Article:

Rossiter, J. orcid.org/0000-0002-1336-0633 and Aftab, M.S. (2022) Recent developments in tuning methods for predictive functional control. Processes, 10 (7). 1398. ISSN 2227-9717

https://doi.org/10.3390/pr10071398

processes

Article

Recent Developments in Tuning Methods for Predictive Functional Control

John Anthony Rossiter * **and Muhammad Saleheen Aftab**

Department of Automatic Control and System Engineering, University of Sheffield, Mappin Street, Sheffield S1 3JD, UK; msaftab1@sheffield.ac.uk
* Correspondence: j.a.rossiter@sheffield.ac.uk

Abstract: Predictive functional control (PFC) is a popular alternative to PID because it exploits model information better and enables systematic constraint handling while also being cheap and computationally efficient. A recent overview paper reviewed some recent proposals for improving the tuning efficacy. This paper extends and develops upon that review paper by introducing some exciting new proposals for how to making tuning more intuitive and, thus, easier for unskilled operators. Moreover, there are early indications that these proposals are easily modified for use in nonlinear cases while maintaining a very low cost and a simple and fast online computation.

Keywords: predictive control; challenging dynamics; tuning; stability properties

check for updates

Citation: Rossiter, J.A.; Aftab, M.S. Recent Developments in Tuning Methods for Predictive Functional Control. *Processes* **2022**, *10*, 1398. https://doi.org/10.3390/pr10071398

Academic Editor: Jie Zhang

Received: 8 June 2022
Accepted: 8 July 2022
Published: 18 July 2022

1. Introduction

Predictive functional control (PFC) is a popular control technique—especially for single-input single-output (SISO) applications [1–7]—because it typically outperforms PID while being of similar cost and implementation complexity. PFC exploits model information explicitly and, thus, in principle, can deal with more challenging dynamics than PID can. Moreover, as it is based on prediction, it is able to embed systematic constraint handling. Finally, it is designed to have just a single degree of freedom and, thus, to be computationally efficient and relatively easy and cheap to code [8,9]. Thus, it is a competitor with traditional approaches, such as PID, but not with more expensive model predictive control (MPC) variants, such as DMC [10,11].

Nevertheless, as highlighted in the recent review paper [1], simplicity and computational efficiency come at a cost, that is, the tuning of PFC is often not as straightforward or effective as one would like [12]; thus, recent research efforts, such as the development of closed-loop PFC [13,14], Laguerre PFC [15], PFC with an improved feedforward component [16], and designs based on model decomposition [17,18], have focussed improving tuning efficacy. This paper will not repeat the review of those methods (see [1] for details), but will rather develop and extend them. We note that a priori stability guarantees [18,19] are rarely considered with PFC, as they are considered unrealistic with such low computational complexity. If one wants greater assurances of robustness and stronger guarantees, one will likely need to pay much more and use a more advanced MPC approach.

This paper makes a new proposal that is based on changing the fundamental concept of the coincidence point. Historically, PFC algorithms have deployed a coincidence point as the main defining principle for the control law, alongside the selection of a desired dominant closed-loop pole. In simple terms, the prediction is forced to coincide with a pre-determined value/dynamic n_y steps ahead, with n_y being the coincidence horizon. All of the recently developed proposals have used the same defining principle and have focussed on how the input degrees of freedom are parameterised; changing the input predictions has a significant impact on the efficacy of the overall design.Indeed, this latter point will be preserved in the proposal of this paper, as it is well known in the MPC field [20,21] that consistency between open-loop predictions and closed-loop behaviour is a solid foundation for sensible MPC design. Recent work has shown that there are two input parameterisations that, in general, seem to be more effective in ensuring this consistency: (i) Laguerre-based parameterisations [15] and (ii) parameterisations that exploit an inner feedback loop [13,14,22]. Consequently, this paper will mainly focus on those two input parameterisations.

One difficulty with using the coincidence point alongside a desired dynamic was exposed fairly recently [16], where it was shown that this definition often embedded lag into the closed-loop responses; indeed, it did so for all systems except those with dynamics that were close to first order. This was because of inconsistency from one sample to the next, which was brought about by the implicit requirement for the system to behave like a first-order system throughout its response, not just asymptotically. The proposal [16] was to modify, but not remove, the coincidence point's definition, and indeed, such a modification was shown to be effective, although the resulting algorithm's definition was not so clean to present; computationally, the change in loading was minimal and, thus, irrelevant. Indeed, the proposal was reduced to implementing any disturbance and target information into the coincidence point's definition through a pre-defined feedforward compensator.

This paper takes a different track entirely and removes the coincidence point/first-order dynamic pairing altogether. The plan is still to base the control design around a sensible and—critically for adoption—simple and intuitive benchmark response, but here, we use a natural behaviour of the system as opposed to an "ideal first-order response", as the latter is often unrealistic. A core question is that of selecting this benchmark response in a computationally simple and intuitive fashion, and then, secondly, building on this to optimise the closed-loop behaviour in some intuitive sense. What is particularly exciting about the new proposal given here is that modifications for dealing with the nonlinear case seem to be elementary, and thus, this provides an effective tool for nonlinear control design. Specifically, this paper will focus on two innovations of PFC.

1. How do we make tuning as intuitive and, thus, as simple as possible, thus empowering lower-skilled operators?
2. How do we define meaningful and simple PFC control laws using common input parameterisations?

Section 2 introduces the core concepts of PFC and a concise summary of possible variations and constraint-handling approaches. Section 3 introduces the newly proposed approach, which builds on some preliminary ideas in [23] and shows how it can be modified to deal with different input parameterisation choices. This is followed by Section 4, which gives some case study implementations. The paper finishes with a conclusion and a description of possible areas of future study.

2. Overview of Traditional PFC

Before introducing the proposed innovations, it is important for the reader to be familiar with the basic concepts, notation, and recent algorithms of PFC [8,9,12,24]. These are summarised in the following subsections.

2.1. Modelling

It is common, but not necessary, to use transfer-function-based models; for example:

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

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.

Remark 1. *Parameter uncertainty and disturbances are typically accounted for by using a simple disturbance model, as shown in Figure 1. The disturbance signal is defined as* $d_k = y_{p,k} - y_{m,k}$, *where y_p is the output measurement of the actual process G_p, whereas y_m is determined from the system model G_m (essentially a parallel computer simulation). The signal d_k is used to correct predictions based on the system model for any bias. Hereafter, we drop the subscripts m, p except where they are needed for clarity.*

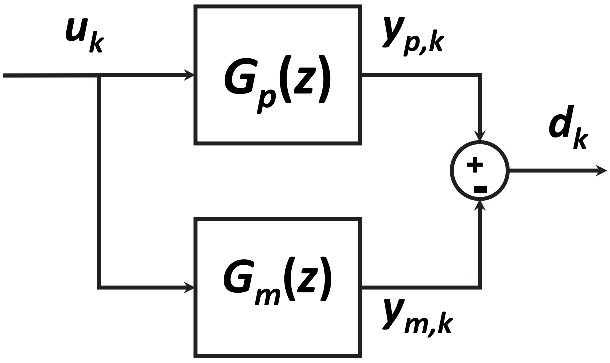


Figure 1. Independent model structure and disturbance estimate.

