Proceedings Paper:

Aftab, M.S. and Rossiter, J.A. orcid.org/0000-0002-1336-0633 (2022) Predictive functional control for difficult dynamic processes with a simplified tuning mechanism. In: Proceedings of 2022 UKACC 13th International Conference on Control (CONTROL). 2022 UKACC 13th International Conference on Control (CONTROL), 20-22 Apr 2022, Plymouth, UK. Institute of Electrical and Electronics Engineers , pp. 130-135. ISBN 9781665452014

https://doi.org/10.1109/Control55989.2022.9781444

Predictive Functional Control for Difficult Dynamic Processes with a Simplified Tuning Mechanism

Muhammad Saleheen Aftab
Dept. of Automatic Control and Systems Engineering
University of Sheffield
Sheffield, UK
msaftab1@sheffield.ac.uk

John Anthony Rossiter
Dept. of Automatic Control and Systems Engineering
University of Sheffield
Sheffield, UK
j.a.rossiter@sheffield.ac.uk

***Abstract*—Predictive functional control (PFC) is a cheap and simplified model predictive controller, which competes with PID in price and performance. While the tuning process in PFC for simple dynamics is well established and straightforward, it becomes far more ambiguous and often less effective for processes exhibiting challenging behaviour, such as poor damping, instability and/or non-minimum phase characteristics. In this paper, we present a *relative* PFC algorithm that, when implemented with pre-stabilised prediction dynamics if needed, simplifies performance tuning to merely adjusting one parameter. Furthermore, it provides far superior closed-loop control in practical scenarios, where the conventional PFC and PID fail to perform, as demonstrated with three simulation case studies.**

***Index Terms*—predictive functional control, pre-stabilisation, tuning**

I. Introduction

Predictive functional control (PFC) is a simplified and cost-effective model based predictive controller that competes with PID in cost and performance [1]. Being model based, it inherits most attributes from the mainstream MPC; properties such as dead-times and constraints handling are straightforward to implement unlike PID which requires additional complexity such as a Smith predictor [2] and anti-windup techniques [3]. Moreover, controller tuning in PFC distinctively relates to a physical characteristic (i.e. system rise time) which makes the tuning process comparatively meaningful. Consequently numerous successful PFC applications have been reported in the literature [4], [5].

For a well damped open-loop process, the conventional PFC operates by enforcing a match, the so-called coincidence, between the predicted and the desired response at a future sample by assuming constant control moves, where the desired response represents an ideal exponential trajectory initiated on the current output. By doing so, PFC comfortably achieves any desirable performance for stable first-order systems provided the coincidence occurs exactly one sample ahead [5], [6]. Similarly, parameter tuning guidelines for overdamped higher order systems are well established [7], although 100% target tracking is usually not achieved due to the initial lag in the system dynamics.

However, controller tuning becomes significantly less straightforward when difficult open-loop dynamics are present; for example processes with poor damping, instability and/or non-minimum phase characteristics have been particularly challenging to control [7], [8]. Clearly it is counter-intuitive to match an ideal exponential trajectory with such exotic behaviour at merely one future sample and expect a well-behaved response, although the overall closed-loop may still work due to the receding horizon. Nevertheless, such a design is highly unreliable and prone to failure, especially with uncertainties and/or tight actuation limits.

The primary reason for poor performance in challenging applications is the use of a constant input within the predictions which clearly lacks enough flexibility to handle such dynamics. An obvious solution in such cases is to use a more flexible parametrisation of the input function (see for instance [9]–[11]); nevertheless, these modifications deal with one aspect at a time, for instance, using Laguerre function for tuning improvement [9] and input shaping/pre-stabilisation to handle difficult dynamics [10], [11]. Furthermore, a recent study has pointed out the anomaly in prediction mechanism for higher order dynamics wherein the initialisation of target trajectory on the current process output embeds unnecessary delay into the future target values causing poorer tuning efficacy [12].

In this study, we tackle this discrepancy in two stages. Firstly, the concept of *pre-stabilisation* is utilised, if necessary, to transform difficult open-loop dynamics into a well-damped closed-loop prediction behaviour [10], [13], [14]. Secondly, a *relative* PFC algorithm is presented which simplifies controller tuning to simply selecting one parameter that speeds up or slows down the closed-loop performance as compared to a suitable benchmark response. Simulation case studies highlight the superior efficacy and performance of the proposal.

