processes

MDPI

Article

Input Shaping Predictive Functional Control for Different Types of Challenging Dynamics Processes

Muhammad Abdullah [1,2] and John Anthony Rossiter [1,*]

[1] Department of Automatic Control and System Engineering, University of Sheffield, Mappin Street, Sheffield S1 3JD, UK; MAbdullah2@sheffield.ac.uk

[2] Department of Mechanical Engineering, International Islamic University Malaysia, Jalan Gombak, Kuala Lumpur 53100, Malaysia

[*] Correspondence: j.a.rossiter@sheffield.ac.uk

Received: 2 July 2018; Accepted: 23 July 2018; Published: 7 August 2018

check for updates

Abstract: Predictive functional control (PFC) is a fast and effective controller that is widely used for processes with simple dynamics. This paper proposes some techniques for improving its reliability when applied to systems with more challenging dynamics, such as those with open-loop unstable poles, oscillatory modes, or integrating modes. One historical proposal considered is to eliminate or cancel the undesirable poles via input shaping of the predictions, but this approach is shown to sometimes result in relatively poor performance. Consequently, this paper proposes to shape these poles, rather than cancelling them, to further enhance the tuning, feasibility, and stability properties of PFC. The proposed modification is analysed and evaluated on several numerical examples and also a hardware application.

Keywords: predictive control; unstable; underdamped; integrating; input shaping

1. Introduction

Predictive functional control (PFC) is a simple controller that is effective for small-scale single-input-single-output (SISO) applications, especially for low-order and stable processes [1–3]. The main advantages of PFC compared to its prime competitor—that is, proportional integral derivative (PID)—are its ability to handle constraints and its transparent tuning parameters. Indeed, it must be emphasised that the performance of PFC should not be benchmarked against more advanced model predictive control (MPC) strategies [4], because the implementation is relatively much cheaper and requires only low computation with very straightforward coding [5,6].

Nevertheless, despite its apparently attractive benefits, PFC has received relatively little attention in the academic literature because of the lack of a priori stability guarantees [7,8], which are possible with more advanced MPC approaches. Consequently, several recent works have developed the basic concept of PFC to improve its overall tuning properties while providing a confident assurance in the resulting closed-loop performance and constraint handling [9–12]. However, most of these modifications perform well only with low-order and simple dynamical processes. For a system with open-loop unstable poles, significant underdamping, or integrating dynamics, PFC is quite challenging to tune effectively [13], and the resulting divergent or oscillating predictions may give rise to infeasibility and/or robustness issues.

PFC practitioners often deploy a type of cascade structure to handle a challenging dynamics process, where an inner loop is used to improve the dynamics for an outer loop to control [14,15]. This modification enables a user to retain an independent model (IM) structure that handles uncertainties, while retaining a similar tuning concept as the conventional approach. Nevertheless, the inner PFC can only accept a proportional-type controller to avoid any overcomplication when handling constraints [6]. This restriction makes it difficult to have a systematic selection of gain, rather than an ad hoc approach. Moreover, a back-calculation or anti-wind-up technique is required to avoid any saturation while satisfying the constraints [6], which also can be conservative and thus affect performance.

A fundamental conceptual weakness of a simplistic PFC approach is that one is basing decisions on an open-loop prediction, which may have undesirable, possibly divergent, dynamics; matching this to a desirable closed-loop dynamic will lead to an ill-posed approach. Hence, an alternative approach is to pre-stabilise/pre-shape the output predictions by shaping the future input dynamics so that the effect of unwanted poles on the predictions are alleviated or cancelled [16]. This modification can retain the systematic tuning concept of PFC, in addition to facilitating a recursive feasibility guarantee feature for constraint handling. However, the performance of this control law is not always desirable, since the cancellation of specified modes (poles) within the predictions using a minimal number of control input changes often requires an aggressive input trajectory [16–18], which is not implementable or ideal for some cases.

In practice, less aggressive shaping to remove the undesirable modes from the prediction [19] is preferable, and this forms the main thrust of the proposal in this paper. More specifically, the proposal here is that, rather than using a small, finite number of control moves (effectively, finite impulse response (FIR) parameterisation), such as in Generalized Predictive Control (GPC) Added the defination and conventional PFC, the future input moves will be parameterised using an infinite impulse response (IIR) instead. The preference for IIR over FIR is due to IIR being more convenient to manipulate and define than a high-order FIR, in general. In turn, by allowing the output modes to evolve over many more samples, the required input will be less aggressive. Nevertheless, due to the desire for simplicity and transparency that is a central tenet of PFC, in this paper, we choose not to generalise the parameterisation for different dynamics. Instead, this work seeks to provide some rigour on how to effectively and systematically shape the future input for a given dynamic and, moreover, how to ensure some recursive feasibility properties during constraint handling.

Section 2 of this paper presents a brief formulation of conventional PFC. Section 3 introduces the concept of input shaping PFC, together with the constraint handling approach. Section 4 demonstrate the proposed algorithms on several numerical examples and also on some laboratory hardware. Section 5 provides the conclusions.

2. Conventional PFC

This section presents, in brief, the main concepts, notation, and formulation of PFC, together with a systematic constraint handling approach. For more detailed derivations, theory, and concepts of PFC, an interested reader may refer to the references, e.g., [5,6,13,20]. Without loss of generality, a controlled autoregressive and integrated moving average (CARIMA)-based model is used instead of the standard independent model (IM) structure to derive the required unbiased predictions, as this form is more amenable to the algebra required to implement the shaping. Hence, the model will take the form:

$$a(z)y_k + b(z)u_k + \frac{\zeta_k}{\Delta(z)}; \quad \Delta(z) = 1 - z^{-1} \tag{1}$$

where $b(z) = b_1 z^{-1} + \ldots$, $a(z) = 1 + a_1 z^{-1} + \ldots$, y_k, u_k are the outputs and inputs, respectively, at sample k, and ζ_k is an unknown zero mean random variable used to capture uncertainty.

2.1. Control Law

PFC design is based on the assumption that a closed-loop response should follow a first-order dynamic from the current state y_k to the desired target r [20]. In practice, one aims to achieve this by ensuring a matching using the open-loop prediction, but only at a single point. In other words, the predicted output trajectory is chosen to satisfy the following equality:

$$y_{k+n|k} = (1-\lambda^n)r + \lambda^n y_k \tag{2}$$

where $y_{k+n_y|k}$ is the n-step ahead system prediction at sample time k, and λ is the desired closed-loop pole that will determine the speed of convergence. PFC practitioners typically select the desired closed-loop time response (CLTR) which has an explicit link with the corresponding closed-loop pole, that is, $\lambda = e^{\frac{-3T}{CLTR}}$, where T is the sampling period [6].

The predictions for the CARIMA model (1) are standard in the literature (e.g., [4,21]), so only the final form is given here. For input increments Δu_{k+i}, the n-step ahead future output prediction is formed as:

$$\underset{\rightarrow}{y}_{k+1|k} = H\Delta\underset{\rightarrow}{u}_k + P\Delta\underset{\leftarrow}{u}_k + Q\underset{\leftarrow}{y}_k \tag{3}$$

where the left and right arrow underlying vectors represent past and future variables, respectively. The parameters H, P, Q depend on the model parameters, and for a model of order m:

$$\Delta\underset{\rightarrow}{u}_k = \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+n-1} \end{bmatrix}; \Delta\underset{\leftarrow}{u}_k = \begin{bmatrix} \Delta u_{k-1} \\ \Delta u_{k-2} \\ \vdots \\ \Delta u_{k-m} \end{bmatrix}; \underset{\leftarrow}{y}_k = \begin{bmatrix} y_k \\ y_{k-1} \\ \vdots \\ y_{k-m} \end{bmatrix}; \; \underset{\rightarrow}{y}_{k+1|k} = \begin{bmatrix} y_{k+1} \\ y_{k+2} \\ \vdots \\ y_{k+n} \end{bmatrix} \tag{4}$$