2.2. Prediction

Predictions underpin PFC. These are based on a system model (1) and are corrected for bias [10,20]. This is routine in the literature, so we will avoid fine details here and summarise the result for an n-step-ahead output prediction in terms of future inputs $u_{k+i},\ i \geq 0$:

$$E[y_{p,k+n|k}] = H_n \underset{\rightarrow}{u_k} + P_n \underset{\leftarrow}{u_k} + Q_n \underset{\leftarrow}{y_{m,k}} + d_k \tag{2}$$

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

It is important to note that the outputs in this prediction are based on the model G_m and its states, not the process, and the term d_k is used to correct for bias. The row vectors H_n, P_n, and Q_n depend on the model parameters [20] and can be computed very efficiently.

2.3. Conventional PFC Control Law

Historical PFC design is intuitive in that one uses a target first-order dynamic as an ideal closed-loop response [24]. In essence, the aim is to cause the closed-loop system to have the associated time constant (integral action and, thus, zero offset are implicit). The link between the target pole λ and time constant T_s is $\lambda = e^{\frac{-T}{T_s}}$ for a sample time T.

A conventional algorithm attempted to imply the required closed-loop dynamic by forcing the open-loop predictions to meet the following coincidence condition:

$$E[y_{p,k+n|k}] = (1 - \lambda^n) r + \lambda^n y_{p,k} \tag{3}$$

where r is the set point and λ is the desired closed-loop pole. This equality is easy to solve with predictions (2), and thus, the control law is defined by selecting u_k with $\Delta u_{k+i} = 0,\ i > 0$ such that:

$$\underbrace{H_n \mathbf{e}_1}_{h_n} u_k + P_n \underset{\leftarrow}{u_k} + Q_n \underset{\leftarrow}{y_k} + d_k = (1 - \lambda^n) r + \lambda^n y_p(k) \tag{4}$$

Thus, after rearrangement, the control law (3) is implemented by solving:

$$\Delta u_k = \frac{1}{h_n} \left[(1 - \lambda^n) r + \lambda^n y_k - Q_n \underset{\leftarrow}{y_k} - P_n \underset{\leftarrow}{u_k} \right] \tag{5}$$

2.3.1. Efficacy of Conventional PFC

The control law (5) works well when the open-loop system has a behaviour that is close to a monotonic step response, such as in first-order systems; for these cases, the tuning parameter λ is then reasonably effective. However, for systems with more complex dynamics or significant lag in the initial response, the tuning parameter is much less effective [12,15,16]. The developments summarised in this paper are focussed towards the latter cases.

2.3.2. Constraint Handling

Constraint handling is not a main discussion point in this paper, so it is briefly included here for completeness. The authors recommend approaches similar to those in mainstream MPC [15,20,25,26]. Numerous inequalities representing constraints for the future input/output predictions are stacked into a single vector/matrix inequality expression as follows.

$$\underline{\Delta u} \leq \Delta u_k \leq \overline{\Delta u}; \quad \underline{u} \leq u_k \leq \overline{u}; \quad \underline{y} \leq y_k \leq \overline{y}, \ \forall k \tag{6}$$

$$C\Delta u_k \leq \mathbf{f}_k \tag{7}$$

$$C = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \\ H\mathbf{e_1} \\ -H\mathbf{e_1} \end{bmatrix}; \quad \mathbf{f}_k = \begin{bmatrix} \overline{u} - u_{k-1} \\ -\underline{u} + u_{k-1} \\ \overline{\Delta u} \\ -\underline{\Delta u} \\ L\overline{y} - P\underleftarrow{\Delta u_k} - Q\underleftarrow{y_k} - Ld_k \\ -L\underline{y} + P\underleftarrow{\Delta u_k} + Q\underleftarrow{y_k} + Ld_k \end{bmatrix}; \quad L = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

where $\mathbf{f}_k$ depends on past data in $\underleftarrow{\Delta u_k}, \underleftarrow{y_k}$ and on the limits. Given that there is a single degree of freedom, a simple *for loop* within the code can find the choice of u_k closest to (5) to ensure that (7) is satisfied very efficiently and, indeed, to ensure recursive feasibility properties.

Remark 2. *Small modifications to the algebra in* (7) *are needed for different PFC algorithms, but the concepts are the same, so those details are omitted, as they are well understood in the literature.*

2.4. Recently Proposed Variations of PFC

A number of proposals [1,16] have appeared in the literature to improve the tuning efficacy of PFC because of the poor correspondence between the chosen λ and the resulting closed-loop pole, thus undermining a core selling point of PFC, that is, the ease of tuning. This issue was exacerbated for systems with more challenging dynamics, such as unstable open-loop poles, significant underdamping, or integrating dynamics; hence, modifications to the basic PFC algorithm are essential. For example, there are several obvious weaknesses in conventional PFC that have been exposed in the recent literature:

1. The definition of the coincidence point [16] can induce lag and inconsistency in decision making.
2. The parameterisation of the future inputs is too simplistic for many cases and is inconsistent with the desirable closed-loop input trajectory [1].
3. The tuning parameter λ is not as effective as one would like [12].

The recent review [1] concluded that, of the various options that had been considered, input shaping [26–29] would not be pursued for the time being. Apart from the algebra being somewhat awkward, these approaches can be quite sensitive to parameter uncertainty, which is difficult—or, rather, systematic guidance for deciding the precise shaping needed does not yet exist, and moreover, they do not deal with the first and third fundamental weaknesses listed above. Similarly, pole placement approaches [18,30], while interesting, required the selection of a number of poles for which, again, no systematic guidance yet exists. Moreover, the constraint-handling algebra was much messier than with other methods.

In summary, two approaches seem to have positive properties and worth pursuing in the future [1]: (i) Laguerre PFC (LPFC) [15] and (ii) closed-loop PFC (CLPFC) [13,14]. These both tackle the weakness in the parameterisation of the future input trajectory, but in different ways. In essence, the aim is to ensure that the open-loop prediction is as consistent as possible [20,31] with the closed-loop input, as this ensures consistency of decision making from one sample to the next, and moreover, it helps with recursive feasibility assurances. In both cases, it has been shown that these approaches tackle the second and third weaknesses in the list above. LPFC is slightly simpler and would be favoured where effective, but CLPFC is generally required and more effective for systems with challenging open-loop dynamics, and it is essential for unstable open-loop dynamics [13,22].

Hence, as LPFC and CLPFC will underpin the proposals in this paper, the two control laws will be concisely presented.

2.4.1. Laguerre PFC (LPFC)

LPFC [15,26] assumes that the input predictions converge to the desired steady state following a simple exponential curve; the dynamics of that curve are matched with the target closed-loop pole λ for simplicity. Hence, we parameterise the predicted future inputs as follows:

$$\underset{\rightarrow}{u}(z) = u_{ss} + \sum_i L_i(z)\eta_i; \quad L_1(z) = 1 + \lambda z^{-1} + \lambda^2 z^{-2} + \cdots \tag{8}$$

where u_{ss} is the expected steady-state input that will remove the offset (for a constant asymptotic target).

The LPFC is defined by using input prediction (8) in the control law definition (3). This requires some modification of the prediction Equations (2) and (4) in order to deal with the change in the input predictions, but such details are trivial and available in the references, so they are not included here.

Nevertheless, a core reflection on LPFC is that, despite it being effective for some systems and an improvement on the conventional PFC approach, there is still poorer consistency between the closed-loop poles and the target pole λ than is satisfactory. This is largely, but not solely, due to the issues introduced by the definition of the coincidence point or control law (3), which underpins all PFC approaches to date [16].

