Article:

Rossiter, J.A. (2015) Input shaping for PFC: how and why? Journal of Control and Decision, 3 (2). ISSN: 2330-7706

https://doi.org/10.1080/23307706.2015.1083408

To appear in the *Journal of Control and Decision*
Vol. 00, No. 00, Month 20XX, 1–14

Input shaping for PFC: how and why?

J.A. Rossiter[a*]
[a] Department ACSE, University of Sheffield, S1 3JD, UK

(*Received 00 Month 20XX; accepted 00 Month 20XX*)

Predictive functional control (PFC) is a highly successful strategy within industry, but for cases with challenging dynamics the most effective tuning approaches are still an active research area. This paper shows how one can deploy some insights from the more traditional model predictive control literature in order to enable systematic tuning and in particular, to ensure that the key PFC tuning parameter, that is the desired closed-loop time constant, is effective. In addition to enabling easier and more effective tuning, the proposed approach has the advantage of being simple to code and thus retaining the simplicity of implementation and tuning that is a key selling point of PFC. This paper focuses on design for open-loop unstable processes and also processes with significant under-damping in their open-loop behaviour.

Keywords: Predictive control, unstable, non-minimum phase, constraint handling

1. Introduction

PID compensators are effective on a large number of industrial control loops, but are less effective on a minority of more challenging processes such as those with significant interaction, constraints and non-minimum phase, oscillatory or unstable dynamics. While numerous alternatives have been proposed in the literature, the most popular is model predictive control (MPC) e.g. (Maciejowski, 2001; Richalet and Donovan, 2009; Rossiter, 2003). Within typical industrial applications of MPC, there is a clear divide between: (i) conventional large scale, and expensive, implementations which tend to use dynamic matrix control (Cutler and Ramaker , 1980) or dual-mode approaches (Sockaert and Rawlings, 1998) and (ii) smaller scale, mostly single input single output, less expensive implementations based on the predictive functional control (PFC) algorithm (Richalet et al., 1978, 2004). This paper will focus on the latter approach and thus a key aim is to propose simple and cheap predictive control algorithms which can be coded with relative simplicity.

A nominal PFC approach is based on common human principles for control (Haber et al., 2011): (i) anticipate the effect of a fixed control move on the output at some point in the future and choose the control move such that one moves closer to the target than now; (ii) repeat this process at each sample and (iii) an underlying embedding of the desired convergence time/time constant. For systems with monotonic dynamics, one can argue intuitively (a theoretical proof is less obvious except for 1st order models) that such a strategy will give rise to control moves that gradually move the output to the target (Rossiter et al., 2015). Moreover, as PFC is

*Corresponding author. Email: j.a.rossiter@shef.ac.uk

based on a prediction, one can easily incorporate system delays and constraints in a systematic fashion and this explains the large numbers of industrial applications, e.g. Changenet et al. (2008); Fallasohi et al. (2010); Richalet and Donovan (2009).

However, a key issue for academia is that although intuitively reasonable, it is less obvious whether PFC can be mathematically assured to give good behaviour (Rossiter et al., 2015) or indeed tuned in a systematic fashion. Some examples where PFC is much harder to tune effectively include systems with unstable open-loop dynamics, non-minimum phase behaviour and significant open-loop under-damping. PFC practitioners have a number of alternative structures (Richalet et al., 2004) for different scenarios although there does not exist in the literature a clear and systematic statement or explanation of these alternatives. For example, in many cases practitioners deploy a form of cascade of PFC designs, where an inner design is selected to give an improved dynamic for the outer PFC to control. In general, a key selling point is that a modified PFC that works, and retains some of the simple PFC insights, is often better than the obvious alternative of PID, especially as constraints can still be handled systematically.

A key motivation for this paper is to propose modifications to PFC which will improve the tuning, feasibility and stability properties, but while retaining simplicity of implementation. Specifically, the paper explores how some insights from more conventional long range MPC can be deployed to advantage within a PFC framework while being both systematic and also simple. The 2nd section will give a brief background on a nominal PFC algorithm. The next two sections will introduce the concept of input prediction shaping and show how this improves PFC performance and tuning substantially in some cases. The paper finishes with examples, conclusions and future work.