In conventional PFC [6,20], within the predictions we select $\Delta u_{k+i} = 0, \; i > 0$. Using this and substituting the n-step ahead prediction from (3) into equality (2) gives:

$$H_n \mathbf{e}_1 \Delta u_k + P_n \Delta\underset{\leftarrow}{u}_k + Q_n \underset{\leftarrow}{y}_k = (1-\lambda^n)r + \lambda^n y_k \tag{5}$$

where $\mathbf{e}_i$ is the ith standard basic vector and $H_n = \mathbf{e}_n^T H, P_n = \mathbf{e}_n^T P, Q_n = \mathbf{e}_n^T Q$. Hence, the PFC control law is given as:

$$\Delta u_k = \frac{1}{h_n}\left[(1-\lambda^n)r + \lambda^n y_k - Q_n \underset{\leftarrow}{y}_k - P_n \Delta\underset{\leftarrow}{u}_k\right]; \quad h_n = H_n \mathbf{e}_1 \tag{6}$$

Remark 1. *Figure 1 shows a comparison of simplified flow diagrams, where both PFC and MPC share the same structure, yet have a different optimisation process, where the constraint handling is embedded inside the main block. As for PID, the control input is obtained simply by tuning the gains, while the constraints are handled via a rule base [22].*

Remark 2. *It is noted that PFC performs well for a system with close to a monotonic step response, such as a first-order system and overdamped second-order dynamics, assuming, of course, a sensible choice of the tuning parameters λ and n [11–13]. However, the same tuning procedure may not work for systems with less simple open-loop dynamics, leading to ill-posed decision making and unreliable control.*

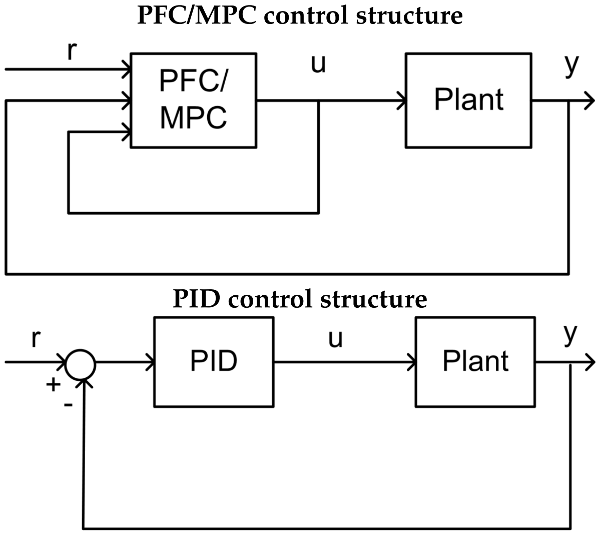


Figure 1. Comparison of flow diagrams for predictive functional control (PFC), model predictive control (MPC), and proportional integral derivative (PID).

2.2. Constraint Handling

The constraint handling approach presented here is adapted from the standard MPC literature [21,23] but with simpler code, and it is more systematic and less conservative than the back-calculation typically used in conventional PFC algorithms [24]. Similarly, it will be more efficient than the ad hoc approaches used with PID. Assume constraints, at every sample, on input rate, input, and states, as follows:

$$\underline{\Delta u} \leq \Delta u_k \leq \overline{\Delta u}; \quad \underline{u} \leq u_k \leq \overline{u}; \quad \underline{y} \leq y_k \leq \overline{y}, \ \forall k \tag{7}$$

The user needs to:

1. Find a suitable horizon m [23] over which to compute the entire set of future output predictions in a single vector: $\underset{\rightarrow k+1|k}{y} = H\mathbf{e_1}\Delta u_k + P\Delta \underset{\leftarrow}{u}_k + Q\underset{\leftarrow}{y}_k$. The horizon for output predictions $\underset{\rightarrow k+1}{y}$, and thus the row dimension of H, should be long enough to capture all core dynamics!
2. Combine the input constraints, rate constraints, and output predictions into a single set of linear inequalities:
$$C\Delta u_k \leq \mathbf{f}_k \tag{8}$$
$$C = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \\ H\mathbf{e_1} \\ -H\mathbf{e_1} \end{bmatrix}; \quad \mathbf{f}_k = \begin{bmatrix} \overline{u} - u_{k-1} \\ -\underline{u} + u_{k-1} \\ \overline{\Delta u} \\ -\underline{\Delta u} \\ L\overline{y} - P\Delta \underset{\leftarrow}{u}_k - Q\underset{\leftarrow}{y}_k \\ -L\underline{y} + P\Delta \underset{\leftarrow}{u}_k + Q\underset{\leftarrow}{y}_k \end{bmatrix}; \quad L = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$
where $\mathbf{f}_k$ depends on past data in $\Delta \underset{\leftarrow}{u}_k, \underset{\leftarrow}{y}_k$ and on the limits.
3. The input/output predictions will satisfy constraints if inequalities (8) are satisfied, and thus the PFC algorithm should consider these explicitly.

Next, a single simple loop is utilised to find the Δu_k closest to the unconstrained solution of (6), while satisfying (8).

Algorithm 1. Simple PFC algorithm with systematic constraint handling

```
At each sample:

1. Define the unconstrained value for $\Delta u_k$ from (6).
2. Define the vector $\mathbf{f}_k$ of (8) (it is noted that $C$ does not change).
3. Use a simple loop covering all the rows of $C$ as follows:

    (a) Check the ith constraint that is the ith row $\mathbf{e}_i^T C\Delta u_k \leq \mathbf{f}_{k,i}$ of $C\Delta u_k \leq \mathbf{f}_k$.
    (b) If $\mathbf{e}_i^T C\Delta u_k > f_{k,i}$, then set $\Delta u_k = (f_{k,i})/[\mathbf{e}_i^T C]$, else leave $\Delta u_k$ unchanged.
```

Remark 3. *For a simple and stable open-loop process and for suitably large m, Algorithm 1 is guaranteed to be recursively feasible and, moreover, to converge to a possible value for Δu_k that is closest to the unconstrained choice [12]. However, this benefit does not apply to systems with more challenging dynamics, such as when the open-loop prediction is divergent.*

3. Input Shaping PFC

This section presents the concept of input shaping to pre-stabilise (or pre-condition) the open-loop predictions of processes with undesirable dynamics. The shaping of input predictions can be done either via explicit pole cancellation or pole shaping, and both methods can be used to formulate a PFC control law which retains a recursively feasible constrained solution. A key issue, however, is whether one deploys FIR or IIR parameterisations of the predictions for the future input increments $\Delta u_{k+i|k}$, $i \geq 0$.

3.1. Pre-Stabilisation of Predictions via Pole Cancellation

For systems with poor open-loop dynamics, the constant input assumption of typical PFC does not provide enough flexibility within the predicted input to both cancel the effect of an undesirable pole and to get nice convergent behaviour [13,17,19]. Thus, it is crucial to first stabilise the prediction before implementing it into a control law [25]. The first step is to factorise the poles in the denominator:

$$\Delta y(z) = \frac{b(z)}{a(z)}\Delta u(z); \quad a(z) = a^-(z)a^+(z) \tag{9}$$

where $a^+(z)$ contains the undesirable poles. Utilising the Toeplitz/Hankel form [21], the future output predictions can be computed as:

$$[C_{a^-\Delta}][C_{a^+}]\underset{\rightarrow k+1}{y} + H_A \underset{\leftarrow k}{y} = C_b \Delta \underset{\rightarrow k}{u} + H_b \Delta \underset{\leftarrow k}{u} \tag{10}$$

where for general polynomial $f(z) = f_0 + f_1 z^{-1} + ... + f_n z^{-n}$,

$$C_f = \begin{bmatrix} f_0 & 0 & 0 & \dots \\ f_1 & f_0 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots \\ f_n & f_{n-1} & f_{n-2} & \dots \end{bmatrix}; H_f = \begin{bmatrix} f_1 & f_2 & \dots & f_n \\ f_2 & f_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

Rearranging prediction (10) in more compact form, we get:

$$\underset{\rightarrow k+1}{y} = [C_{a^-\Delta}]^{-1}[C_{a^+}]^{-1}[C_b \Delta \underset{\rightarrow k}{u} + \underbrace{H_b \Delta \underset{\leftarrow k}{u} - H_A \underset{\leftarrow k}{y}}_{\mathbf{p}}] \tag{11}$$

from which the presence of the undesirable modes are transparent through the factor $[C_{a^+}]^{-1}$.