2.4.2. Closed-Loop or Pre-Stabilised PFC (CLPFC)

It was recognised in the mainstream MPC literature from the early 1990s that one effective way of reparameterising the input predictions in order to be more consistent with the desirable behaviour was to deploy an inner feedback loop. Indeed, this was also recognised in some PFC proposals [9,24], although those implementations were clumsy and contained other inconsistencies. More recently, it was suggested that researchers should investigate this more systematically in a PFC context [13,14,32]. The basic idea is to form an internal feedback loop, such as in Figure 2, and then to form an outer loop based on PFC to determine the loop input signal v_k. This approach is particularly useful for handling systems with difficult open-loop dynamics, as the inner loop can soften those dynamics, meaning that the outer loop, namely, PFC, has less work to do and is working with predictions that are already closer to something desirable.

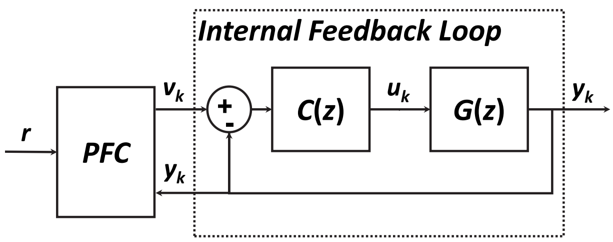


Figure 2. Cascade structure with PFC.

It should be emphasised that the feedback designs for the inner loop [33] are deliberately simple classical designs, so the overall design process is not over complicated. The primary focus is on moving the poles to more desirable positions, that is, reducing damping, removing instability, and so forth. The PFC loop works on the model for the inner loop, but is otherwise defined as a standard with input v_k and output y_k. Once the outer PFC loop is added, there are some nuances with prediction and constraint handling, as discussed in the references, but, in truth, this detail makes little substantive difference in the implementation and coding complexity, and the tuning is much improved.

Nevertheless, as with LPFC, it is noted that CLPFC also suffers from the same inherent weakness associated with the coincidence point's definition [16]. Consequently, this paper seeks to explore whether an alternative approach would improve the overall algorithm and tuning.

2.5. Summary

The consensus of the recent literature is that LPFC and CLPFC are simple developments of the conventional PFC that, in general terms, should be favoured. However, these do not deal with the fundamental problem of lag caused by the definition of the coincidence point. A recent work [16] demonstrated that a change in the way that reference and disturbance information is embedded can be extended to the LPFC and CLPFC algorithms. However, this paper proposes a conceptually different approach, which is the removal or modification of the coincidence point's definition altogether.

3. Using Benchmark Behaviour to Define a New PFC Approach

For obvious reasons, the original PFC approach used a first-order response (possibly with delay) as a benchmark behaviour. However, as is evident from all of the recent work in the literature, such a benchmark may not be suitable for many systems and leads to unexpected behaviours, such as lag and inconsistent decision making.

An alternative approach is to begin from a behaviour that we can guarantee is sensible and achievable; at the very least, this would ensure that consistent and sensible decision making is followed. However, in line with the general aim of PFC, which is computational and conceptual simplicity, this behaviour should be self-evident and not manufactured. Thus, an obvious choice is *open-loop* behaviour.

- For many benign systems, the open-loop dynamics are almost acceptable, and thus, feedback is mainly needed for handling uncertainty and relatively small improvements in the speed of response and damping.
- For systems with less satisfactory open-loop dynamics, a simple classical feedback loop, as in Section 2.4.2, will deliver realistic and sensible dynamics, albeit with the potential for some improvement.

This paper will focus on the speed of response, so an assumption is made that the open-loop dynamics are too slow and faster dynamics are desirable, for example, to increase productivity. The following subsections will introduce three alternative control laws of increasing complexity and cost to deal with scenarios of increasing challenges, but all deploy the basic principle that open-loop behaviour is a realistic benchmark behaviour. A core component for ensuring comparability with PID in terms of implementation costs and coding is the requirement of having just a single degree of freedom.

3.1. A PFC Law Delivering Open-Loop Dynamics: Steady-State PFC (SSPFC)

The very simplest PFC law of all focuses on removing steady-state offset, but does not concern itself with the closed-loop poles. Such a control law is easy to define [20], as, in essence, it reduces to deploying an estimator for the required steady-state input u_{ss}. For example, using the predictions of (2), one can deduce that, for a simple transfer function model $G_m = b(z)/a(z)$ and $u_k = u_{ss}$, $\forall k$:

$$\lim_{k\to\infty} y_{m,k} = \frac{b(1)}{a(1)} u_{ss} = G(1)u_{ss}; \tag{9}$$

Noting the required bias correction, we can define an SSPFC law as:

$$u_k = u_{ss} = \frac{a(1)}{b(1)}[R - d_k]; \quad d_k = y_{p,k} - y_{m,k} \tag{10}$$

The core point of this section is that the SSPFC will be taken to define a reasonable benchmark behaviour. It is reasonable and achievable because we know that it is simply a step response. The proposal hereafter is to use this as a benchmark, as opposed to the first-order behaviour implicit in (3). The coincidence point's definition will be modified, as explained in the following subsection.

Benchmark behaviour: Using predictions (2) and the assumption that $u_k = u_{ss}$, $\forall k$, the associated n-step-ahead predictions are given as:

$$y_{k+n|k} = H_n L u_{ss} + P_n \underleftarrow{u_k} + Q_n \underleftarrow{y_k} + d_k; \quad L = [1, \cdots, 1]^T \tag{11}$$

3.2. Changing the Speed of Response: Speed-Up SSPFC

In the event of no uncertainty, the control law (10) will give a closed-loop input that is a step function. Clearly, there may be cases where the operator desires a slower change in the input or a faster initial acceleration of the output. A suitable input trajectory that delivers this variability can be defined as in (8), that is, in exactly the same way as with LPFC. There are two degrees of freedom: (i) the implied pole λ and (ii) the initial deviation $u_k - u_{ss} = \eta$. As discussed earlier in Section 2.4.2, it can be assumed that λ is defined automatically based on other criteria, and thus, just one degree of freedom remains to be determined.

Before we move on to defining the associated PFC law, first, we need the output predictions associated with (8). These are given as:

$$y_{k+n|k} = H_n[L u_{ss} + M\eta_k] + P_n \underleftarrow{u_k} + Q_n \underleftarrow{y_k} + d_k; \quad M = [1, \lambda, \cdots, \lambda^{n-1}]^T \tag{12}$$

The key difference with respect to the proposed Speed-Up SSPFC is that the selection of η is based on a different criterion from that of CPFC (which used (3)). As noted above, the benchmark behaviour is defined by using (11), whereas the actual predictions are given in (12). We compare the implied tracking errors (e_{bench} and e_{SPFC}, respectively) of these two predictions n-steps ahead. Note, as discussed many times [12], that n needs to be large enough to avoid issues linked to transient behaviour, but not too large:

$$e_{bench} = R - (H_n L u_{ss} + P_n \underleftarrow{u_k} + Q_n \underleftarrow{y_k} + d_k) \tag{13}$$

$$e_{SPFC} = R - (H_n[L u_{ss} + M\eta_k] + P_n \underleftarrow{u_k} + Q_n \underleftarrow{y_k} + d_k) \tag{14}$$

In order to get faster convergence to the steady state, it would be reasonable to enforce (equivalent to the coincidence condition) that:

$$\gamma e_{SPFC} = e_{bench}; \quad \gamma > 1 \tag{15}$$

The parameter γ, in effect, enables a speed-up in the convergence rate, which is related to the ratio of the errors.