2. Background on PFC

2.1. *Nominal PFC law*

The basic premise of a PFC law is to force the n-step ahead system prediction $y_{k+n|k}$ assuming a constant future input $u_{k+i|k} = u_k, i = 0, ..., n$, to be closer to the target r than the current output y_k. The ratio of the error reduction per sample λ is in effect the target closed-loop pole (equivalent to settling time). Hence, a nominal PFC law is derived from enforcing, at each sample, the equality:

$$y_{k+n|k} = (1-\lambda^n)r + \lambda^n y_k; \quad u_{k+i|k} = u_k, \ \forall i \geq 0 \tag{1}$$

The key design choices are the coincidence horizon n and the target closed-loop pole λ. The popularity of PFC links to the fact that the desired closed-loop settling time (often denoted TRBF: $\lambda = e^{-\frac{T}{TRBF}}$, T the sampling time) makes intuitive sense and thus, if the choice made by the designer is effective in achieving the desired closed-loop dynamics, then the design method is easy to deploy.

Remark 1 There are some subtleties linked to offset free tracking and disturbance estimation which are omitted from (1) to improve the clarity of presentation of the core concepts. The simulation results do include the relevant algebra.

Remark 2 For systems with close to monotonic step responses such as 1st order or over-damped 2nd order dynamics, it is well known that PFC is effective and the design/implementation is relatively straightforward, as long as λ is chosen within a sensible (achievable) range. Consequently this paper will not revisit such cases.

2.2. *Nominal PFC for systems with challenging dynamics*

The nominal PFC law of (1) can be difficult to tune on some systems (Rossiter et al., 2015).

(1) For systems with a significant non-minimum phase characteristic, the coincidence horizon must be large and this undermines the efficacy of λ as a tuning parameter. In simple terms, PFC can work well but usually only for λ close to or slower than the open-loop dynamics.
(2) For systems with a significant lag or slow initial response, a similar observation follows.
(3) PFC can work for some unstable open-loop systems, but often it fails and a more nuanced implementation, such as a cascade loop, is required, thus making the PFC design more complex as two interacting loop designs are now needed.
(4) For systems with strong open-loop oscillation, the assumption of a constant future input is inconsistent with the need to reduce damping so again a more nuanced PFC implementation is required.

Rather than giving detailed consideration of the various cascade loop implementations favoured in the PFC community for these challenging cases, the purpose of this paper is to propose what could be considered a more systematic and easier to implement solution in that the design choices to be made are transparent and the coding simple.

2.3. *Prediction and the nominal PFC law*

A core part of a PFC algorithm is the n-step ahead prediction. Prediction based on linear models is well known in the literature (e.g. Rossiter (1993)) so only the results are given here. For a linear model, the n-step ahead *unbiased* prediction can take the following form where parameters H, P, Q depend upon the model parameters:

$$y_{k+n|k} = H\Delta \underset{\rightarrow k}{u} + P\Delta \underset{\leftarrow k}{u} + Q\underset{\leftarrow}{y} \tag{2}$$

where for a model of order m:

$$\Delta \underset{\rightarrow k}{u} = \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+n-1} \end{bmatrix}; \Delta \underset{\leftarrow k-1}{u} = \begin{bmatrix} \Delta u_{k-1} \\ \Delta u_{k-2} \\ \vdots \\ \Delta u_{k-m} \end{bmatrix}; \underset{\leftarrow k}{y} = \begin{bmatrix} y_k \\ y_{k-1} \\ \vdots \\ y_{k-m} \end{bmatrix};$$

Using prediction equation (2) as the basis for PFC law computations, one should substitute directly into (1). First define H_1 to be the first column of H as predicted control moves Δu_{k+i} are assumed zero for $i > 0$ and hence:

$$\begin{array}{c} H_1\Delta u_k + P\Delta \underset{\leftarrow k}{u} + Q\underset{\leftarrow}{y} = (1-\lambda^n)r + \lambda^n y_k \\ \Downarrow \\ \Delta u_k = \frac{(1-\lambda^n)r + \lambda^n y_k - (P\Delta \underset{\leftarrow k-1}{u} + Q\underset{\leftarrow k}{y})}{H_1} \end{array} \tag{3}$$

Remark 3 Practitioners in PFC tend to use independent model forms with a correction term to compute unbiased predictions. Such decisions have no impact on nominal behaviour and tuning, which is the focus of this paper. However, readers will be interested to know that the choice of prediction structure (Clarke et al., 1987) can have a significant impact on loop sensitivity and thus this issue forms a future research area.

3. Prediction shaping for unstable open-loop processes

3.1. *Review of MPC literature*

It was noted in the early predictive control literature (Rawlings and Muske (1993); Rossiter et al. (1997)) that a nominal predictive control law often worked poorly on open-loop unstable processes. The reason for this was that the predictions were based on open-loop behaviours and thus, in essence, one was trying to combine a set of divergent predictions in order to find a nice, convergent, prediction. Clearly this is ill-posed. A better posed problem is to look for a linear combination of convergent predictions. Hence, the key proposal in the literature was to *shape* or constrain the input predictions so that the associated output predictions were convergent.

