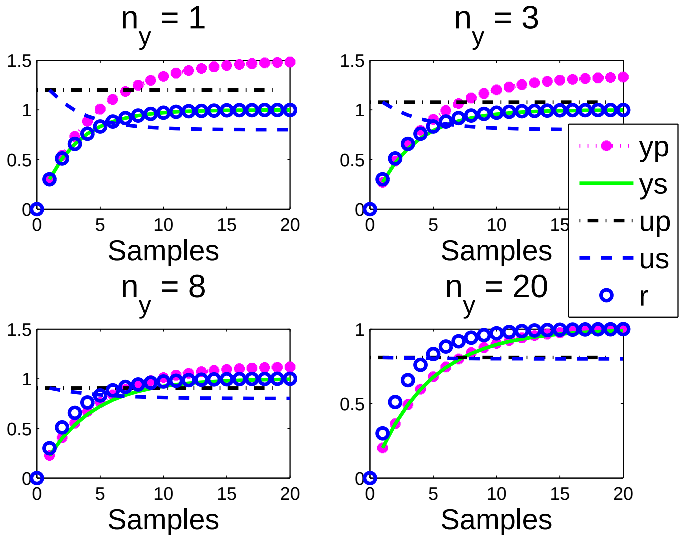

Figure 6. Closed-loop PFC input and output us, ys responses overlaid with predictions up, yp made at the first sample for $a > \rho$.

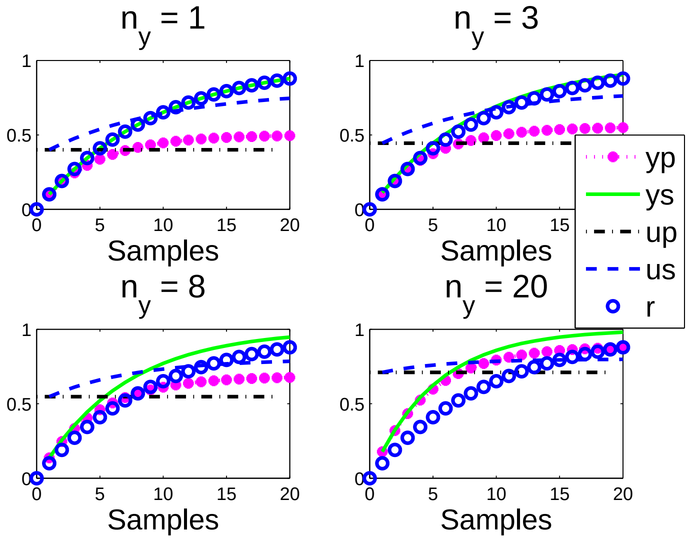

Figure 7. Closed-loop PFC input and output us, ys responses overlaid with predictions up, yp made at the first sample for $a < \rho$.

1. When $\rho \neq a$, for small n_y, there is clearly some inconsistency from one sample to the next, as the input prediction up at the first sample is different from the closed-loop evolution us that results.

For example, when $\rho = 0.7, a = 0.8, n_y = 1$, the input change from Sample 1 to Sample 2 is from 1.2 to 1.08.

2. Despite the inconsistency between implied predictions yp and actual behaviour ys, with $n_y = 1$, the target behaviour r has been achieved.
3. As n_y increases, the choice of ρ becomes less and less significant (the output yp does not track the target r), but there is much better consistency between predictions yp and behaviour ys.

Remark 4. *There is an interesting inconsistency noted here. If the coincidence horizon is small, the design parameter ρ is effective in changing the closed-loop dynamics, but the implied long-term predictions are a poor indicator of final behaviour. Conversely, if the coincidence horizon is large, the design parameter ρ has no affect, but the predictions are consistent. Of course, readers may note that PFC never claimed to take any account of long-term predictions within the decision making process, and therefore, it was not implicit that one would keep the input constant at future samples!*

4. Tuning PFC for Higher Order Systems

The reader will not be surprised with the observation that PFC may be less effective when the expectation of a closed-loop behaviour close to a first order response is invalid or not realistic, as with higher order systems, and therefore, more care is needed. A weakness of PFC is the lack of the explicit inclusion of long-term effects. While for systems with simple first order dynamics this matters little, the repercussions for systems with more challenging dynamics can be more serious, especially if one introduces constraint handling (some examples are in Section 5). Although a well-trained PFC user can demonstrate that PFC works well if tuned effectively, it is important to understand the difficulties that can arise if one makes poor tuning decisions.