Hence, Speed-Up PFC is defined by solving the equality (13) for η, where the user needs to define γ, λ, and n. This reduces to:

$$\gamma H_n M \eta_k = (\gamma - 1)(R - H_n L u_{ss} + P_n \underleftarrow{u_k} + Q_n \underleftarrow{y_k} + d_k) \tag{16}$$

The control input is given as $u_k = u_{ss} + \eta_k$.

Remark 3. *It would be reasonable for γ and λ of the conventional PFC to be related. So, if the dominant open-loop dynamics in the benchmark behaviour have a pole ρ, then the corresponding speed-up is consistent with $\lambda^n = \rho^n/\gamma$. However, it takes very few case studies to realise that the real scenario is somewhat more nuanced, and an alternative choice of γ is needed.*

A more systematic choice of γ is given in the following, which mainly applies to linear systems. In simple terms, this is conceptually similar to the design of proportional control for a PI compensator. The size of the proportion is directly related to the speed of response, as it dictates the magnitude of the initial input action for an error. Hence, we select γ to ensure that the initial reaction to an error has the desired ratio; as this is a linear system, for simplicity, zero initial conditions can be used to derive a suitable value.

1. Define the desired speed-up factor as β.
2. For initial conditions of zero, determine the inputs u_{bench}, u_{SPFC} associated, respectively, with e_{bench} and e_{SPFC} and determine the dependence of this ratio on γ.
3. Select γ so that:
$$\frac{u_{SPFC}}{u_{bench}} = \beta \tag{17}$$
Hence:
$$\frac{u_{SPFC}}{u_{bench}} = \frac{(\gamma-1)(R - H_n L u_{ss})}{\gamma H_n M u_{ss}} + 1 = \beta \tag{18}$$
4. Noting that (nominal case) $u_{ss} = R/G(1)$, this reduces to;
$$\frac{u_{SPFC}}{u_{bench}} = \frac{(\gamma-1)(G(1) - H_n L)}{\gamma H_n M} + 1 = \beta \tag{19}$$
which gives a simple and unique choice for γ.

The efficacy of control law (16)–(19) when taken as a pair, especially when it comes to the tuning of a controller by an unskilled operator, will be demonstrated in the section on numerical examples. Nevertheless, it must be emphasised that this approach relies on the assumption that the benchmark open-loop behaviour has largely satisfactory dynamics. The next section considers what to do when that is not the case.

Remark 4. *The control law (16) is equivalent to the benchmark (10) with $\beta = 1$; hence, both are denoted as SSPFC.*

3.3. Speed-Up PFC with Feedback (CLSSPFC)

There will be many cases where the behaviour of SSPFC as in (16) will be poor—for example, when the open-loop dynamics are poor (e.g., under-damping and instability). Indeed, such issues have been a core motivation for much of the recent work on the tuning of PFC [1].

In such cases, one would need a more sensible benchmark, such as that obtained with an inner prestabilising loop (see Figure 2 and Section 2.4.2). The definition of the associated PFC is exactly analogous to that of SPFC, but just with an alternative definition of the benchmark and speed-up predictions, which are now based on the input signal v_k to the inner loop.

Speed-Up SSPFC with Feedback is defined as follows:

1. Define the transfer function G_{inner} of the inner loop.
2. Define the associated inner-loop predictions and speed-up predictions as:

$$e_{bench,inner} = R - H_{n,inner} L v_{ss} + P_{n,inner} \underset{\leftarrow}{v_k} + Q_{n,inner} \underset{\leftarrow}{y_k} + d_k \tag{20}$$

$$e_{SFPFC,inner} = R - H_{n,inner}[L v_{ss} + M\eta] + P_{n,inner} \underset{\leftarrow}{v_k} + Q_{n,inner} \underset{\leftarrow}{y_k} + d_k \tag{21}$$

3. Solve the following identity to solve for η.

$$\gamma e_{SFPFC,inner} = e_{bench,inner}; \quad \gamma > 1 \tag{22}$$

4. Define the inner-loop input as: $v_k = v_{ss} + \eta$.

As discussed in the previous subsection, the choice of γ will be derived from the equivalent of (19).

Remark 5. *The speed-up parameter β is akin to a multiplying factor that dictates the convergence speed of the closed-loop response. For example, a value selected between 0 and 1 would cause a slow-down, whereas a value greater than 1 would result in faster behaviour compared to that of the benchmark with $\beta = 1$. In many cases, a speed-up up to 2–3 is generally sufficient; beyond that, one may expect severe input aggression, which is unsuitable for practical implementations.*

3.4. Summary of Proposals

This section outlined three alternative control laws of increasing complexity to deal with systems of varying challenge/open-loop dynamics. These are based on a replacement for the definition of the coincidence point; in essence, a different benchmark behaviour is used, which is more strongly related to the actual system dynamics and, thus, more realistic. Instead of choosing a desired pole/time constant, the user defines the speed-up constant β, that is, how much faster (or indeed slower) than the benchmark behaviour they would like to be; thus, this retains the simple and intuitive nature of conventional PFC.

4. Numerical Illustrations

This section provides some numerical illustrations using linear case studies to demonstrate the efficacy and simplicity of the proposed speed-up approach.

4.1. Case Studies

For consistency with the earlier paper [1], this paper will use some case studies with different and challenging dynamics that merit the use of the closed-loop formulations, but we also include some more simple examples for which the simple speed-up PFC is appropriate.

4.1.1. Examples with Relatively Straightforward Dynamics

These examples can be effectively controlled with conventional methods; thus, the main contribution here is the consideration of the ease and transparency of tuning.

Over-damped second- and third-order systems: Models with relatively benign dynamics are useful for considering the basic functionality of the proposed controllers, that is, (10) and (16).

$$B_1 = \frac{-0.1z^{-1} + 0.4z^{-2}}{(1 - 0.5z^{-1})(1 - 0.9z^{-1})} \tag{23}$$

$$B_2 = 10^{-3} \frac{0.1665z^{-1} + 0.0157z^{-2} - 0.1505z^{-3}}{(1 - 0.98z^{-1})(1 - 0.967z^{-1})(1 - 0.951z^{-1})} \tag{24}$$

Non-minimum phase characteristics: While models B_1, B_2 have some lag and even a small non-minimum phase characteristic, it is worth considering a system with a much more aggressive non-minimum phase behaviour alongside a near-integral mode.

$$B_3 = \frac{-6.69z^{-1} + 7.856z^{-2} + 2.392z^{-3} + 0.001445z^{-4}}{[1 - 1.233z^{-1} + 0.5356z^{-2} - 0.005799z^{-3} + 3.249 * 10^{-11}z^{-4}} \tag{25}$$

4.1.2. Examples with Challenging Dynamics

These examples are not amenable to open-loop PFC formulations [1] and, thus, require the algorithm from Section 3.3. A simple nominal feedback law $C(z)$, as required for CLSSPFC (and Figure 2), is included in the model parameter definitions for convenience.