In order to find convergent predictions, one needs to identify and separate the unstable modes. Let the system model be given as:

$$y(z) = \frac{b(z)}{a(z)}u(z); \quad a(z) = a^-(z)a^+(z) \tag{4}$$

where $a^+(z)$ contains all the poles outside the unit circle. Conceptually the predicted input sequence $u(z)$ must be selected to cancel $a^+(z)$ in order for the output predictions to be convergent and this must be done at every sample in such a way that it takes into account the changing initial conditions.

Remark 4 Subsequent work in MPC (Rossiter et al. (1998)) has proposed a cascade structure for MPC where an inner loop gives stable behaviour and the outer loop is used to tailor performance. Such an approach is not explored here as it is likely to lead to more challenging constraint handling which is not consistent with the PFC philosophy of requiring very simple and brief code. Moreover, it requires the question of how to select the pre-stabilising controller to be answered. Nevertheless, this is a focus of future work.

3.2. *Motivation for input shaping in PFC*

PFC is based on a common sense human approach that if we have a prediction at the current sample that is moving us closer to the target, then at the next sample, I should be able to improve upon this and give a prediction that goes even closer. It is clear therefore that it is implicit within the PFC methodology that:

- one is using convergent predictions.
- at each sample, ideally one should build upon the action of the previous sample rather than starting afresh.

Typical PFC laws have assumed $u_{k+i|k} = u_k, \forall i \geq 0$ or equivalently $\Delta u_{k+i|k} = 0, \forall i > 0$. Hence the degree of freedom is the current control increment Δu_k. For unstable open-loop processes such a choice of future inputs must give a divergent output prediction because there is not enough flexibility to cancel $a^+(z)$. This divergence indicates that the output prediction is not useful for decision making and that is the key flaw in a simplistic implementation of PFC for unstable open-loop processes!

3.3. *Derivation of input shaping to give stabilising predictions*

A key concept in predictive control is that the class of predictions includes those which converge close to the target so that one can choose a set of input predictions giving rise to good behaviour. For unstable open-loop processes, one mechanism for ensuring this was discussed in Rossiter et al. (1997). The approach is presented next before the paper moves on to show how this can be deployed within a PFC framework.

This section outlines the algebra needed to parameterise the future input predictions which ensure convergent output predictions (Rossiter, 2003). The model equation in terms of input increments is given as:

$$b(z)\Delta u(z) = a^+(z)a^-(z)\Delta(z)y(z) \tag{5}$$

where it is assumed that $b(z) = b_1 z^{-1} + \cdots$, $a(z) = 1 + a_1 z^{-1} + \cdots$ and $\Delta = 1 - z^{-1}$. Using this to form future output predictions one can write:

$$[C_{a^+}][C_{a^-\Delta}]\underset{\rightarrow k+1}{y} + H_A \underset{\leftarrow k}{y} = C_{zb}\Delta \underset{\rightarrow k}{u} + H_{zb}\Delta \underset{\leftarrow k-1}{u} \tag{6}$$

where for $f(z) = f_0 + f_1 z^{-1} + \cdots + f_n z^{-n}$,

$$C_f = \begin{bmatrix} f_o & 0 & 0 & \cdots \\ f_1 & f_0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \\ f_n & f_{n-1} & f_{n-2} & \cdots \end{bmatrix} H_f = \begin{bmatrix} f_1 & f_2 & \cdots & f_n \\ f_2 & f_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots \\ 0 & 0 & \cdots & 0 \end{bmatrix}; \quad \underset{\rightarrow k+1}{y} = \begin{bmatrix} y_{k+1} \\ y_{k+2} \\ \vdots \\ y_{k+n} \end{bmatrix}$$

Next, express the output predictions (6) in terms of past data and future input increments, hence:

$$\underset{\rightarrow k+1}{y} = [C_{a^-\Delta}]^{-1}[C_{a^+}]^{-1}[C_{zb}\Delta \underset{\rightarrow k}{u} + \underbrace{H_{zb}\Delta \underset{\leftarrow k-1}{u} - H_A \underset{\leftarrow k}{y}]}_{p} \tag{7}$$