In cases where the open-loop response does not include significant oscillation or instability and moves smoothly to the steady state, it is possible to demonstrate that matching the step response to a first order response at some point in the future still makes good intuitive sense and is likely to lead to a well-posed decision making or control law; of course, this is notwithstanding the fact that a rigorous proof or analysis is future work and may even not be possible in many cases. This section will give some demonstrations of how one can be systematic at designing a PFC control law that works reasonably well, that is choosing a coincidence horizon that works reasonably well and achieves close to the target time constants.

4.1. Coincidence Horizon Issues with Higher Order Models

It is normal for higher order systems, even over-damped ones, to have a lag in their step response, and consequently, forcing the system output to precisely match a typical first order response could be counter intuitive. See Figure 8 for an example of this; the figure overlays the step responses of:

$$G = \frac{0.2z^{-1}}{1 - 0.8z^{-1}}; \quad G_2 = \left(\frac{0.2z^{-1}}{1 - 0.8z^{-1}}\right)^2; \quad G_3 = \frac{0.1z^{-1}}{1 - 0.9z^{-1}} \tag{20}$$

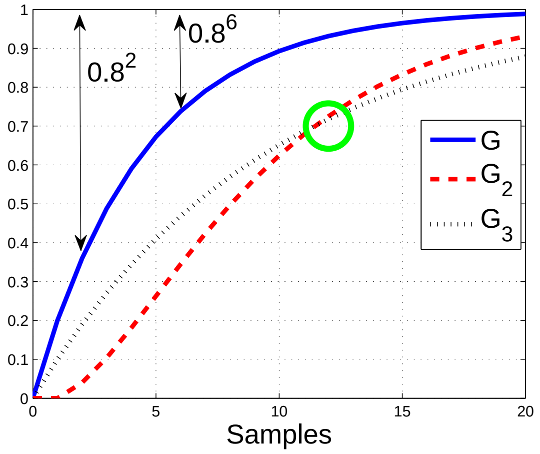

Figure 8. Target first order step response with poles of 0.8 and 0.9 overlaid with a step response of a second order system.

From this observation, it is rather obvious that using a coincidence horizon of one for higher order systems makes no sense at all, and indeed, one may need to target a much slower implied first order pole, such as G_3, along with a higher coincidence horizon ($n_y = 12$ is marked in this figure) to get sensible answers! Indeed, it can be shown that using $n_y = 1$ can imply a controller using "plant inversion", which is not robust.

Readers will note that one can only force coincidence with faster poles (G_2 and G) by having over actuation during transients, so the implied steady state within the prediction is well above the target. Readers are also reminded, however, of Theorem 3 and that using too large n_y PFC essentially reduces to choosing the steady-state input, so the tuning parameter ρ would have a smaller affect.

The proposal here is linked to the underlying common sense or intuitive nature of the PFC approach, that is the argument that the part of the prediction which is ignored (further into the future than the coincidence point) is implicitly moving closer to the desired steady-state, and hence, iterative decision making will be well posed if it gradually moves the coincidence point nearer and nearer to the target (in absolute terms). This will happen automatically if n_y is based on a point where the step response is increasing, and changes relative to the steady state are close to monotonic thereafter.

Conjecture 1. *For systems with close to monotonically convergent behaviour after immediate transients, a good choice of coincidence horizon is one where the open-loop step response has risen to around 40% to 80% of the steady state and the gradient is significant (formal proofs and details constitute future work).*

Remark 5. *A manual for training engineers [4,5] on PFC suggests that a suitable value for n_y might also be determined by identifying the so-called inflexion point on the step response curve, that is, where the gradient is maximum. However, care must be exercised, as the maximum gradient may not be enough on its own, especially with non-minimum phase systems, as noted briefly with system G_4.*

4.2. PFC Tuning for Systems with Relatively Simple Dynamics

Consider a simple second order system:

$$G_2 = \frac{0.04z^{-1}}{(1-0.8z^{-1})^2} \tag{21}$$

and overlay the trajectories of several typical targets with the actual open-loop step response (see Figure 9) and possible choices of coincidence points.