Lemma 1. *Selection of the future input sequence* $\Delta \underset{\rightarrow k}{u}$, *at each sample, such that the following equality is satisfied:*

$$[C_b \Delta \underset{\rightarrow k}{u} + \mathbf{p}] = C_{a^+}\gamma \tag{12}$$

where γ *is a convergent sequence or a polynomial, will ensure that the corresponding output predictions in (11) do not contain the undesirable modes in* a^+.

Proof. This is self-evident by substitution of (12) into (11), which gives:

$$\underset{\rightarrow k+1}{y} = [C_{a^-\Delta}]^{-1}[C_{a^+}]^{-1}[C_{a^+}]\gamma = [C_{a^-\Delta}]^{-1}\gamma \tag{13}$$

so that only the acceptable modes in $a^-(z)$ remain in the predictions, along with any components in the convergent sequence γ. It is noted that this choice automatically includes the initial conditions within $\mathbf{p}$ and thus updates each sample as required. □

Remark 4. *Requirement (12) can be solved by a small number of simultaneous equations [21], where the minimal-order solution can be represented as:*

$$\Delta \underset{\rightarrow k}{u} = P_1\mathbf{p}; \quad \gamma = P_2\mathbf{p} \tag{14}$$

for suitable P_1, P_2. *The required dimension of non-zero elements in vector* $\Delta \underset{\rightarrow k}{u}$ *corresponds to at least one more than the number of undesirable modes* (n_{a^+}), *while the order of* γ *is usually taken as* $n_\gamma = n_p - n_{a^+}$, *where* n_p *is the effective dimension of* $\mathbf{p}$ *(which depends upon the column dimensions of* H_b, H_A*).*

To ensure the future manipulated control moves are convergent, while adding some flexibility for modifying the output predictions, the input requirement in (14) can be enhanced to:

$$\Delta \underset{\rightarrow k}{u} = P_1\mathbf{p} + C_{a^+}\phi \tag{15}$$

where the vector parameter ϕ denotes the degrees of freedom (d.o.f.) within the predictions.

Theorem 1. *Using the new shaped input (15) ensures that the undesirable modes do not appear in the output predictions, irrespective of the choice of* ϕ. *The output predictions are convergent if* ϕ *is finite-dimensional or a convergent infinite dimensional sequence.*

Proof. Substitute input prediction (15) into output prediction (12), and the predictions become:

$$\begin{aligned} \underset{\rightarrow k+1}{y} &= [C_{a^-\Delta}]^{-1}[C_{a^+}]^{-1}[C_b\Delta \underset{\rightarrow k}{u} + \mathbf{p}] \\ &= [C_{a^-\Delta}]^{-1}[C_{a^+}]^{-1}[C_{a^+}\gamma + C_bC_{a^+}\phi] \\ &= [C_{a^-\Delta}]^{-1}[C_b\phi + \gamma] \end{aligned} \tag{16}$$

The prediction can be represented with an equivalent z-transform:

$$\underset{\rightarrow}{y}(z) = \frac{[1, z^{-1}, z^{-2}, \cdots][\gamma + C_b\phi]}{a^-(z)} = \frac{\gamma(z) + b(z)\phi(z)}{a^-(z)} \tag{17}$$

It is known from Lemma 1 that the contribution from γ gives a convergent prediction, and thus overall convergence is obvious as long as $\phi(z)$ is convergent (or an FIR). □

Remark 5. *Noting the definition of* $\mathbf{p}$ *in (11), the n-step ahead output prediction with prediction class (15) and (17) can be put in a more common form as:*

$$y_{k+n|k} = H_{s,n}\phi + P_{s,n}\Delta \underset{\leftarrow}{u}_k + Q_{s,n}\underset{\leftarrow}{y}_k \tag{18}$$

where $H_{s,n}$, $P_{s,n}$, and $Q_{s,n}$ are suitable matrices, and the additional subscript 's' is used to denote shaping *and ϕ is taken to be FIR (equivalently a finite dimensional vector). Note, however, that typically for PFC, ϕ is a scalar. Also, it is easy to show [18] that choosing $\phi = 0$ will automatically give the same input predictions as those deployed at the previous sample, which enables consistency of predictions from one sample to the next.*

3.2. Pre-Stabilisation via Pole Shaping

It is known that dead-beat pole cancellation can require aggressive inputs, and the minimal-order solutions to (12) are in effect dead-beat input predictions [16,19]. Although dead-beat solutions are easy to define and thus have some advantages in terms of computation and transparency, in practice, a user may desire a less aggressive shaping that is more implementable in a real system. Alongside this, the popularity of dual mode approaches in the literature [26] is partially because they allow the implied input predictions to converge to the steady state asymptotically, rather than in a small, finite number of steps. Thus, a logical question to ask is whether a smoother solution to (12)—that is, one where the implied solutions for $\gamma(z)$, $\phi(z)$ used in (17) have some poles, say $\alpha(z)$—would work better for PFC.

The mainstream MPC community has focussed on optimal control solutions, but, given that PFC is intended to be simple and low-dimensional, the proposal here is that it is more reasonable to investigate the potential of simple default choices for the asymptotic dynamics $\alpha(z)$ within the input and output predictions. Clearly, this choice can be strongly linked to the target closed-loop behaviour and/or system knowledge.

Proposal 1. *By definition, the integrator has a pole on the unit circle—that is, factor $(1 - z^{-1})$—and, conversely, cancelling the pole as in (12) is equivalent to enforcing a pole on the origin—that is, factor $(1 - 0\,z^{-1})$. Hence, the choice of pole factor $\alpha = (1 - 0.5\,z^{-1})$ represents a simple half-way house trade-off between these two choices.*

Proposal 2. *For a process with significant underdamping, the implied $\alpha(z)$ will have only real poles which are chosen to be close to the real parts of the oscillatory poles. This will reduce the undesirable oscillation in the output predictions, but not change the convergence speed, albeit the input may then be somewhat oscillatory.*

Proposal 3. *For open-loop unstable systems, a simple default solution simply inverts the unstable poles, that is, defining $\alpha(z)$ such that $a^+(z_i) = 0 \;\Rightarrow\; \alpha(1/z_i) = 0$.*

Lemma 2. *The dynamics $\alpha(z)$ will be present in the predictions if the following Diophantine equation is used to solve the input/output prediction pairing.*

$$b(z)w(z) + \alpha(z)p(z) = a^+(z)\hat{\gamma}(z); \quad p(z) = [1, z^{-1}, ...]\mathbf{p} \tag{19}$$

$$\Rightarrow \quad \Delta \underset{\rightarrow}{u}(z) = \frac{w(z)}{\alpha(z)}, \quad \underset{\rightarrow}{y}(z) = \frac{\hat{\gamma}(z)}{a^-(z)\Delta(z)\alpha(z)}$$

Proof. First, note that (19) is equivalent to solving:

$$[C_b C_\alpha^{-1}\mathbf{w} + \mathbf{p}] = C_{a^+}C_\alpha^{-1}\hat{\gamma} \tag{20}$$

and, moreover, Equation (20) follows directly from enforcing (12) while assuming $\Delta \underset{\rightarrow k}{u} = C_\alpha^{-1}\mathbf{w}$. Hence, substituting this $\Delta \underset{\rightarrow k}{u}$ into (11) gives:

$$\begin{aligned} \underset{\rightarrow k+1}{y} &= [C_{a^-\Delta}]^{-1}[C_{a^+}]^{-1}[C_b \Delta \underset{\rightarrow k}{u} + \mathbf{p}] \\ &= [C_{a^-\Delta}]^{-1}[C_{a^+}]^{-1}[C_b C_\alpha^{-1}\mathbf{w} + \mathbf{p}] \\ &= [C_{a^-\Delta}]^{-1}[C_{a^+}]^{-1}[C_{a^+}C_\alpha^{-1}\hat{\gamma}] = C_{a^-\Delta}^{-1}C_\alpha^{-1}\hat{\gamma} \end{aligned} \tag{21}$$