Writing this in terms of an equivalent z-transform, it is clear that the predictions $\underset{\rightarrow k+1}{y}$ are convergent iff:

$$[1, z^{-1}, \cdots, z^{-n+1}][C_{zb}\Delta \underset{\rightarrow k}{u} + p] = a^+(z)\gamma(z) \tag{8}$$

where γ is a convergent sequence or a polynomial. Requirement (8) has a minimal order solution which can be represented as:

$$\Delta \underset{\rightarrow k}{u} = P_1 p; \quad \gamma = P_2 p; \quad \gamma(z) = [1, \cdots, z^{-n+1}]\gamma \tag{9}$$

for suitable P_1, P_2 where the solution for this involves a small number of simultaneous equations. A general solution for the future control moves which ensure convergent output predictions is therefore given as:

$$\Delta \underset{\rightarrow k}{u}(z) = [1, \cdots, z^{-n+1}]P_1 p + a^+(z)\phi(z) \tag{10}$$

where $\phi(z)$ constitutes degrees of freedom (d.o.f.); for PFC it is normal to use just one d.o.f. and thus $\phi(z) = \phi$, that is a constant. Combining (8-10) one can write:

$$[1, \cdots, z^{-n+1}][C_{zb}\Delta \underset{\rightarrow k}{u} + p] = a^+(z)\gamma(z) + zb(z)a^+(z)\phi \tag{11}$$

where γ depends upon the initial conditions and ϕ constitutes the d.o.f.. The corresponding output predictions are given from substitution of (11) into (7):

$$\begin{aligned} \underset{\rightarrow k+1}{y} &= [C_{a^-\Delta}]^{-1}[C_{a^+}]^{-1}[C_{zb}\Delta \underset{\rightarrow k}{u} + p] \\ &= [C_{a^-\Delta}]^{-1}[C_{a^+}]^{-1}[C_{zb}C_{a^+}\phi + C_{a^+}\gamma] \\ &= [C_{a^-\Delta}]^{-1}[C_{zb}\phi + P_2 p] \end{aligned} \tag{12}$$

Extracting just the nth row and using the definition of p in (7), one can write:

$$y_{k+n} = H_n\phi + P_n\Delta \underset{\leftarrow k-1}{u} + Q_n \underset{\leftarrow k}{y} \tag{13}$$

for suitably defined H_n, P_n, Q_n.

Remark 5 It is logical to choose γ so that the associated part of the prediction matches the planned trajectory from the previous sample and thus, conceptually, ϕ represents only updates or changes to the plan from the previous sample.

Remark 6 Because the input parameterisation of (13) is based on process measurements and unbiased predictions, it is robust to some parameter uncertainty and disturbances.

3.4. *PFC algorithm with shaped predictions*

The previous section showed how one can form output predictions (12) which were guaranteed convergent (nominal case) by shaping the future input prediction to take appropriate consideration of both initial conditions and also ensuring that future d.o.f. imply control moves which cancel any unstable modes. Having determined the n-step ahead prediction of (13) one can now define PFC in a conventional manner and moreover one which includes an effective but easy to code constraint handling facility.

Algorithm 1 PFC for unstable open-loop processes Use the predictions implicit in (13) in the PFC control law of (1)to identify a suitable ϕ. Hence:

$$\begin{gathered} H_n\phi + P_n\Delta \underset{\leftarrow k-1}{u} + Q_n \underset{\leftarrow k}{y} = (1-\lambda^n)r + \lambda^n y_k \\ \Downarrow \\ \phi = \frac{(1-\lambda^n)r + \lambda^n y_k - (P_n\Delta \underset{\leftarrow k-1}{u} + Q_n \underset{\leftarrow k}{y})}{H_n} \end{gathered} \tag{14}$$

Use this ϕ in the predicted input of (10) to determine to current Δu_k.

Remark 7 Constraint handling can be included by modifying ϕ as necessary to ensure any associated predictions in (11,12) satisfy constraints; this technique requires a simple loop and is common practice in industrial PFC implementations.

Theorem 1 *The algorithm of (1) is recursively feasible in the presence of constraints. That is, choosing ϕ so that input predictions satisfy constraints at the current sample guarantees that one can make the same statement at the next sample.*

Proof: By definition the choice of $\phi = 0$ ensures feasibility in the nominal case because the input prediction component $P_1 p$ is the unused part of the input prediction from the previous sample and this is known to satisfy constraints by assumption. One can ensure feasibility at start up by beginning from a steady-state.

Remark 8 Readers may note that although algorithm 1 is effective, as noted in the conclusion section, there are avenues of further study which give optimism that more improvements on a classic PFC are possible, albeit at the price of a small increase in the algebraic requirements.

4. Numerical example of open-loop unstable system

This section will demonstrate the efficacy of algorithm 1 as compared to the nominal PFC law of (3). Notable points are that the proposed algorithm allows:

(1) Systematic tuning in that the user choice λ has a clear impact on closed-loop behaviour, whereas it often does not for conventional PFC.
(2) The impact of coincidence horizon on closed-loop behaviour is much smaller with the proposed algorithm making design easier.
(3) Effective constraint handling without instability despite the system being open-loop unstable - something which the obvious alternative of PID cannot do a simple manner.
(4) Robustness to parameter uncertainty (figure omitted to save space).