The rest of the paper is organised as follows: Section II briefly reviews the technicalities associated with conventional PFC, before moving on to the concept of pre-stabilised prediction dynamics in Section III. Next, the proposed relative PFC algorithm is presented in Section IV, followed by the tuning and closed-loop performance evaluation with computer simulations discussed in Section V. Finally, the paper concludes in Section VI highlighting the main contributions of the study.

II. Review of Predictive Functional Control

This section briefly reviews the basic characteristics of a conventional PFC algorithm. Consider a nth order transfer function model $a(z)\hat{y}_k = b(z)u_k$ of a well-damped open-loop process, which is used recursively to obtain $i-$step ahead predictions as follows [6]:

$$y_{k+i|k} = \mathbf{H}\underrightarrow{\mathbf{u}}_k + \mathbf{P}\underleftarrow{\mathbf{u}}_{k-1} + \mathbf{Q}\underleftarrow{\hat{\mathbf{y}}}_k + d_k \quad i = 1, 2, \ldots \quad (1)$$

where the vectors $\mathbf{H}$, $\mathbf{P}$ and $\mathbf{Q}$ are derived from the model parameters $a(z)$ and $b(z)$, with the associated input and output vectors defined accordingly:

$$\underrightarrow{\mathbf{u}}_k = \begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+i} \end{bmatrix}; \underleftarrow{\mathbf{u}}_{k-1} = \begin{bmatrix} u_{k-1} \\ u_{k-2} \\ \vdots \\ u_{k-n+1} \end{bmatrix}; \underleftarrow{\hat{\mathbf{y}}}_k = \begin{bmatrix} \hat{y}_k \\ \hat{y}_{k-1} \\ \vdots \\ \hat{y}_{k-n+1} \end{bmatrix} \quad (2)$$

The term $d_k = y_k - \hat{y}_k$ is added to remove prediction bias (y_k being the true process output and $\hat{y}_k$ the model output) and ensure offset free tracking. An ideal first order reference, initiated on the current y_k, is also defined:

$$r_{k+i} = R - (R - y_k)\rho^i \quad i = 1, 2, \ldots \quad (3)$$

where R is the set-point and ρ is the target pole (the primary tuning parameter), defined as $\rho = e^{-T_s/\tau}$ with T_s and τ being the sampling time and the target time constant respectively.

At each sample k, the current control u_k is used to enforce a match between the predicted y_k and r_k at a coincidence point n_y samples ahead. The prediction is based on an assumption of a constant future control signal $u_k = u_{k+1} = \cdots = u_{k+ny}$, but the decision is re-evaluated and updated at every sampling instant, thus forming a feedback mechanism. The conventional PFC control law is obtained using (1)-(3):

$$u_k = \frac{1}{h}[R - (R - y_k)\rho^{n_y} - (\mathbf{P}\underleftarrow{\mathbf{u}}_{k-1} + \mathbf{Q}\underleftarrow{\hat{\mathbf{y}}}_k + d_k)] \quad (4)$$

where $h = \sum_{j=1}^{n_y} H(j)$ and $H(j)$ is the j^{th} element of $\mathbf{H}$ and it is re-iterated that the conventional PFC tuning parameters are $\rho,\ n_y$.

Remark 1. *With the input prediction being constant, it is straightforward to implement simple saturation for a systematic handling of input constraints. Thus before applying to the plant, u_k is verified such that [6]:*

$$|u_k| > U \Rightarrow |u_k| = U, \quad |\Delta u_k| > D_U \Rightarrow |\Delta u_k| = D_U \quad (5)$$

where $\Delta u_k = u_k - u_{k-1}$ represents the sample wise rate of actuation. State and output constraints can also be handled relatively simply (iff feasible).