It is evident, therefore, that the desired poles are in the predictions for both the input and output. □

Remark 6. *The new requirement (20) can be solved similarly to (12), where the minimal-order solution for* $\mathbf{w}$ *and* $\hat{\gamma}$ *are:*

$$\mathbf{w} = \hat{P}_1\mathbf{p}; \quad \hat{\gamma} = \hat{P}_2\mathbf{p} \tag{22}$$

Theorem 2. *A convergent prediction class which embeds both the desired asymptotic poles and some degrees of freedom (d.o.f.) can be defined from:*

$$\mathbf{w} = \hat{P}_1\mathbf{p} + C_{a^+}\phi; \quad \Delta \underset{\rightarrow k}{u} = [C_\alpha]^{-1}[\hat{P}_1\mathbf{p} + C_{a^+}\phi] \tag{23}$$

where convergent IIR or FIR ϕ *constitutes the d.o.f.*

Proof. This is analogous to Theorem 1 and is based on superposition. The additional component in $\mathbf{w}$—that is, $C_{a^+}\phi$—necessarily cancels the undesirable poles and gives overall convergent output predictions. So, using (21), then: Added hat on top of gamma in Equation (24).

$$\begin{aligned} \underset{\rightarrow k+1}{y} &= C_{a^-\Delta}^{-1}C_{a^+}^{-1}[C_b\Delta \underset{\rightarrow k}{u} + \mathbf{p}] \\ &= C_{a^-\Delta}^{-1}C_{\alpha}^{-1}\hat{\gamma} + C_{a^-\Delta}^{-1}C_{a^+}^{-1}C_{\alpha}^{-1}[C_{a^+}C_b\phi] \\ &= C_{a^-\Delta}^{-1}C_{\alpha}^{-1}[\hat{\gamma} + C_b\phi] \end{aligned} \tag{24}$$

□

Remark 7. *By extracting the* n^{th} *row and noting the definition of* $\mathbf{p}$ *in (11), the n-step ahead prediction from (24) can be rearranged in a more general form as:*

$$y_{k+n|k} = h_{n,\alpha}\phi + P_{n,\alpha}\Delta\underset{\leftarrow}{u_k} + Q_{n,\alpha}\underset{\leftarrow}{y_k} \tag{25}$$

for suitable $h_{n,\alpha}$, $P_{n,\alpha}$, $Q_{n,\alpha}$, *and it is noted that as is conventional for PFC,* ϕ *has just a single non-zero parameter in order to retain computational simplicity and to have just a single d.o.f. for satisfying the control law (2).*

3.3. Proposed Shaping PFC Control Laws

Since the shaped predictions of (18) and (25) are derived in a general form, two new control laws—Pole Cancellation PFC (PC-PFC) and Pole Shaping PFC (PS-PFC)—can be formulated in a conventional manner after selecting a suitable coincidence horizon n and desired closed-loop pole λ.

[PC-PFC] The d.o.f ϕ is computed by substituting prediction (18) of PC-PFC into equality (2), and thus:

$$\phi = \frac{1}{h_{n,s}}\left[(1-\lambda^n)r + \lambda^n y_k - Q_{n,s}\underset{\leftarrow}{y_k} - P_{n,s}\Delta\underset{\leftarrow}{u_k}\right] \tag{26}$$

then, the current input increment Δu_k is determined simply by inserting ϕ into the predicted input of (15).

[PS-PFC] The d.o.f ϕ is computed by substituting prediction (25) of PS-PFC into equality (2), and thus:

$$\phi = \frac{1}{h_{n,\alpha}}\left[(1-\lambda^n)r + \lambda^n y_k - Q_{n,\alpha}\underset{\leftarrow}{y_k} - P_{n,\alpha}\Delta\underset{\leftarrow}{u_k}\right] \tag{27}$$

then, the current input increment Δu_k is determined simply by inserting ϕ into the predicted input of (23).

3.4. Constraint Handling Approaches with Recursive Feasibility

A core advantage of MPC, in general, is that the optimised predictions can be restricted to ones which satisfy constraints; the d.o.f. within the predictions are used to ensure constraint satisfaction. For PC-PFC and PS-PFC, the d.o.f. in the predictions is the variable ϕ. This section gives a brief overview of how the constraint inequalities ensuring (7) depend upon ϕ. We will use PS-PFC and assume that the reader can easily find the equivalent matrices for PC-PFC (for which, in effect, $\alpha = 1$).

Noting the definition of future input increments in (23) and output predictions in (25), the constraints inequalities for (7) can be defined as:

$$\begin{aligned} L\underline{\Delta u} &\le C_\alpha^{-1}[\hat{P}_1\mathbf{p} + C_{a+}\phi] \le L\overline{\Delta u}; \\ L\underline{u} &\le C_{I/\Delta}C_\alpha^{-1}[\hat{P}_1\mathbf{p} + C_{a+}\phi] + Lu_{k-1} \le L\overline{u}; \\ L\underline{y} &\le H_\alpha\phi + P_\alpha \underset{\leftarrow}{\Delta u}_k + Q_\alpha \underset{\leftarrow}{y}_k \le L\overline{y}. \end{aligned} \tag{28}$$

where $C_{I/\Delta}$ is a lower triangular matrix one ones, and L is a vector of ones with an appropriate dimension (typically a horizon long enough to capture the core dynamics in the predictions). The reader should note that the horizon for the predictions used in (28) will, in general, be much longer than the coincidence horizon used in (27), as one needs to ensure that the implied long-range predictions satisfy constraints. The inequalities can be combined for convenience as follows, although this is not necessary for online coding where efficient alternatives may exist:

$$C\phi \le \mathbf{f}_k \tag{29}$$

$$C = \begin{bmatrix} C_{I/\Delta}C_\alpha^{-1}C_{a+} \\ -C_{I/\Delta}C_\alpha^{-1}C_{a+} \\ C_\alpha^{-1}C_{a+} \\ -C_\alpha^{-1}C_{a+} \\ H_\alpha \\ -H_\alpha \end{bmatrix}; \quad \mathbf{f}_k = \begin{bmatrix} L[\overline{u} - u_{k-1}] - C_{I/\Delta}C_\alpha^{-1}\hat{P}_1\mathbf{p} \\ L[-\underline{u} + u_{k-1}] + C_{I/\Delta}C_\alpha^{-1}\hat{P}_1\mathbf{p} \\ L\overline{\Delta u} - C_\alpha^{-1}\hat{P}_1\mathbf{p} \\ -L\underline{\Delta u} + C_\alpha^{-1}\hat{P}_1\mathbf{p} \\ L\overline{y} - P_\alpha\underset{\leftarrow}{\Delta u}_k - Q_\alpha\underset{\leftarrow}{y}_k \\ -L\underline{y} + P_\alpha\underset{\leftarrow}{\Delta u}_k + Q_\alpha\underset{\leftarrow}{y}_k \end{bmatrix}$$

Algorithm 2. [PS-PFC with constraint handling]

```
At each sample:
1. Define the unconstrained value for $\phi$ from (27).
2. Update the vector $\mathbf{f}_k$ of (29) (it is noted that $C$ does not change).
3. Use a simple loop covering all the rows of $C$ as follows:
    (a) Check satisfaction of the ith constraint using: $\mathbf{e}_i^T C\phi \le f_{k,i}$.
    (b) If $\mathbf{e}_i^T C\phi > f_{k,i}$, then set $\phi = (f_{k,i})/[\mathbf{e}_i^T C]$, else leave $\phi$ unchanged.
```

Theorem 3. *In the presence of constraints, Algorithm 2 is recursively feasible where the computed ϕ will not only enforce the input/output predictions to satisfy constraints at the current sample, but will also guarantee 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 component $\hat{P}_1\mathbf{p}$ 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 sensible state. □

Readers should note that using the pre-stabilised/shaped predictions is essential for this recursive feasibility result, which is not available for more conventional PFC approaches, for which the implied long-range predictions may be divergent. Thus, this Theorem is an important contribution of this paper.