The open-loop unstable system considered is:

$$G_1 = \frac{0.2z^{-1} - 0.4z^{-1}}{1 - 1.9z^{-2} + 0.84z^{-2}} \tag{15}$$

4.1. *Efficacy of λ as a tuning parameter*

A key selling point of PFC is that the user can select the desired closed-loop dynamics directly by selecting λ. However, as is demonstrated here, this is often ineffective for a classical PFC approach, especially with unstable open-loop processes. For clarity, the number of samples it should take to reach approximately 90% of target with a specified λ is shown in table 1.

Table 1. Settling time (to 10% error) dependence upon λ.

λ	0.5	0.6	0.7	0.8	0.9
Samples	4	5	6-7	10-11	21-22

The upper figure 1 shows the closed-loop responses with a default PFC law of (1) using open-loop predictions (upper figure) and from this we can conclude.

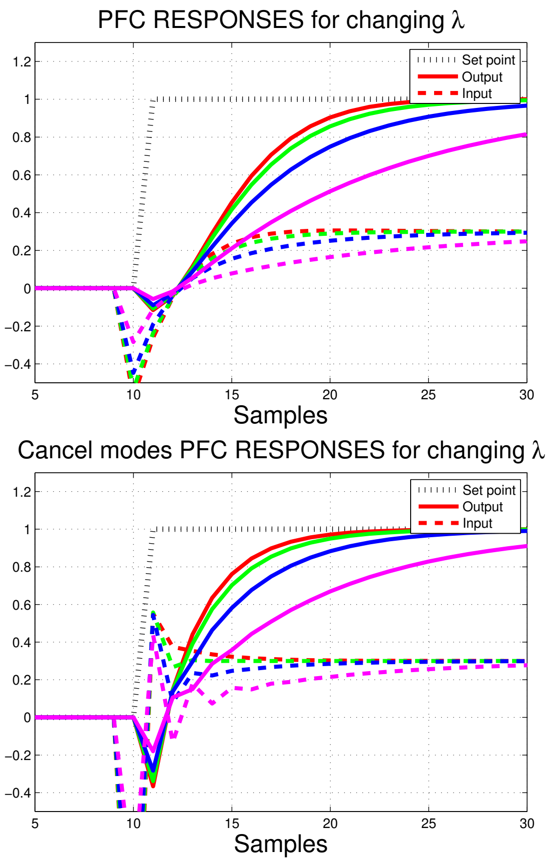

Figure 1. Behaviour of a nominal PFC algorithm and algorithm 1 for $n = 6, \lambda = [0.6, 0.7, 0.8, 0.9]$ (the linking of line plots to λ should be obvious) on G_1.

- Although stable behaviour is possible, the link between the selected λ and the closed-loop behaviour that results is very weak. This can be seen by comparing the response time scales with the 'target' convergence rate in table 1.
- Consequently, λ has not been an effective tuning parameter and this undermines a core part of the PFC design.

Conversely, with algorithm (1) the parameter λ is far more effective as seen in the lower figure 1.

- There is a close correspondence between the closed-loop responses and the choice for λ.
- Consequently, λ has been an effective tuning parameter and this makes the PFC approach sensible.

Readers may be interested to note that effective PFC design tools which guarantee a close correspondence between the tuning parameter λ and the resulting closed-loop behaviour is a currently active research area with, as yet, few results in the literature Rossiter et al. (2015).

4.2. *Sensitivity of design to choice of coincidence horizon*

A significant advantage of using shaped predictions for open-loop unstable process is that the implementation of control law (1) is much less sensitive to the choice