1. Following Conjecture 1 and due to the lag in the process (the step response takes a few samples to start moving significantly), it is not reasonable to use $n_y < 5$. Similarly, $n_y > 15$ will put too much emphasis on the steady state.
2. It is clear that for $\rho = 0.9$, an intercept exists between the step response and target, and hence, $n_y = 8$ would be a logical choice of coincidence point.
3. For smaller and larger choices of ρ, there is no intercept in the range $5 \leq n_y \leq 15$, and hence, to force coincidence (essentially to move the step response up or down), PFC will either have to under- or over-actuate using these choices of ρ. This is what we expect, as a faster ρ requires over-actuation and a slower ρ requires under-actuation.
4. Limits on the actuation permitted will be a guide as to what range of ρ is permissible to request and, indeed, the appropriate n_y. For example, $\rho = 0.7$ requires an input over actuation of around 2.5 to get coincidence at $n_y = 5$ or around 1.8 if $n_y = 8$.

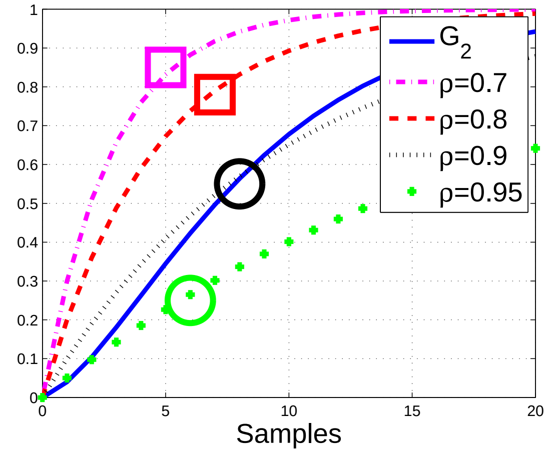

Figure 9. Open-loop step response overlaid with several possible targets and potential coincidence points.

The actual closed-loop responses with PFC are given in Figure 10, and these reinforce the comments above: faster ρ (that is smaller values of ρ) requires over-actuation in transients, a choice of $\rho = 0.8$ and exact coincidence, giving an almost constant input and a slower ρ (that is, $\rho > 0.8$), giving under-actuation. In all cases, it is clear that a sensible choice of n_y has resulted in the choice of ρ being an effective tuning parameter, that is the closed-loop outputs converge roughly at the same speed as the target.

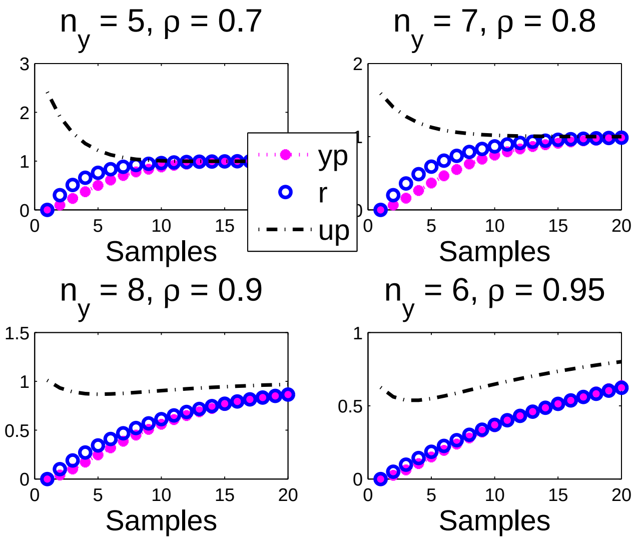

Figure 10. Closed-loop step responses for a second order system G_2 with various ρ, n_y overlaid with the underlying target trajectory.