III. Pre-stabilised Prediction Dynamics

While the standard PFC works sufficiently well with simple dynamic problems, it performs poorly in challenging applications [7] and indeed appropriate selection of (ρ, n_y) may no longer be systematic or effective. The problem with difficult open-loop predictions obtained from unstable or poorly damped dynamics is the potential loss of numerical robustness due to large inconsistency between sample to sample computation of prediction matrices. The resulting predictions are, therefore, highly unreliable and could eventually lead to ill-posed decision making and loss of feasibility even if the unconstrained performance appears satisfactory [6]. The accepted practice in the mainstream MPC literature in such cases is to form closed-loop predictions using some form of classical feedback compensation [15], [16]. Based on a similar approach, a *pre-stabilised* PFC algorithm has been developed which demonstrates manifold performance improvement in comparison to the conventional PFC [10], [13], [14]. This concept is summarised below and will be utilised by the proposed Relative PFC algorithm presented in the following section.

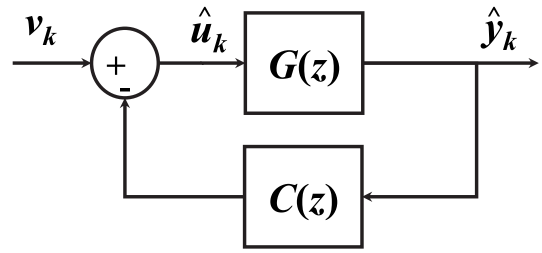

Fig. 1. Pre-stabilisation loop structure.

A. Concept of Pre-stabilisation

Consider a difficult open-loop process modelled as a nth order strictly proper transfer function $G(z)$ given as:

$$G(z) = \frac{\hat{y}_k}{\hat{u}_k} = \frac{b(z)}{a(z)} \quad (6)$$

where $a(z) = 1 + a_1 z^{-1} + \cdots + a_n z^{-n}$, $b(z) = b_1 z^{-1} + \cdots + b_n z^{-n}$ and $a(z)$ has factors including unstable and/or complex poles. $G(z)$ is compensated using a mth order bi-proper feedback controller $C(z)$, as shown in Fig. 1. Note that:

$$C(z) = \frac{q(z)}{p(z)} \quad (7)$$

where $p(z) = 1 + p_1 z^{-1} + \cdots + p_m z^{-m}$ and ,$q(z) = q_0 + q_1 z^{-1} + \cdots + q_m z^{-m}$. The resulting pre-stabilised prediction model is then:

$$G_s(z) = \frac{\hat{y}_k}{v_k} = \frac{p(z)b(z)}{p(z)a(z) + q(z)b(z)} = \frac{\beta(z)}{\alpha(z)} \quad (8)$$

where v_k is now the decision variable computed via an outer PFC loop. The actual process input u_k is related to v_k indirectly via the model input $\hat{u}_k$ ($u_k = \hat{u}_k$ only in the absence of uncertainties) as detailed in [10]. Here, we will use the final result:

$$u_k = B_0 v_k + f_k; \quad f_k = -\mathbf{A}\underleftarrow{\mathbf{u}}_{k-1} + \mathbf{B}\underleftarrow{\mathbf{v}}_{k-1} + \mathbf{E}\underleftarrow{\mathbf{d}}_k \quad (9)$$

where vectors $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{E}$ are obtained from the parameters $a(z)$, $\alpha(z)$, $p(z)$ and $q(z)$. Evidently, after pre-stabilisation, the degree-of-freedom is reparametrised appropriately, given a suitable inner controller, which can now work easily with the difficult dynamics.

Remark 2. *The parametrisation of u_k in* (9) *clearly makes the simple saturation policy for constraint handling less straightforward to implement; nevertheless, the methods for constraint validation in such cases are well documented (see for instance [10], [13], [17]). Since the current work does not bring any particular novelty in this regard, the available constraint handling algorithm [10] will be utilised in the simulation studies presented in the later section.*

B. Design of pre-stabilising compensator

The reader is reminded of the core purpose of pre-stabilisation, that is to transform the challenging open-loop dynamics into something more manageable for PFC. This includes filtering out unwanted oscillations from poorly damped systems and stabilising the open-loop unstable systems. Therefore, any standard feedback compensator that does the job without overly complicating the design is suitable. Nevertheless, it is recommended to start with the simple options such as P(D) or lead compensation [18] which are sufficient for a majority of first and second order difficult dynamics, and only implement more sophisticated alternatives such as pole placement [10] or pole cancellation [13] if the simpler choices are ineffective.