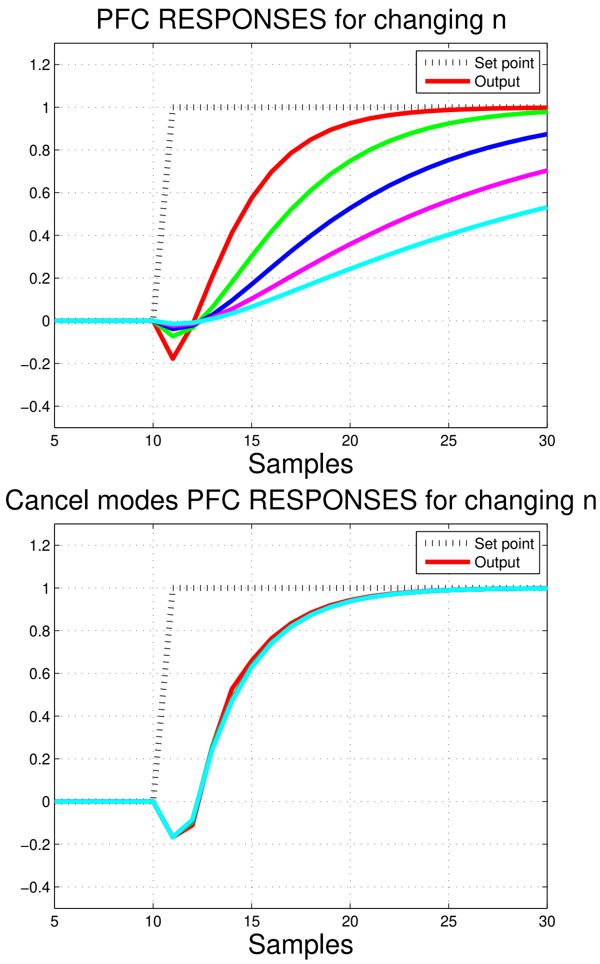

Figure 2. Behaviour of a nominal PFC algorithm and algorithm 1 for $n = [5, 7, 9, 11, 13], \lambda = [0.7]$ on G_1 shows consistent behaviour for different n only for the cancel modes approach of algorithm 1.

of n. Figure 2 shows the closed-loop responses from control law (3) with various choices of n; it is clear that for many of these the performance is poor, has weak correspondence with λ and therefore the approach is unreliable. In effect the user has to search for a suitable choice of n but without the assurance that a good value exists.

Conversely, the PFC algorithm 1 is relatively insensitive to the choice of n in that stable performance is achieved, and indeed, for this case, the closed-loop performance is reliably linked to the target λ.

4.3. *Ability to do constraint handling*

Figure 3 shows that the proposed algorithm can deal with constraints (here an input rate limit) in a straightforward fashion and without risk to stability; this is clearly a significant advantage. Using an equivalent constraint handling approach with the PFC law of (1) gives rise to closed-loop instability because the decisions are based on divergent predictions and thus no recursive feasibility result exists!

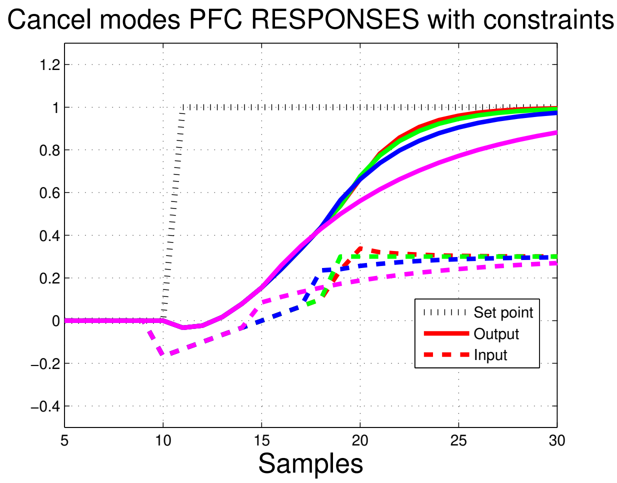

Figure 3. Demonstration of constraint handling with PFC algorithm 1 for $n = 6, \lambda = [0.6, 0.7, 0.8, 0.9]$ $|\Delta u| < 0.2, -0.22 < u < 0.45$ on G_1.

5. Prediction shaping for oscillatory open-loop dynamics

The previous section demonstrated that pre-stabilised predictions can be deployed in PFC very effectively. However, one set of problems that have not been widely explored within conventional MPC are those with stable but challenging dynamics, such as non-minimum phase characteristics or significant oscillation. The lack of detailed attention is likely because the use of an input horizon of 3-5 is normally sufficient to ensure good input predictions and thus there has been less need. However, a typical PFC approach uses just one degree of freedom to enable simple optimisation and coding. The obvious contradiction here is that the future input sequence may require far more involved shaping then a simple step in order to deliver an appropriate output prediction and thus the assumption $\Delta u_{k+i|k} = 0$ is inappropriate and indeed counter intuitive.

A human based predictive strategy takes account of the underlying system dynamics and therefore plans future inputs with whatever shaping is required to give suitable predictions. It is therefore rather obvious that an ideal PFC strategy should do the same!