4.3. PFC Tuning for a Non-Minimum Phase System

Next, take a third order system with some non-minimum phase characteristics, but otherwise smooth behaviour.

$$G_4 = \frac{-0.0569z^{-1} + 0.0514z^{-2} + 0.0502z^{-3}}{1 - 1.9842z^{-1} + 1.3301z^{-2} - 0.3012z^{-3}} \tag{22}$$

Overlay the responses of several typical targets with the actual open-loop step response (see Figure 11). Due to non-minimum phase characteristic, it is clear that using Conjecture 1, the coincidence horizon should be in the range $7 \leq n_y \leq 12$. Potential choices of coincidence points are marked on the figure. Readers may also note that the inflexion point is around $n_y = 5$, and clearly, as the magnitude of the step response is still very small here, such a design choice for n_y is inappropriate and can be shown to give very poor closed-loop performance!

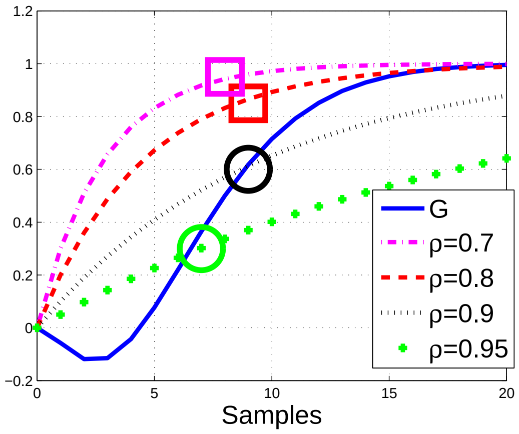

Figure 11. Open-loop step response of G_4 overlaid with several possible targets and potential coincidence points for a non-minimum phase system.

The actual closed-loop responses with PFC are given in Figure 12, and these reinforce the expectations. Notwithstanding the fact that one cannot follow a first order target in immediate transients due to the non-minimum phase characteristic, nevertheless, as long as n_y is well chosen, the choice of ρ has been effective in affecting the closed-loop settling time and input activity, and thus, ρ is an effective tuning parameter.

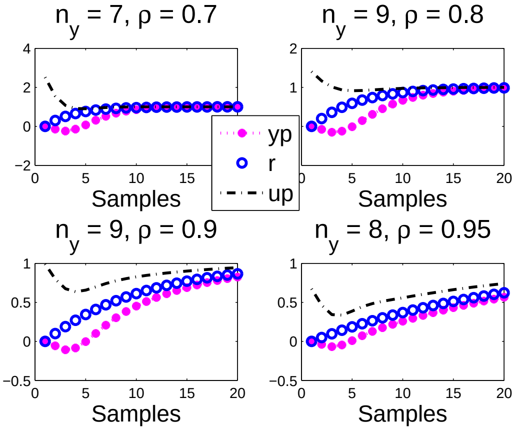

Figure 12. Closed-loop step responses for a third order system G_4 with various ρ, n_y overlaid with the underlying target trajectory.

4.4. PFC Tuning for an under-Damped System

Next, take an under-damped system.

$$G_5 = \frac{-0.0565z^{-1} + 0.0495z^{-2}}{1 - 1.5643z^{-1} + 0.6703z^{-2}} \tag{23}$$

Overlay the responses of several typical targets with the actual open-loop step response (see Figure 13). In order to follow Conjecture 1, it is better to select a coincidence horizon in the range $3 \leq n_y \leq 6$. Potential choices of the coincidence point are marked on the figure.

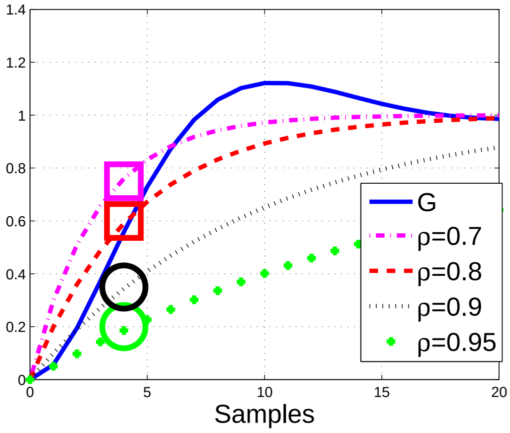

Figure 13. Open-loop step response of G_5 overlaid with several possible targets and potential coincidence points for an under-damped system.