Remark 8. *Although Algorithm 2 allows recursive feasibility, which is a strong result, ironically, the use of PC-PFC or PS-PFC does not not give any a priori stability and/or performance guarantees in general, which is a well-understood weakness of PFC approaches [7] and a consequence of wanting a very simple and cheap control approach.*

4. Numerical Examples

This section presents several numerical examples of the proposed Pole Shaping PFC (PS-PFC) in handling different types of challenging dynamics, while comparing its performance with the Pole Cancellation PFC (PC-PFC) and conventional PFC (PFC). These examples will highlight:

- the impact of input shaping on the open-loop behaviour;
- the trade-off in the closed-loop performance;
- the efficacy of constraint handling;
- the efficacy on laboratory hardware.

For demonstration purposes, the first three processes with varying dynamics are investigated in a MATLAB simulation environment in Sections 4.1–4.4. The final example in Section 4.5 is on laboratory hardware.

4.1. Description of Case Studies

The case studies presented here are inspired from the real process applications. However, for clarity of presentation, the model parameters are not specific to a given piece of apparatus, but rather are generic to attain suitable dynamics which enable an explicit comparison between the control laws. In this work, it is assumed that there is no plant model mismatch. The robustness properties of these controllers will remain as future work, although it is noted that, as with most predictive controllers, disturbance rejection and offset free tracking is implicit.

4.1.1. Case Study 1: Boiler Level Control

In the process industry, the use of a boiler is frequent, and the level of water needs to be controlled within the manufacturer's specified limits. Exceeding the allowable limits may lead to water overflow, overheating, and/or damage to many components. Conversely, if the level is low, the water wall tubes may overheat and cause tube ruptures, resulting in expensive repairs and other potential hazards. Hence, the prime control objective is to raise the water level at the boiler start-up point while retaining it at a constant steam load. Since the conversion process from water to steam is very slow, a typical model for this process is usually a first-order system with an integrator and stable zero [27]. In a discrete form, one of the poles should reside in a unit circle. The relationship between the output water level (m), $y(z)$, and the input water flow rate (m^3 s^{-1}), $u(z)$, can be represented by a representative model, such as G_1:

$$G_1 = \frac{0.1z^{-1} + 0.4z^{-2}}{(1-0.8z^{-1})(1-z^{-1})} \tag{30}$$

4.1.2. Case Study 2: Depth Control of Unmanned Free-Swimming Submersible (UFSS)

In the marine application, the depth of an unmanned submarine can be controlled by deflecting its elevator surface, whereby the vehicle will rotate about its pitch axis; the associated vertical forces due to the water flow beside the vehicle enable the vehicle to sink or rise. Since a step input deflection may create an oscillatory angle of dive due to the water current, typical dynamics to represent this system often consist of at least one stable pole and two complex poles with stable zeros [22]. Thus a representative third-order underdamped process G_2 can be assumed to represent this pitch control system:

$$G_2 = \frac{0.85z^{-1} - 1.5z^{-2} + 0.85z^{-2}}{(1-0.6z^{-1})(1-1.6z^{-1}+0.8z^{-2})} \tag{31}$$

with the output as the pitch angle (rad) and input as the input elevator deflection (m).

4.1.3. Case Study 3: Temperature Control of Fluidised Bed Reactor

A fluidised bed reactor is used to produce a variety of multiphase chemical reactions that are highly exothermic and can be considered as unstable. The reactor bed temperature needs to be controlled by manipulating the coolant flow rate to avoid overheating and other potential hazards. In this case, a drastic change in flow rate will trigger a reaction between the chemicals that releases extra energy and increases the bed temperature. In fact, the change in flow rate needs to follow specific dynamics to avoid this reaction while stabilising the temperature. A reduced control model includes at least one stable pole, typically, and one unstable pole [28] to relate the dynamics between the output temperature (°C) and input coolant flow rate ($m^3\ s^{-1}$). Inspired by this system, a representative second-order unstable process G_3 is considered as a good case study:

$$G_3 = \frac{0.2z^{-1} - 0.26z^{-2}}{(1 - 0.9z^{-1})(1 - 1.5z^{-1})} \tag{32}$$

4.2. The Impact of Input Shaping on Predictions and Feasibility

The prime purpose of shaping the future input dynamics is to eliminate the effect of unwanted poles in the future predictions. Nevertheless, it is also undesirable to have an overaggressive input activity, which may not be implementable in a real plant. To analyse this issue, the prediction behaviour of PFC, PC-PFC, and PS-PFC for processes G_1, G_2, and G_3 are plotted in Figure 2. From these results, it can be observed that:

- For an integrating process, such as G_1, the constant input prediction of PFC leads to a divergent output prediction, and thus, output constraints can only be satisfied if the input is selected to be zero! Hence, the PFC plots do not appear in this example, as the constraint handling forces a choice of $u_k = 0, \forall k$.
- For G_1, the default input prediction (Equation (15)) for PC-PFC (blue-dotted line) is of dead-beat form and aggressive, whereas the input prediction (Equation (23)) for PS-PFC moves smoothly to the steady state and is less aggressive. There is no significant difference in the corresponding output predictions.
- For the underdamped system G_2, the output prediction from PFC includes a significant oscillation, which is undesirable and will also cause constraint handling to be conservative. The differences between PC-PFC and PS-PFC predictions are similar to those noted for G_1, that is, PS-PFC has a much smoother and less aggressive input prediction, albeit slow, and output prediction, due to the choice of α. Of course, this difference means that the constraint handling for PS-PFC will be far more preferable and less conservative, in general. Conversely, since PC-PFC cancelled out two of their oscillatory open-loop poles, a sudden spike or aggressive damping is expected in the output response.
- For the unstable process G_3, a conventional PFC cannot be used because the divergent predictions will automatically violate constraints so that no feasible choice for u_k will exist. Once again, it is seen that the predictions for PS-PFC are preferable to those from PC-PFC.

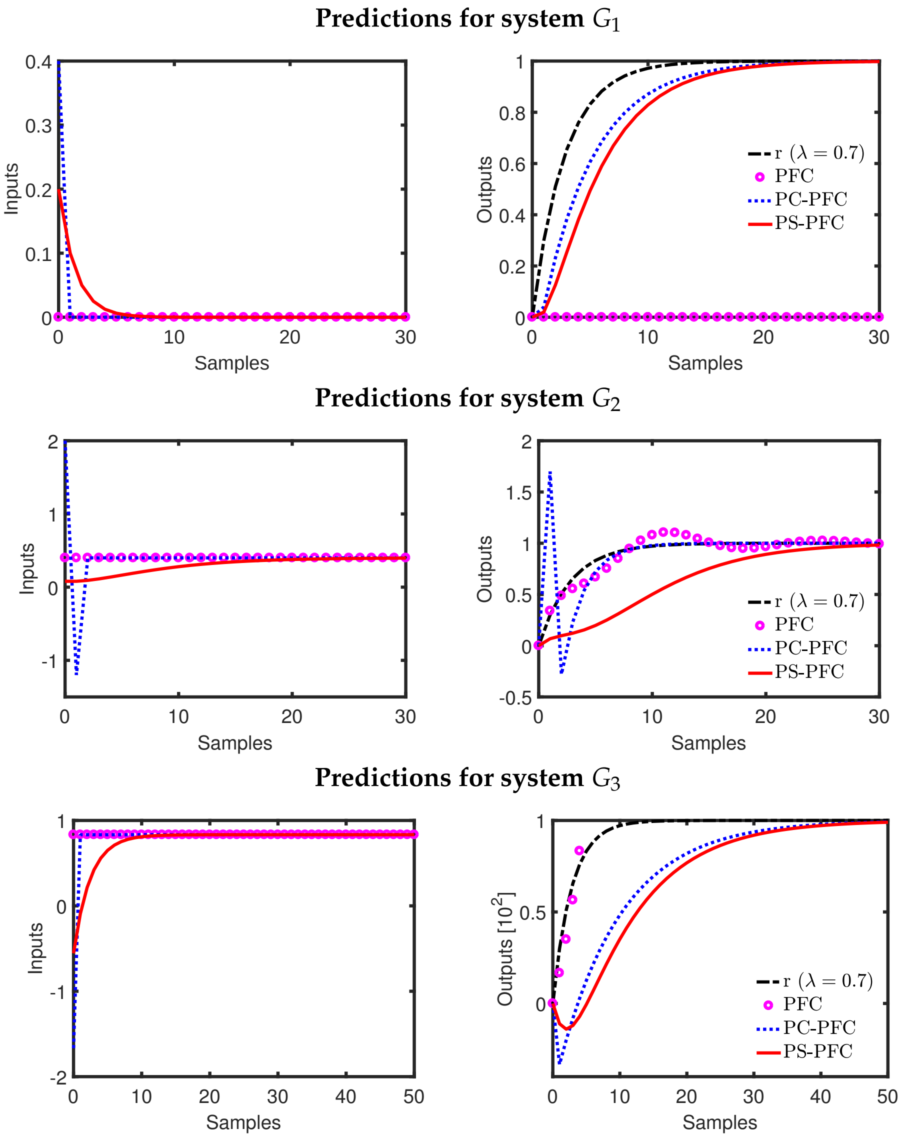


Figure 2. Input and output predictions with PFC, PC-PFC, and PS-PFC for processes G_1, G_2, G_3.