5.1. *Shaping with oscillatory characteristics*

When a process has oscillatory open-loop dynamics, but the ideal output response is desired to be much less oscillatory, then the oscillatory modes must be cancelled to some extent by the input sequence. Unsurprisingly, this requirement is identical in concept to section 3.2. The only difference is that $a^+(z)$ is defined as the oscillatory poles. With this minor update, the proposed approach is identical to algorithm 1

5.2. *Numerical examples with under-damped behaviour*

This section will demonstrate the efficacy of algorithm 1 for systems with oscillatory dynamics as compared to a nominal PFC law is given in (3). The comparisons follow the same lines as those in section 4, that is, a comparison of the efficacy of λ as a tuning parameter. Two systems with significant overshoot and oscillation in

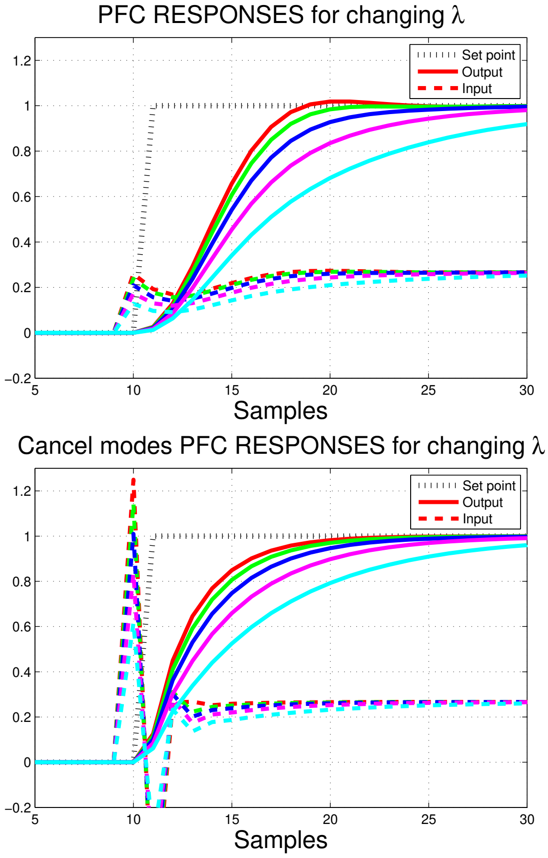

Figure 4. Behaviour of a nominal PFC algorithm and algorithm 1 for $n = 5, \lambda = [0.7, 0.75, 0.8, 0.85, 0.9]$ on model G_3 (the linking of line plots to λ should be obvious).

open-loop are considered:

$$G_3 = \frac{0.1z^{-1} + 0.2z^{-2}}{1 - 2.2z^{-1} + 1.76z^{-2} - 0.48z^{-3}}; \quad G_4 = \frac{-0.1z^{-1} + 0.4z^{-2} + 0.2z^{-3}}{1 - 2.5z^{-1} + 2.33z^{-2} - 0.801z^{-3}} \tag{16}$$

Figures 4, 5 shows the closed-loop responses for G_3, G_4 respectively.

- With a default PFC law of (1) it is clear that while the tuning parameter λ has some impact, nevertheless the impact is weak and moreover there is some overshoot and a noticeable lag in the output responses.
- Conversely, with the proposed PFC algorithm 1, the parameter λ has a much stronger impact and also the output responses are better; of course one pays for this, as expected, by seeing the oscillatory modes moved to the system inputs.

Remark 9 Algorithm 1 is making much more precise use of the system model information in order to improve performance and therefore one would expect the control loop to be more sensitive to modelling errors.

6. Conclusions and future work

This paper has shown how input prediction shaping can bring significant benefits to a PFC design in the case of systems with challenging open-loop dynamics.

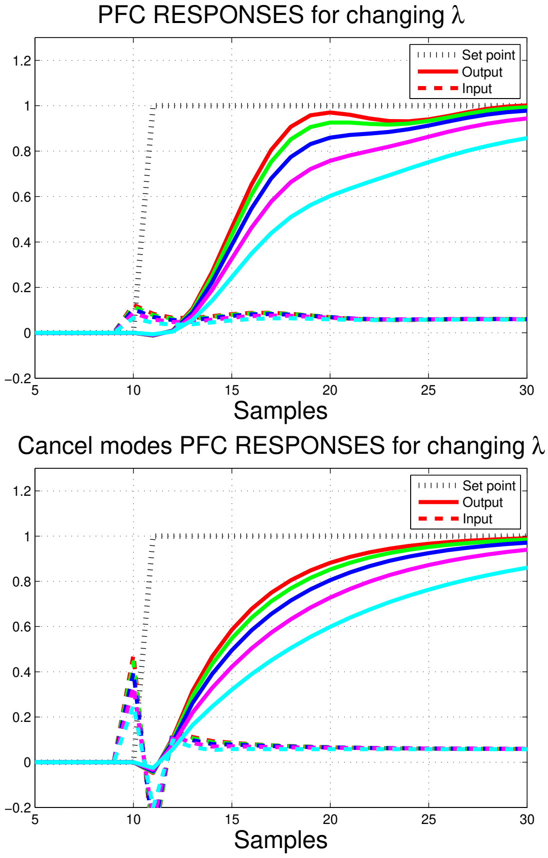

Figure 5. Behaviour of a nominal PFC algorithm and algorithm 1 for $n = 6, \lambda = [0.7, 0.75, 0.8, 0.85, 0.9]$ on model G_4 (the linking of line plots to λ should be obvious).