The actual closed-loop responses with PFC are given in Figure 14, and once again, it is seen that the choice of ρ has been effective in affecting the closed-loop settling time and input activity and, thus, is an effective tuning parameter.

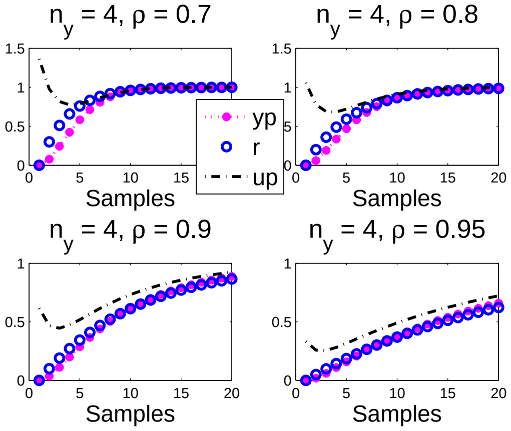

Figure 14. Closed-loop step responses for an under-damped system G_5 with various ρ, n_y overlaid with the underlying target trajectory.

4.5. Summary

This section has shown that even for processes with underlying dynamics that are not close to a first order system, nevertheless, PFC can be an effective strategy and, moreover, the tuning parameter ρ, that is the target closed-loop time constant, can be used effectively. This, of course, is invaluable to technical users who wish to specify response times rather the controller gains. In practice, the desired settling time (CLTR) is used, which can be represented as an equivalent ρ, as shown earlier. It is noted that one needs to select the coincidence horizon carefully, and the proposal here involves an overlaying of typical targets with the open-loop step response as a simple mechanism (Conjecture 1) of identifying a suitable n_y. One can, of course, do an iterative design, underpinned by simulations with various n_y, to determine the final choice.

5. PFC for Processes with Difficult Dynamics

It was hinted in earlier sections that the main concepts in PFC may not carry over well to processes with oscillatory, unstable or other non-simple dynamics. In simple terms, the assumption of a constant input is not sufficient to create an output that will have good characteristics [11]. In order then to make PFC more effective and easier to tune, it may be necessary to undertake some pre-conditioning to form input predictions and possibly output targets, which are more realistic, but notably, without following the normal predictive control route of performance indices and quadratic programming optimisations, which are non-trivial to implement. This section illustrates some of those scenarios, and notably, these are systems that fail the core requirements to apply Conjecture 1.

5.1. Second Order Model with Oscillatory Dynamics

It is clear that a system with oscillatory dynamics cannot easily match a first order target. Moreover, it is intuitively obvious that matching the predicted response at a single coincidence point in the future could be a very poor indicator of actual closed-loop behaviour. The tuning difficulties are well illustrated with the following system:

$$G_6 = \frac{0.1z^{-1} + 0.4z^{-2}}{1 - z^{-1} + 0.8z^{-2}} \tag{24}$$

The results of using PFC are given in Figure 15 and in essence demonstrate:

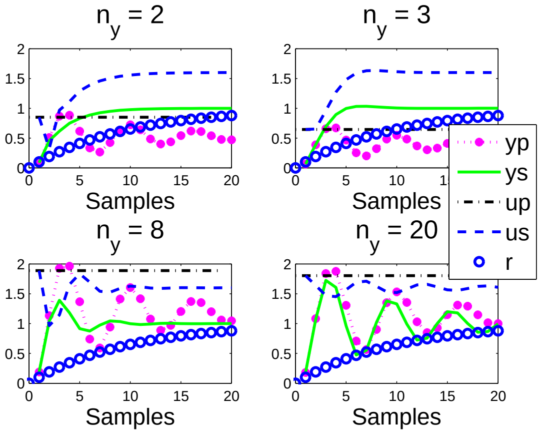

Figure 15. Closed-loop PFC input and output us, ys responses overlaid with predictions up, yp made at the first sample for an oscillatory system G_6.