They are based on the behaviours of real systems—for example:

1. Boiler-level systems [34] often include an integrator. The basic SSPFC is not defined for systems with zero steady-state input, that is, $u_{ss} = 1$.

$$G_1 = \frac{0.01z^{-1} + 0.04z^{-2}}{(1 - 0.8z^{-1})(1 - z^{-1})}; \quad C(z) = 0.461 \tag{26}$$

2. Autonomous underwater vehicles [35] typically include oscillatory open-loop dynamics.

$$G_2 = \frac{0.1885z^{-1} - 0.1827z^{-2}}{1 - 1.227z^{-1} + 0.956z^{-2}}; \quad C(z) = -1.88 \tag{27}$$

3. Several industrial systems have unstable dynamics—for example, the temperature of a fluidised bed reactor [36].

$$G_3 = \frac{2.102z^{-1} + 0.4011^{-2}}{(1 - 1.465z^{-1} + 0.0585z^{-2}}; \quad C(z) = \frac{0.303 - 0.0123z^{-1}}{1 - 0.0853z^{-1}} \tag{28}$$

In the following analysis, the closed-loop transfer functions of the inner loop from using $G(z), C(z)$ in the loop structure of Figure 2 will be used with CLPFC and CLSSPFC.

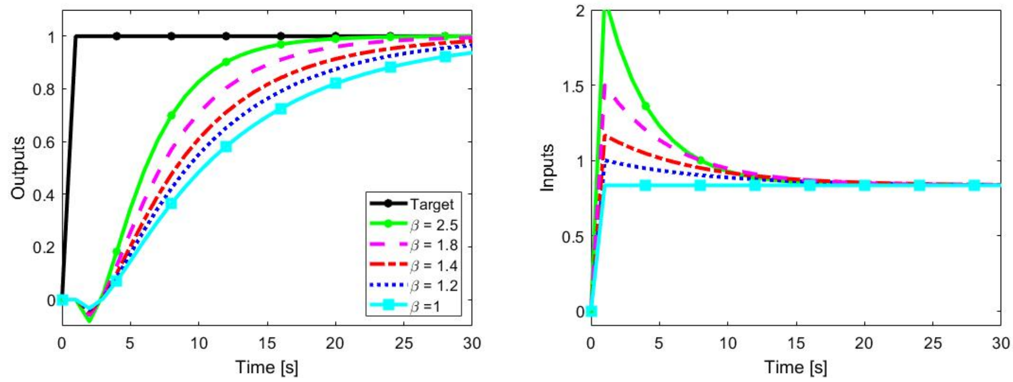
4.2. Tuning Efficacy with the Basic SSPFC Controller

Again, the illustrations will be divided into the easier examples of Section 4.1.1 and harder examples of Section 4.1.2. In both cases, the emphasis is on showing the efficacy of the proposed speed-up approach for controller tuning.

A number of figures are provided in this section to show the closed-loop performance for SSPFC (16) with various choices of β, as well as to contrast this with conventional PFC. PID is tuned using MATLAB auto-tuners to ensure some consistency. Note that, in these examples, the numerical value of n was chosen as per the conjecture presented in [12], i.e., within the time window when the benchmark step response with $\beta = 1$ increased from 40% to 80% of the implied steady state with a significant gradient.

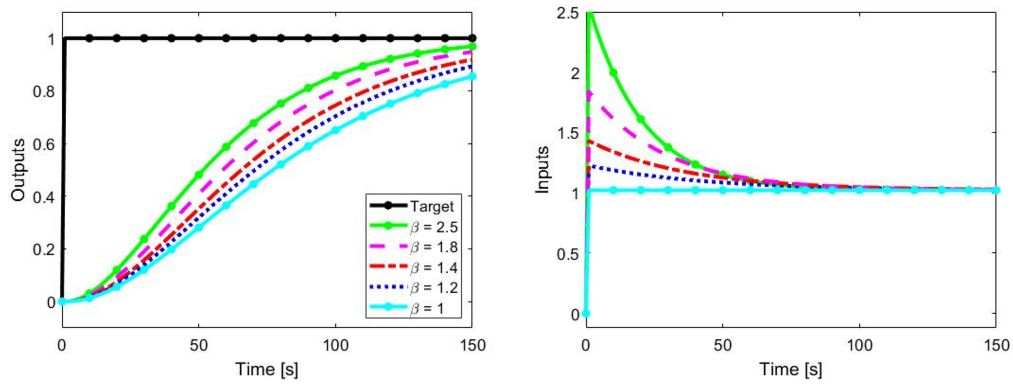
1. Figures 3–5 show the input and output responses for SSPFC with various choices of β. It is clear that this gives a simple and effective mechanism for varying the closed-loop speed of response with the corresponding changes in input activity.
2. Figure 6 compares the tuning for $\beta = 1.8$ with the corresponding efficacy of more conventional approaches. It is clear that, apart from B_2, which is a simple over-damped system, the conventional approaches have, by comparison, significant lag and, thus, are not as effective at achieving a specified behaviour.

In summary, the proposed SSPFC provides much more transparent tuning, which would be straightforward for a non-expert user to deploy.



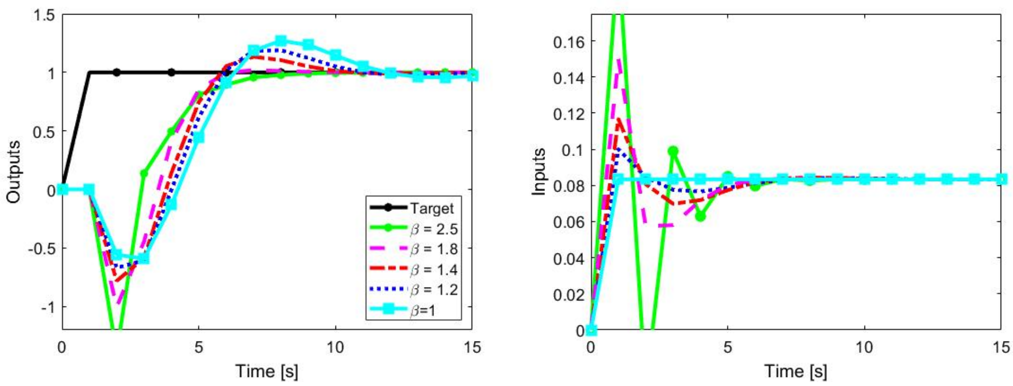
(**a**) $\lambda = 0.85$, $n = 15$ (**b**) $\lambda = 0.85$, $n = 15$

Figure 3. Comparison of the closed-loop performance of (16) for various choices of β on B_1.



(**a**) $\lambda = 0.97$, $n = 50$ (**b**) $\lambda = 0.97$, $n = 50$

Figure 4. Comparison of the closed-loop performance of (16) for various choices of β on B_2.



(**a**) $\lambda = 0.8$, $n = 5$ (**b**) $\lambda = 0.8$, $n = 5$

Figure 5. Comparison of the closed-loop performance of (16) for various choices of β on B_3.

4.3. Illustrations of the Speed-Up in CLSSPFC

A number of figures (Figures 7–9) are provided in this section to show the closed-loop performance for CLSSPFC with various choices of β. The reader is reminded that the case studies used here have challenging open-loop dynamics and, thus, are typically not amenable to simpler design approaches. Consequently, the plots of PFC and SSPFC are largely not included because they were unsatisfactory (e.g., see PFC in Figure 8b). The auto-tuned PID responses were also poor.