In summary, PS-PFC produces the best prediction behaviour because it ensures convergent predictions which will satisfy constraints with less aggressive input predictions, and thus less conservative constraint handling, than given by PC-PFC/PFC.

4.3. Tuning Efficacy and Closed-Loop Performance: The Unconstrained Case

First we give a brief discussion on PFC tuning for completeness. In general terms, a good practice guidance is to select the coincidence horizon in between 40% and 80% rise of the step input response to the steady-state value [13]: here, G_1 ($4 \leq n \leq 9$), G_2 ($8 \leq n \leq 15$), and G_3 ($11 \leq n \leq 21$). Selecting a smaller horizon will lead to a more aggressive input, while larger horizons reduce the efficacy of λ as a tuning parameter.

In general therefore, the main designer choice is the desired closed-loop pole; for simplicity of illustration, we take the desired pole to be $\lambda = 0.7$. Figure 3 demonstrates the closed-loop performance of PFC, PC-PFC and PS-PFC on the three case study processes with these tunings. It is noted that:

- For all cases, PS-PFC (using a default choice of α) gives the best trade-off between the rate of convergence and the aggressiveness of input activity compared to PFC and PC-PFC. Changes to α could offer a further tuning parameter for varying this trade-off.

- It is notable that PS-PFC gives similar or better output behaviour to PFC/PC-PFC while using a far smoother and less aggressive input trajectory.
- For processes G_2, G_3, the input and output behaviour of PC-PFC is extremely aggressive and would not be implementable in a real application.
- For process G_3, the conventional PFC cannot be stabilised with the given choice of n.

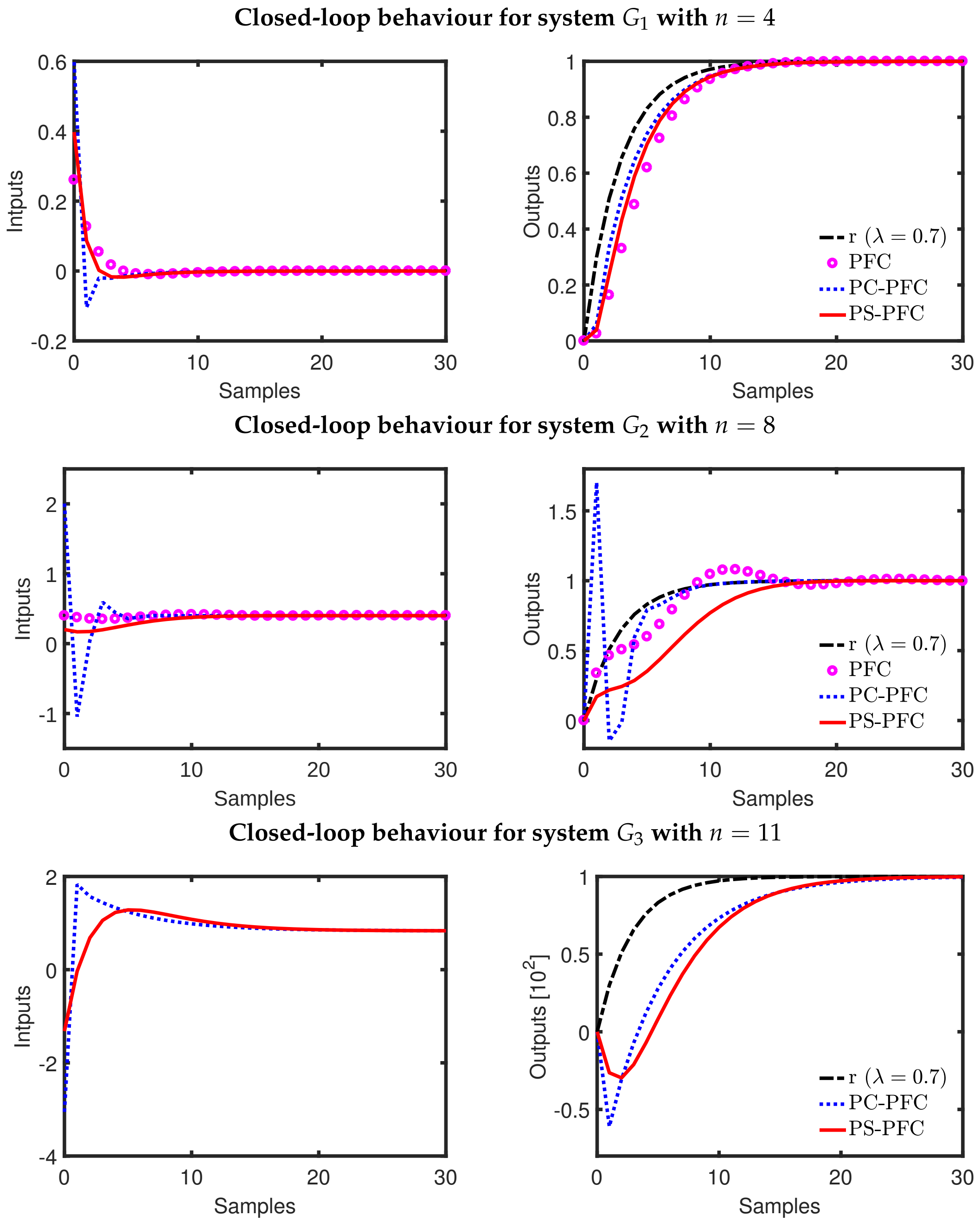


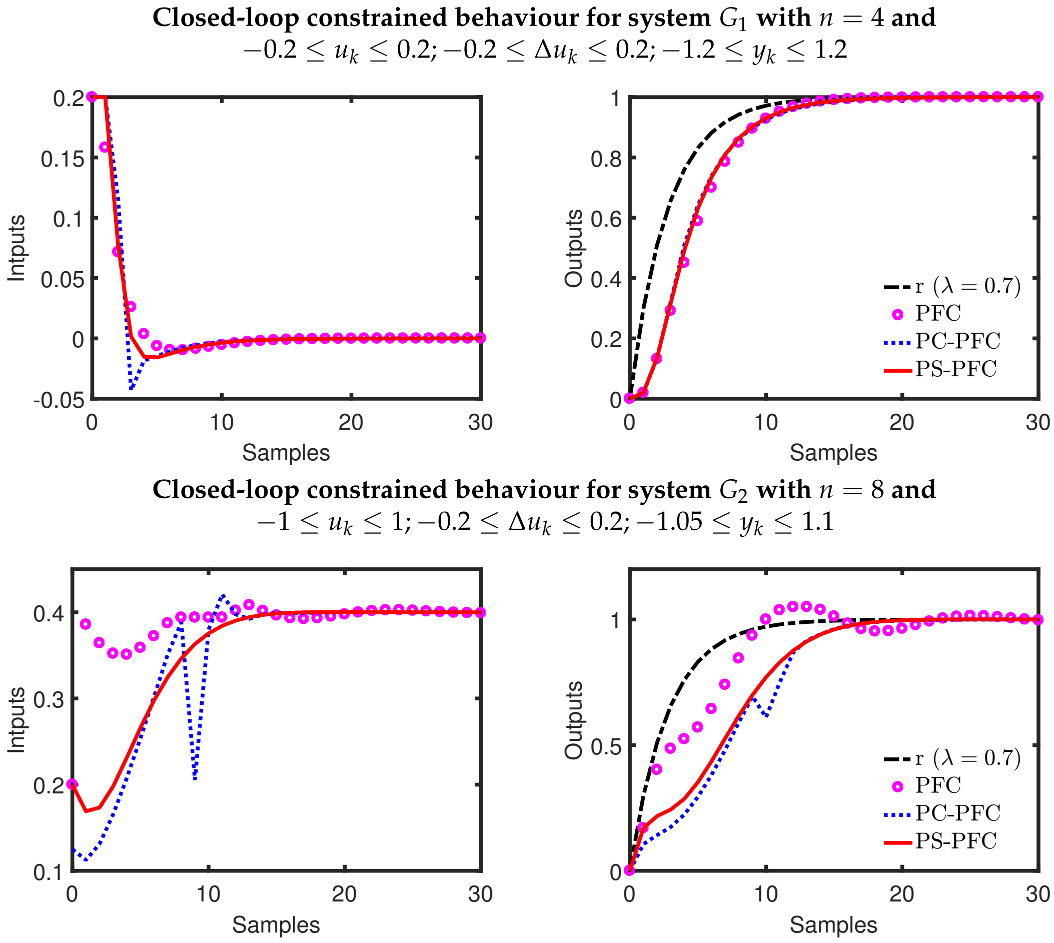
Figure 3. Closed-loop input and output behaviour of PC-PFC and PS-PFC for processes G_1, G_2, G_3.