1. PFC gives good results with $\rho = 0.7$, $n_y = 2, 3$ (fails totally for $n_y = 1$), but it is difficult to argue that this is more than just a fluke, given that the shapes of the closed-loop input trajectory are not close to a constant.
2. For large horizons, the open-loop dynamic dominates and the behaviour is unacceptable.

5.2. Model with Open-Loop Unstable Dynamics

There has been some brief discussion of using PFC with open-loop unstable processes in the past [12]; the proposal there was to prestabilise the predictions by cancelling unstable modes [11] before applying a PFC algorithm. However, before we return to discussions on how the base PFC algorithm could be modified (future work), it is worth demonstrating the extent to which it is effective in its base form. The tuning difficulties are well illustrated with the following system and $\rho = 0.8$:

$$G_7 = \frac{0.1z^{-1} + 0.4z^{-2}}{1 - 1.2z^{-1}} \tag{25}$$

The results of using PFC are given in Figure 16 and, in essence, demonstrate the following:

1. PFC fails for both $n_y = 1$ and large n_y.

2. In this case, reasonable behaviour results for $n_y = 3$, but it would be difficult to argue that this is a systematic design; and certainly, the parameter ρ has not been effective.

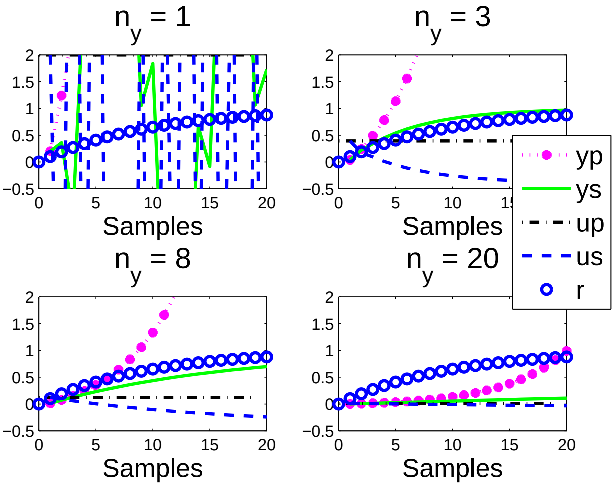

Figure 16. Closed-loop PFC input and output us, ys responses overlaid with predictions up, yp made at the first sample for unstable process G_7.

5.3. Summary

For many systems, the tuning of PFC is not obvious, and the designer is left with, in effect, a global search across ρ, n_y, but without a clear expectation of whether this will give a good result or not. Intuitively, the core failing is the desire to follow a first order trajectory, and using a constant future input as this is clearly unrealistic for systems with complex dynamics. Therefore, to extend the benefits of PFC, one needs to extend the target behaviours and/or input parametrisations. It should be emphasised, however, that PFC practitioners have work-arounds, and modifications for these scenarios and future work will investigate those proposals in depth, alongside the consideration of alternative modifications.

- A reasonable conjecture is that one should begin from a target trajectory that is realistic, as then, ensuring the predictions are close to this trajectory would give a high expectation of good behaviour.
- For systems with oscillatory open-loop dynamics, it is reasonable to conjecture that the future control input should be shaped so that the impact of this oscillation is minimised.
- For unstable open-loop processes, it is well known that one should parametrise the future decisions in such a way that the predictions are all guaranteed convergent [7], preferably to the correct steady state.

In summary, therefore, rather than using the next value of the input u_k as the control decision and assuming that all future inputs are constant, one needs alternative parametrisations, which ensure that the process has underlying good behaviour and, within that parametrisation, include some flexibility that can be used to influence the speed of response.

6. Conclusions

This paper has provided an elementary analysis of PFC and the expected behaviour and tuning. It is clear that generic stability and performance results may not be possible, in general, and, here, have been determined rigorously only for the simple first order case.

6.1. Key Points

For the first order case, it was shown that the choice of pole is meaningful only for a coincidence horizon of one. As the coincidence horizon is increased, this parameter has a decreasing effect on the control law and resulting closed-loop behaviour, and indeed, for high n_y, the PFC control law reduces to estimating the process steady state input; thus, ρ has no impact at all.