1. Figures 7–9 show the input and output responses for CLSSPFC with various choices of β. It is clear that, despite the challenging nature of the control problem, the proposal

gives a simple and effective mechanism for varying the closed-loop speed of response with the corresponding changes in input activity.

2. Figure 10a–c compare the tuning for $\beta = 1.8$ with the corresponding efficacy of more conventional approaches. It is clear that the conventional approaches have, by comparison, some lag and other undesirable behaviours and, thus, are not as effective at achieving a specified behaviour.

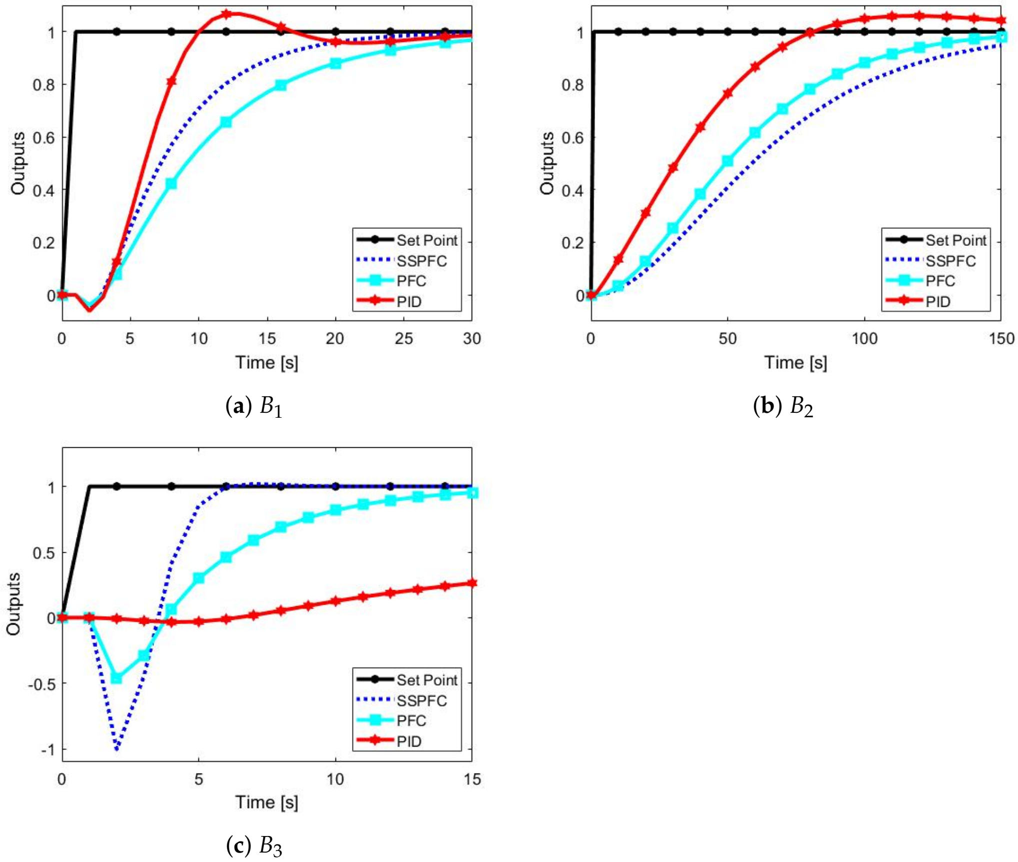
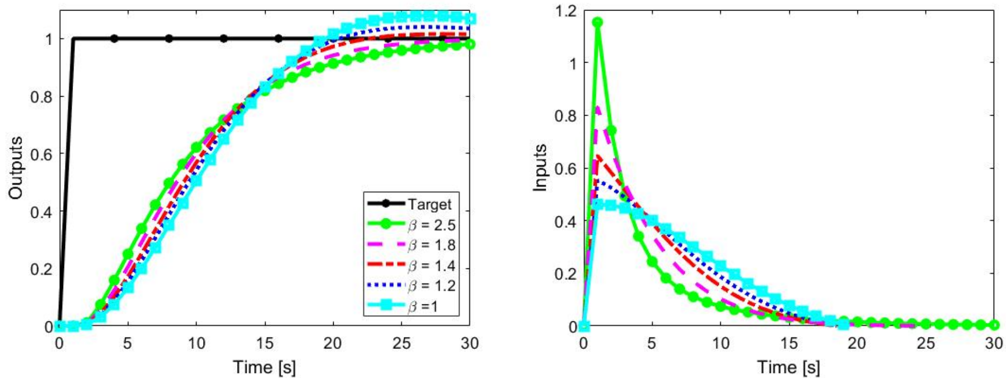


Figure 6. Comparison of the closed-loop performance of PID, SSPFC ($\beta = 1.8$), and PFC on B_1, B_2, B_3.

Remark 6. *The decision to remove some of the details from some figures (e.g., Figure 10b,c) was made to ensure that the most important parts of the line plots are clear and to use consistent sizing from one figure to another in order to allow for comparison. The omitted parts do not contain useful information (it is already evident that those responses were unsatisfactory), and changing the scales to include these would detract from the clarity of the parts on which we want to focus.*

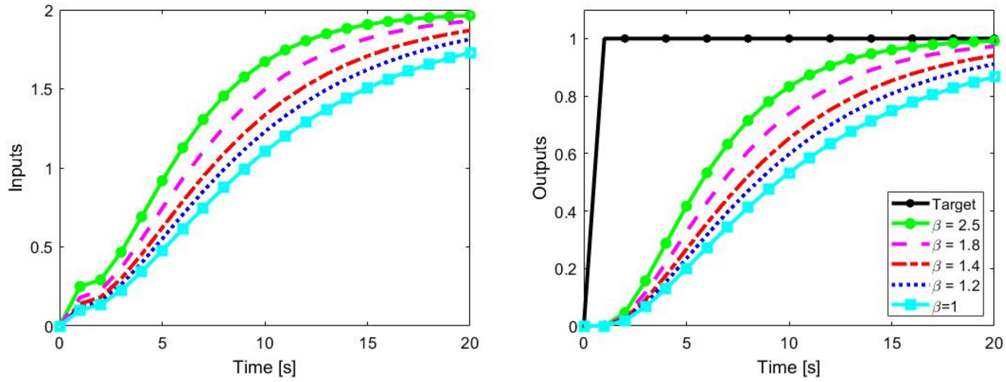
4.4. Disturbance Rejection

For completeness, this section provides a couple of examples to illustrate that the proposed approach and control laws are robust and, thus, deal with uncertainty. Figure 11 shows the impact on the outputs with an output disturbance of 0.1 for sample 25.