4.4. Constraint Handling

As noted in Section 4.2, for many dynamics a conventional PFC approach is infeasible or highly conservative because the output predictions inevitably violate constraints beyond a given horizon. Thus, conventional PFC can only be implemented by using short constraint horizons and thus with the loss of any recursive feasibility assurance; if it does work this is luck not good design and thus should be avoided. For the case studies here, conventional PFC could only be used safely with output constraints for G_2, although in that case we would expect some conservatism due to the oscillations in the output predictions.

PC-PFC and PS-PFC pre-stabilise the output predictions and thus can be used safely and with a recursive feasibility assurance. Figure 4 compares the performance of PFC, PC-PFC, and PS-PFC when constraints are added to the process. Several observations can be noted:

- As expected, PS-PFC and PC-PFC satisfy constraints, retain recursive feasibility throughout and converge safely.
- For process G_1, the constrained performance of the controllers are almost the same, but PS-PFC provides a smoother input transition.
- For process G_2 it is clear that handling the under-damping will cause some challenges to any control law, but clearly PS-PFC provides the best responses.
- For process G_3, the inherent dead-beat input predictions deployed by PC-PFC mean the performance is poor and slow to converge whereas PS-PFC performs well.



Closed-loop constrained behaviour for system G_1 with $n = 4$ and
$-0.2 \leq u_k \leq 0.2; -0.2 \leq \Delta u_k \leq 0.2; -1.2 \leq y_k \leq 1.2$

Closed-loop constrained behaviour for system G_2 with $n = 8$ and
$-1 \leq u_k \leq 1; -0.2 \leq \Delta u_k \leq 0.2; -1.05 \leq y_k \leq 1.1$

Figure 4. *Cont.*

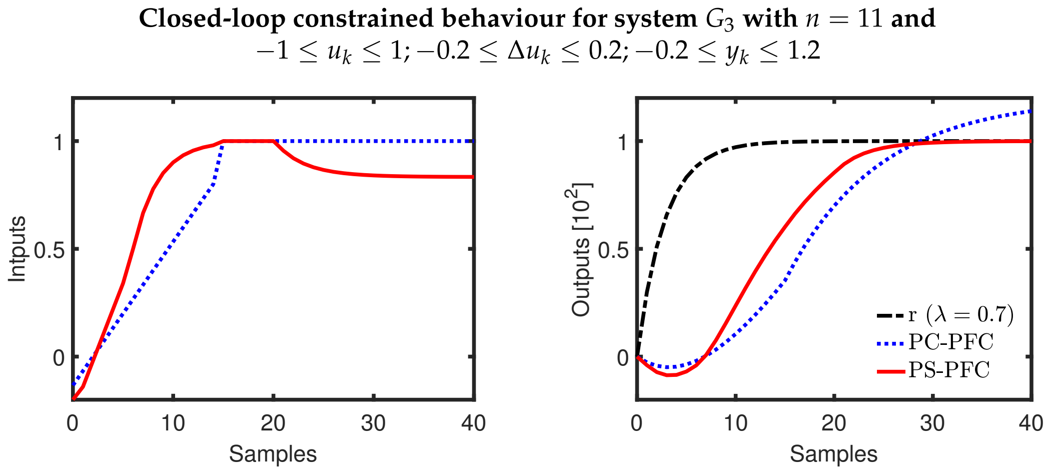


Figure 4. Constrained input and output behaviour of PFC, Pole Cancellation PFC (PC-PFC), and Pole Shaping PFC (PS-PFC) for processes $G1, G_2, G_3$.

4.5. Application of PS-PFC on Laboratory Hardware

This section demonstrates the implementation of PS-PFC on laboratory hardware, that is, a Quanser SRV02 servo-based unit (Quanser, Markham, ON, Canada) (see Figure 5). This device is powered by a Quanser VoltPAQ-X1 amplifier and operates by National Instrument ELVIS II+ (National Instruments, Austin, TX, USA) multifunctional data acquisition via USB connection. The control objective is to track the servo position $\theta(t)$, measured in radians, by manipulating the supplied voltage $V(t)$. This servo will rotate counter-clockwise with positive supplied voltage and vice versa. A second-order model of (33) with an integrator is used to represent the system dynamics (refer to [29] for a more formal derivation) as:

$$\theta(t) = \frac{1.53}{s(0.254s + 1)} V(s) \tag{33}$$

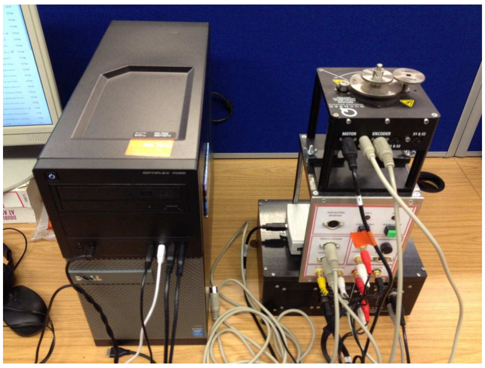


Figure 5. Quanser SRV02 servo based unit.

To implement the proposed control law, the continuous model (33) is discretised with sampling time $0.02s$ to obtain the discrete model of:

$$G_s = \frac{0.0095z^{-2} + 0.0073z^{-1}}{1 - 1.45z^{-1} + 0.45z^{-2}} \tag{34}$$

The plant is set to have a CLTR of $0.2s$ (which is equivalent to $\lambda = 0.89$). Using a similar procedure to that in the previous section, the coincidence horizon is selected at $n = 4$. Figure 6 demonstrates the unconstrained and constrained performances of PS-PFC in tracking the servo position. In the unconstrained case, the controller manages to track the target position with the desired convergence speed. As for the constrained case, all the implied input limits ($-8 \leq u_k \leq 8$), rate limits ($-3 \leq \Delta u_k \leq 3$), and output limits ($-0.8 \leq y_k \leq 0.8$) are satisfied systematically without any conflict.