For higher order systems, the tuning rules and efficacy of the parameters ρ, n_y are not consistent and vary from one example to another depending on the underlying process dynamics. Consequently, some suggestions were made as to how one could determine an appropriate range of n_y, ρ before any fine tuning. The methodology was demonstrated on several examples and found to be effective and, moreover, is similar, but not identical, to the empirical suggestions in [4].

Finally, a rather simple, but useful observation, is that PFC with a large coincidence horizon reduces to an integral control law, which gives a closed-loop with open-loop dynamics (or so-called mean-level control, as noted elsewhere in the predictive control literature). Where such behaviour is acceptable, one can vastly simplify the current PFC code to just a few lines, and moreover, in cases of only input constraints, even the constraint handling becomes trivial.

6.2. Future Work

Aside from the need to consider whether one can give more rigorous tuning rules with guarantees of behaviour or performance with higher order systems, there are also a number of issues not covered here that would be interesting and should be pursued next. For example, one advantage of PFC is the ease with which one can target ramps and parabolas with no offset or lag, so it would be interesting to develop some analysis of how that can be done systematically. When does it work well and what sorts of scenarios produce challenges? Are there simple improvements that can be suggested?

In a similar vein, the constraint handling approaches of PFC are relatively simple, and thus, it is easy to find examples where the strategies would fail. However, the more important point is that for many realistic scenarios, it works quite well, and with a very simple approach to code and maintain. Hence, once again, it would be useful to provide a more formal analysis of the efficacy of PFC approaches to constraint handling.

Finally, it was noted that there do not exist simple proposals in the literature for how the simple concepts of PFC can be adapted to deal with some challenging dynamics; this would be worthwhile, as the benefits of easy coding and tuning would make such proposals useful.

Acknowledgments

Thanks to Jacques Richalet, the founder of PFC, for many discussions and supporting code.

Author Contributions

This paper is a collaborative work between both authors. R.H. focussed on the accurate communication of the concepts and J.A.R. provided initial proposals and coding.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Rossiter, J.A.T. *Model-Based Predictive Control: A Practical Approach*; CRC Press: London, UK, 2003.
2. Maciejowski, J. *Predictive Control with Constraints*; Prentice Hall: Harlow, UK, 2001.
3. Richalet, J.; Rault, A.; Testud, J.L.; Papon, J. Model predictive heuristic control: Applications to industrial processes. *Automatica* **1978**, *14*, 413–428.
4. Richalet, J.; O'Donovan, D. *Predictive Functional Control: Principles and Industrial Applications*; Springer-Verlag: London, UK, 2009.
5. Richalet, J.; Lavielle, G.; Mallet, J. *Commande Predictive*; Eyrolles: Paris, France, 2004.
6. Scokaert, P.O.M.; Rawlings, J.B. Constrained linear quadratic regulation. *IEEE Trans. Autom. Control* **1998**, *43*, 1163–1168.
7. Rossiter, J.A.; Rice, M.J.; Kouvaritakis, B. A numerically robust state-space approach to stable predictive control strategies. *Automatica* **1998**, *34*, 65–73.
8. Rossiter, J.A. Notes on multi-step ahead prediction based on the principle of concatenation. *Proc. Inst. Mech. Eng. I J. Syst. Control Eng.* **1993**, *207*, 261–263.
9. Khadir, M.T.; Ringwood, J.V. Extension of first order predictive functional controllers to handle higher order internal models. *Int. J. Appl. Math. Comput. Sci.* **2008**, *18*, 229–239.
10. Haber, R.; Bars, R.; Schmitz, U. *Predictive Control in Process Engineering: From the Basics to the Applications, Chapter 11: Predictive Functional Control*; Wiley-VCH: Weinheim, Germany, 2011.
11. Rawlings, J.B.; Muske, K.R. The stability of constrained receding horizon control. *Trans. IEEE Autom. Control* **1993**, *38*, 1512–1516.
12. Rossiter, J.A.; Richalet, J. Handling constraints with predictive functional control of unstable processes. *Am. Control Conf.* **2002**, *6*, 4746–4751.

© 2015 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/4.0/).