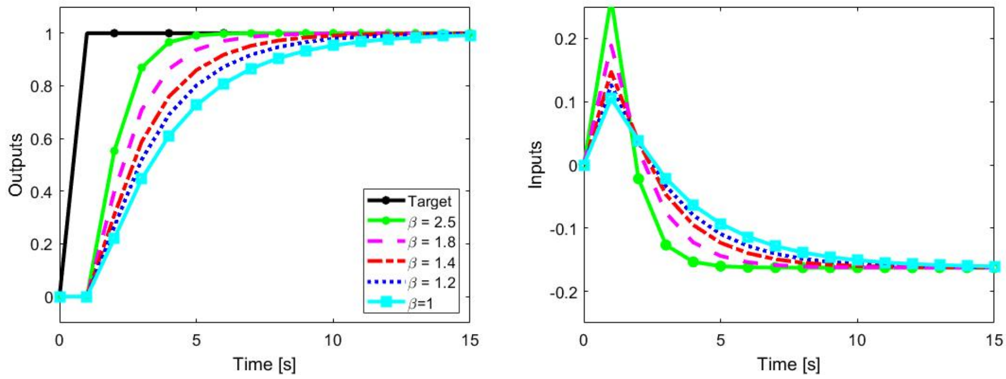
(**a**) $\lambda = 0.97$, $n = 8$ (**b**) $\lambda = 0.97$, $n = 8$

Figure 7. Comparison of the closed-loop performance of (16) for various choices of β on the boiler model.



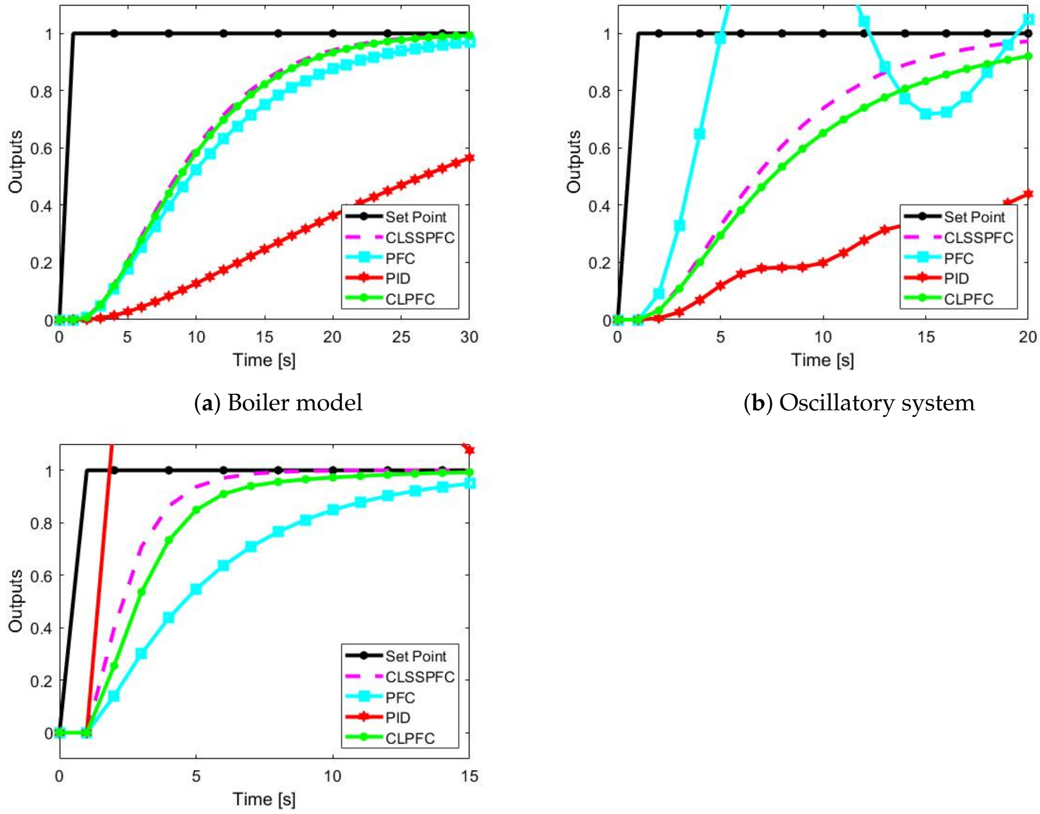
(**a**) $\lambda = 0.86$, $n = 4$ (**b**) $\lambda = 0.86$, $n = 4$

Figure 8. Comparison of the closed-loop performance of (16) for various choices of β on the oscillatory model.



(**a**) $\lambda = 0.6$, $n = 3$ (**b**) $\lambda = 0.6$, $n = 3$

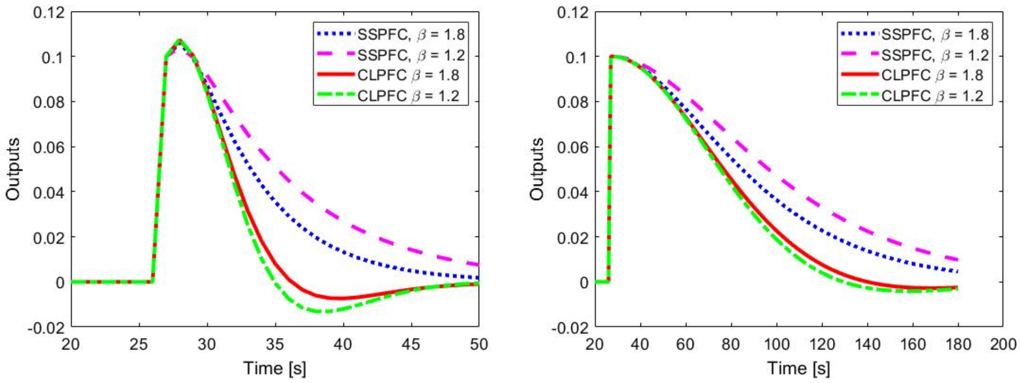
Figure 9. Comparison of the closed-loop performance of (16) for various choices of β on the unstable model.



(**a**) Boiler model (**b**) Oscillatory system

(**c**) *Reactorsystem*

Figure 10. Comparison of the closed-loop performance of PID, SSPFC, and CLPFC on the boiler, oscillatory, and unstable models.

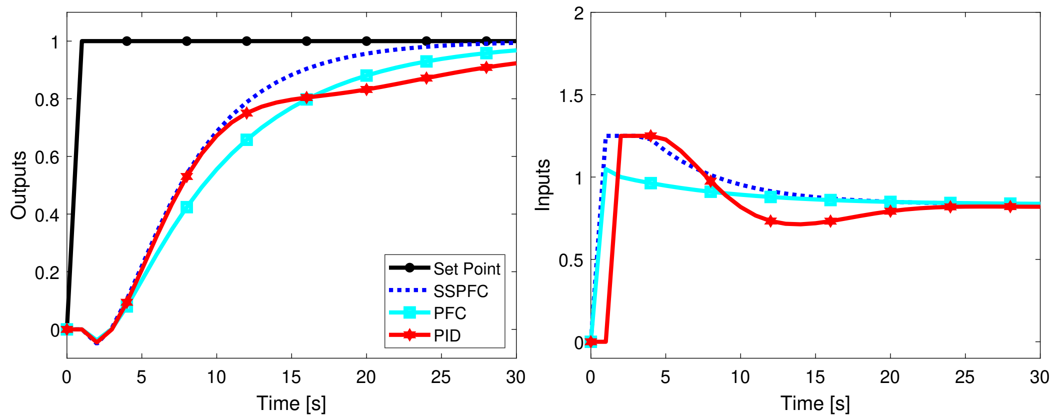


(**a**) B_1 (**b**) B_3

Figure 11. Responses of SSPFC and CLPFC with disturbance uncertainty for models B_1, B_3.

4.5. Constraint Handling

Finally, it is shown that the conventional constraint-handling approach in Section 2.3.2 is applicable; again, a suitable illustration is given in Figure 12, which shows efficacy of the proposal for example B_1 in comparison with PFC and PID.