Specifically, by using a shaping which removes the undesirable dynamics from the predictions, the resulting d.o.f. is able to focus specifically on any remaining design criteria such as the desired closed-loop time constant, and indeed constraint handling. Moreover it retains core attributes of PFC which enable it to occupy a key niche in the market place. Key contributions are that:

(1) the proposed algorithm is relatively simple to code requiring just low order simultaneous equations and a simple loop for constraint handling.
(2) the efficacy of the main tuning parameter λ is significantly improved compared to the conventional algorithm.
(3) the approach includes constraint handling and moreover has recursive feasibility for unstable open-loop processes which is a strong result not available with the conventional approach.

Nevertheless, a number of key conceptual questions remain to be answered and constitute the future work in this area.

6.1. *Cascade structures*

It is clear that a simplistic PFC implementation of (1) using a predicted constant future input is fragile when the open-loop dynamics are not simple, but practitioners would argue that it is becoming more normal to do a two stage design whereby one pre-stabilises using some simple feedback and then uses PFC around the stabilised dynamic. Of course, such a concept is also implicit in dual-mode approaches to MPC (Rossiter et al. (1998)). Consequently, a key future work is to explore the extent to which such an approach is effective compared to the use of input shaping, for example: (i) how is the pre-stabilising loop controller selected? (ii) how are constraints handled efficiently with nested loops? (iii) what are the robustness characteristics of different approaches?

6.2. *More general shaping alternatives*

The input trajectory shaping of this paper was very precise and consequently could lead to quite aggressive input activity, as would be expected to achieve the target output behaviour and cancel out the undesirable modes in just a few control moves. In practice a user may desire less aggressive shaping which nevertheless could be used to improve the basic PFC algorithm. Moreover, the underlying PFC assumption of restricting oneself to a single move within the predictions is an artificial constraint that would often not be applied by humans and thus could be counter productive. Instead, we want to use input predictions with dynamics that evolve over many more samples (Rossiter et al. (2010)). The interesting question is what sort of dynamics (or functions) would be reasonable, and in the spirit of PFC, can this be done with simple code and in an intuitively logical manner. When human based control is deployed, the control action is linked to the desired speed of response as well as the underlying system dynamics:

- To go fast, over actuation is followed by a gradual move to the expected steady-state.
- To achieve open-loop dynamics, move the input directly to the expected steady-state.
- If slower than open-loop dynamics is allowed, then move the input gradually to the steady-state (using the desired time constant within this input sequence).

Clearly these concepts deserve further study.

References

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.

Clarke, D.W., Mohtahdi, C, Tuffs, P., Generalised predictive control, Automatica, 23, pp137-160, 1987.

Cutler, C.R. and Ramaker B.L., Dynamic matrix control a computer control algorithm, ACC, 1980.

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.

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.

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.

Maciejowski, J., Predictive control with constraints, Prentice Hall, 2001.

Rawlings, J.B.; Muske, K.R. The stability of constrained receding horizon control. Trans. IEEE Autom. Control,1993, 38, 1512–1516.

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

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

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

Rossiter, J.A., Notes on multi-step ahead prediction based on the principle of concatenation, Proceedings IMechE., 1993, 207, 261-263.

Rossiter, J.A., J.R. Gossner and B.Kouvaritakis, Constrained cautious stable predictive control, 1997, Proc. IEE Pt.D, 144, 4, pp313-323

Rossiter, J.A.; Richalet, J. Handling constraints with predictive functional control of unstable processes. *Am. Control Conf.*, 2002, 6, 4746–4751.

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

Rossiter, J.A.; Rice, M.J.; Kouvaritakis, B. A numerically robust state-space approach to stable predictive control strategies, Automatica, 1998, 34, 65–73.

Rossiter, J.A., L. Wang, G. Valencia-Palomo, Efficient algorithms for trading off feasibility and performance in predictive control, IJC, 83,4,789, 2010

Rossiter, J.A. and Haber, R., The effect of coincidence horizon on predictive functional control, in Process, Processes, 2015, Vol.3, Issue 1, p25-45, doi: 10.3390/pr3010025

Skrjanc, I., Model predictive control for processes with unstable poles, Asian Journal of Control, 10, 4, pp1–7, 2008

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