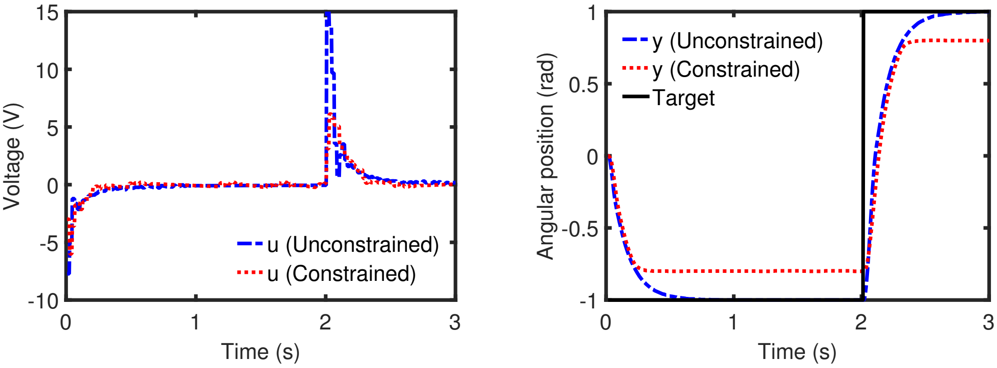


Figure 6. Unconstrained and constrained performance of PS-PFC for process G_s.

5. Conclusions

This paper proposes two potential simple modifications to a conventional PFC algorithm to improve the constraint handling properties for processes with challenging dynamics, such as integrating modes, underdamping, or unstable modes. Both proposals use relatively simple algebra—in effect, the solution of a small number of linear simultaneous equations—to parameterise the future input trajectories which lead to convergent and desirable output behaviours; this is done in terms of a component to deal with the current state (or initial condition) and a free component for control purposes. A core contribution is to show that using the proposed parameterisations allows a simple proof of recursive feasibility so that the constraint handling can be performed more safely and reliably.

A specific novelty of this paper is the proposed PS-PFC algorithm, which gives a pragmatic and simple proposal for deriving input and output prediction pairs which do not require aggressive inputs during transients; the more classical alternative approach of PC-PFC, in general, deploys very aggressive inputs in transients and thus cannot be used in practice. Simulation evidence on a variety of simulation case studies and hardware demonstrates that the proposed PS-PFC algorithm significantly outperforms both a conventional PFC approach and PC-PFC.

Although, as with all predictive control laws, both PC-PFC and PS-PFC are robust to some parameter uncertainty and disturbances, a detailed sensitivity analysis is an important next step. Also, it is of particular interest to compare these approaches with the alternative feedback formulations for PFC [6,15] by way of systematic design, nominal performance, constraint handling, and sensitivity.

Author Contributions: This paper is a collaborative work between both authors. J.A.R. provided initial proposals and accurate communication of the concepts employed in previous MPC and PFC control laws while reviewing the whole project. M.A. developed the code and analysed the concept in various challenging dynamics process within PFC framework.

Funding: This research received no external funding.

Acknowledgments: The first author would like to acknowledge International Islamic University Malaysia and Ministry of Higher Education Malaysia for the scholarship.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Khadir, M.; Ringwood, J. Stability issues for first order predictive functional controllers: Extension to handle higher order internal models. In Proceedings of the International Conference on Computer Systems and Information Technology, Algiers, Algeria, 19–21 July 2005; pp. 174–179.
2. Richalet, J. Industrial applications of model based predictive control. *Automatica* **1993**, *29*, 1251–1274. [CrossRef]
3. Haber, R.; Rossiter, J.A.; Zabet, K. An alternative for PID control: Predictive functional Control—A tutorial. In Proceedings of the 2016 American Control Conference (ACC), Boston, MA, USA, 6–8 July 2016; pp. 6935–6940.
4. Clarke, D.W.; Mohtadi, C. Properties of generalized predictive control. *Automatica* **1989**, *25*, 859–875. [CrossRef]
5. Haber, R.; Bars, R.; Schmitz, U. *Predictive Control in Process Engineering: From the Basics to the Applications*; Wiley-VCH: Weinheim, Germany, 2011.
6. Richalet, J.; O'Donovan, D. *Predictive Functional Control: Principles and Industrial Applications*; Springer: London, UK, 2009.
7. Rossiter, J.A. A priori stability results for PFC. *Int. J. Control* **2016**, *90*, 305–313. [CrossRef]
8. Zabet, K.; Rossiter, J.A.; Haber, R.; Abdullah, M. Pole-placement predictive functional control for under-damped systems with real numbers algebra. *ISA Trans.* **2017**, *71*, 403–414. [CrossRef] [PubMed]
9. Khadir, M.; Ringwood, J. Extension of first order predictive functional controllers to handle higher order internal models. *Int. J. Appl. Math. Comput. Sci.* **2008**, *18*, 229–239. [CrossRef]
10. Rossiter, J.A.; Haber, R.; Zabet, K. Pole-placement predictive functional control for over-damped systems with real poles. *ISA Trans.* **2016**, *61*, 229–239. [CrossRef] [PubMed]
11. Abdullah, M.; Rossiter, J.A. Utilising Laguerre function in predictive functional control to ensure prediction consistency. In Proceedings of the 2016 UKACC 11th International Conference on Control (CONTROL), Belfast, UK, 31 August–2 September 2016 .
12. Abdullah, M.; Rossiter, J.A.; Haber, R. Development of constrained predictive functional control using Laguerre function based prediction. *IFAC-PapersOnLine* **2017**, *50*, 10705–10710. [CrossRef]
13. Rossiter, J.A.; Haber, R. The effect of coincidence horizon on predictive functional control. *Processes* **2015**, *3*, 25–45. [CrossRef]
14. Richalet, J.; Rault, A.; Testud, J.; Papon, J. Model predictive heuristic control: Applications to industrial processes. *Automatica* **1987**, *14*, 413–428. [CrossRef]
15. Zhang, Z.; Rossiter, J.A.; Xie, L.; Su, H. *Predictive Functional Control for Integral Systems*; PSE: Bellevue, WD, USA, 2018. (In Chinese)
16. Rossiter, J.A. Input shaping for pfc: How and why? *J. Control Decis.* **2015**, *3*, 105–118. [CrossRef]
17. Mosca, E.; Zhang, J. Stable redesign of predictive control. *Automatica* **1992**, *28*, 1229–1233. [CrossRef]
18. Rossiter, J.A. Predictive functional control: More than one way to pre stabilise. In Proceedings of the 15th Triennial World Congress, Barcelona, Spain, 21–26 July 2002; pp. 289–294.
19. Rawlings, J.; Muske, K. The stability of constrained receding horizon control. *IEEE Trans. Autom. Control* **1993**, *38*, 1512–1516. [CrossRef]
20. Richalet, J.; O'Donovan, D. Elementary predictive functional control: A tutorial. In Proceedings of the 2011 International Symposium on Advanced Control of Industrial Processes (ADCONIP), Hangzhou, China, 23–26 May 2011; pp. 306–313.
21. Rossiter, J.A. *A First Course in Predictive Control*, 2nd ed.; CRC Press: London, UK, 2018.
22. Nise, N.S. *Control System Engineering*; John Wiley & Sons, Inc.: New York, NY, USA, 2011.
23. Gilbert, E.; Tan, K. Linear systems with state and control constraints: The theory and application of maximal admissible sets. *IEEE Trans. Autom. Control* **1991**, *36*, 1008–1020. [CrossRef]
24. Fiani, P.; Richalet, J. Handling input and state constraints in predictive functional control. In Proceedings of the 30th IEEE Conference on Decision and Control, Brighton, UK, 11–13 December 1991; pp. 985–990.
25. Rossiter, J.A.; Kouvaritakis, B. Numerical robustness and efficiency of generalised predictive control algorithms with guaranted stability. *IEE Proc. D* **1994**, *141*, 154–162.
26. Scokaert, P.O.; Rawlings, J.B. Constrained linear quadratic regulation. *IEEE Trans. Autom. Control* **1998**, *43*, 1163–1169. [CrossRef]

27. Zhuo, W.; Shichao, W.; Yanyan J. Simulation of control of water level in boiler drum. In Proceedings of the World Automation Congress 2012, Puerto Vallarta, Mexico, 24–28 June 2012.
28. Kendi, T.A.; Doyle, F.J., III. Nonlinear control of a fluidized bed reactor using approximate feedback linearization. *Ind. Eng. Chem. Res.* **1996**, *35*, 746–757. [CrossRef]
29. Apkarian, J.; Lévis, M.; Gurocak, H. *Instructor Workbook: SVR02 Based Unit Experiment for LabVIEW Users*; Quanser Inc.: Markham, ON, Canada, 2012.