(**a**) B_1 Outputs (**b**) B_1 Inputs

Figure 12. Comparison of the constrained closed-loop performance for model B_1 with $u \leq 1.25$.

5. Conclusions, Limitations, and Future Work

5.1. Concluding Remarks

This paper has presented, consolidated, and extended a conceptually new approach to PFC tuning. Specifically, it has proposed a solution to a recently understood weakness in the traditional PFC approach. The coincidence point is the lynchpin of existing PFC algorithms, but the definition of this [16] rather surprisingly undermines the core philosophy, as it introduces an inconsistency between objectives from one sample to the next; in effect, this usually introduces lag into the responses, so the desired dynamic is not achieved.

The current paper proposes a radically different alternative to the coincidence point while retaining the core requirement of simplicity for an equality expression based on n-step-ahead predictions in order to define the control law. The proposal here uses a comparison between offset free *open-loop* behaviour and the *optimised* predictions; the aim is to develop systematic but transparent tools for making a closed-loop system faster than an open-loop system. As seen in the figures, the proposed speed-up parameter β is highly transparent and, thus, easy to work with, providing low-skilled operators with a simple handle on closed-loop behaviour.

In addition, this paper has shown that the proposed method can easily be extended to systems with challenging dynamics by deploying closed-loop forms [1], and again, the transparency of the tuning is evident.

5.2. Limitations and Future Work

In general, PFC, like most finite-horizon MPC approaches, only offers a posteriori stability assurances. However, (10) does, in fact, have a simple a priori proof of stability, so it would be important to see whether and how this can be extended to (16).

Another interesting and important direction of future work is to consider, in detail, extensions to the nonlinear case. One major advantage of PFC approaches is that the computational simplicity of the control law's definition means that a nonlinear implementation is, compared to conventional MPC, cheap and simple to code. Hence, the intention is to propose, modify, and illustrate the conceptual approach for a number of nonlinear examples.

Author Contributions: This paper was a collaborative effort between both authors. J.A.R. provided the initial proposals and accurate communication of the concepts. M.S.A. developed the code and analysed the concepts in the case studies. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The second author would like to acknowledge the University of Sheffield for his PhD scholarship, which is funding his studies.

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

References

1. Rossiter, J.A.; Aftab, M.S. Review of Tuning Methods for Predictive Functional Control. *Processes* **2021**, *9*, 1140. [CrossRef]
2. 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.
3. Richalet, J. Industrial applications of model based predictive control. *Automatica* **1993**, *29*, 1251–1274. [CrossRef]
4. Fallasohi, H.; Ligeret, C.; Lin-Shi, X. Predictive Functional Control of an expansion valve for minimizing the superheat of an evaporator. *Int. J. Refrig.* **2010**, *33*, 409–418. [CrossRef]
5. 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.
6. Richalet, J.; Rault, A.; Testud, J.; Papon, J. Model predictive heuristic control: applications to industrial processes. *Automatica* **1987**, *14*, 413–428. [CrossRef]
7. 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.
8. Haber, R.; Bars, R.; Schmitz, U. *Predictive Control in Process Engineering: From the Basics to the Applications*; Wiley-VCH: Weinheim, Germany, 2011.
9. Richalet, J.; O'Donovan, D. *Predictive Functional Control: Principles and Industrial Applications*; Springer: Berlin, Germany, 2009.
10. Clarke, D.W.; Mohtadi, C. Properties of generalized predictive control. *Automatica* **1989**, *25*, 859–875. [CrossRef]
11. Cutler, C.R.; Ramaker, B.L. Dynamic matrix control: A computer control algorithm. *Jt. Autom. Control. Conf.* **1980**, *17*, 72. [CrossRef]
12. Rossiter, J.A.; Haber, R. The effect of coincidence horizon on predictive functional control. *Processes* **2015**, 3, 25–45. [CrossRef]
13. Zhang, Z.; Rossiter, J.A.; Xie, L.; Su, H. Predictive Functional Control for Integrator Systems. *J. Frankl. Inst.* **2020**, *357*, 4171–4186. [CrossRef]
14. Aftab, M.S.; Rossiter, J.A.; Zhang, Z. Predictive functional control for unstable first-order dynamic systems. In *CONTROLO 2020*; Goncalves, J.A., Ed.; Springer Nature Switzerland AG: Cham, Switzerland, 2021; pp. 1–11.
15. Abdullah, M.; Rossiter, J.A. Using Laguerre functions to improve the tuning and performance of predictive functional control. *Int. J. Control* **2021**, *94*, 202–214. [CrossRef]
16. Rossiter, J.A.; Abdullah, M. Improving the use of feedforward in Predictive Functional Control to improve the impact of tuning. *Int. J. Control* **2020**, *95*, 1206–1217. [CrossRef]
17. 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]
18. 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]
19. Rossiter, J.A. A priori stability results for PFC. *Int. J. Control* **2016**, *90*, 305–313. [CrossRef]
20. Rossiter, J.A. *A First Course in Predictive Control*, 2nd ed.; CRC Press: London, UK, 2018.
21. Rawlings, J.; Muske, K. The stability of constrained receding horizon control. *IEEE Trans. Autom. Control* **1993**, *38*, 1512–1516. [CrossRef]
22. Rossiter, J.A.; Kouvaritakis, B. Numerical robustness and efficiency of generalised predictive control algorithms with guaranted stability. *IEE Proc. D* **1994**, *141*, 154–162.
23. Rossiter, J.A.; Aftab, M.S. Exploiting Laguerre polynomials and steady-state estimates to facilitate tuning of PFC. In Proceedings of the 20th European Control Conference (ECC) Proceedings, European Control Association (EUCA), London, UK, 12–15 July 2022.
24. 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, 23–27 May 2011; pp. 306–313.
25. 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]
26. Abdullah, M.; Rossiter, J.A. Input Shaping Predictive Functional Control for Different Types of Challenging Dynamics Processes. *Processes* **2018**, *6*, 118. [CrossRef]
27. Rossiter, J.A. Input shaping for pfc: How and why? *J. Control Decis.* **2015**, *3*, 105–118. [CrossRef]
28. Mosca, E.; Zhang, J. Stable redesign of predictive control. *Automatica* **1992**, *28*, 1229–1233. [CrossRef]
29. 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.
30. 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]

31. Scokaert, P.O.; Rawlings, J.B. Constrained linear quadratic regulation. *IEEE Trans. Autom. Control* **1998**, *43*, 1163–1169. [CrossRef]
32. Aftab, M.S.; Rossiter, J.A. Pre-stabilised predictive functional control for open-loop unstable dynamic systems. In Proceedings of the 7th IFAC Conference on Nonlinear Model Predictive Control, Bratislava, Slovakia, 11–14 July 2021.
33. Aftab, M.S.; Rossiter, J.A. Predictive functional control for challenging dynamic processes using a simple prestabilization strategy. *Adv. Control Appl.* **2022**, *2022*, e102. [CrossRef]
34. Zhuo, W.; Shichao, W.; Yanyan, J. Simulation of control of water level in boiler drum. In Proceedings of the World Automation Congress, Puerto Vallarta, Mexico, 24–28 June 2012.
35. Nise, N.S. *Control System Engineering*; John Wiley & Sons, Inc.: New York, NY, USA, 2011.
36. 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]