IV. Relative PFC Algorithm

Previous studies have highlighted the tuning deficiency of PFC for processes with difficult open-loop dynamics where it generally fails to meet the target performance [7], [19]. Clearly parameter selection in such cases is far less intuitive, and there is an obvious need for a more transparent mechanism that simplifies the tuning procedure. This section presents a *relative* PFC algorithm with simplified tuning as the core contribution, wherein the closed-loop performance is tuned relative to a suitable benchmark, rather than searching for ρ and n_y on absolute terms.

First it is noted that pre-stabilisation, if necessary, transforms the open-loop prediction model into $\alpha(z)\hat{y}_k = \beta(z)v_k$ providing output predictions as follows:

$$y_{k+n_y|k} = \mathbf{H}\underset{\rightarrow}{\mathbf{v}}_k + \mathbf{P}\underset{\leftarrow}{\mathbf{v}}_{k-1} + \mathbf{Q}\underset{\leftarrow}{\hat{\mathbf{y}}}_k + d_k \tag{10}$$

where $\mathbf{H}$, $\mathbf{P}$ and $\mathbf{Q}$ are now determined from $\alpha(z)$ and $\beta(z)$. If one selects $v_{k+i} = v_{ss}$, $\forall i \geq 0$ where v_{ss} is the expected steady-state input, the control law then obtained is the so-called mean level (or open-loop) PFC [6], which mirrors the open-loop transient performance along with offset free tracking. For the pre-stabilised system $G_s(z)$:

$$v_{ss} = \frac{R - d_k}{G_s(1)} \quad \because y_{ss} = y(1) = R \tag{11}$$

where $G_s(1)$ is the steady-state system gain. In practice, it is straightforward to achieve the mean-level PFC by simply selecting a large enough horizon, preferably beyond the settling time of the pre-stabilised step response. With target R and $v_{k+i} = v_{ss}$ $\forall i \geq 0$, the tracking error converges as follows:

$$e_{ss}(k+i) = R - (hv_{ss} + \mathbf{P}\underset{\leftarrow}{\mathbf{v}}_{k-1} + \mathbf{Q}\underset{\leftarrow}{\hat{\mathbf{y}}}_k + d_k) \tag{12}$$

which compares to the error convergence when an alternative fixed input $v_{k+i} = v_k$ $\forall i \geq 0$ is used. In this case:

$$e(k+i) = R - (hv_k + \mathbf{P}\underset{\leftarrow}{\mathbf{v}}_{k-1} + \mathbf{Q}\underset{\leftarrow}{\hat{\mathbf{y}}}_k + d_k) \tag{13}$$

Thus to obtain a faster convergence than the benchmark (12), one has to select a v_k correspondingly more active than v_{ss}. Lemma 1 below formalises this concept.

Lemma 1. *In the nominal state and zero initial conditions, the choice $v_k = \theta v_{ss}$ for the target R provides an error convergence which is γ times* (12) *such that:*

$$\gamma = \frac{G_s(1) - h\theta}{G_s(1) - h} \tag{14}$$

Proof. With d_k, $\underset{\leftarrow}{\mathbf{v}}_{k-1}$ and $\underset{\leftarrow}{\hat{\mathbf{y}}}_k$ all zero, and $v_k = \theta v_{ss}$ the initial errors are related as follows:

$$R - h\theta v_{ss} = \gamma(R - hv_{ss})$$

using (11) then implies:

$$1 - \frac{h\theta}{G_s(1)} = \gamma\left(1 - \frac{h}{G_s(1)}\right)$$

which simplifies to (14) after simple manipulations. □

Lemma 2. *For the chosen input activity θ and the error convergence γ defined above, the Relative PFC (RPFC) control law is given by:*

$$v_k = \gamma v_{ss} + \frac{1-\gamma}{h}\left[R - \left(\mathbf{P}\underset{\leftarrow}{\mathbf{v}}_{k-1} + \mathbf{Q}\underset{\leftarrow}{\hat{\mathbf{y}}}_k + d_k\right)\right] \tag{15}$$

Proof. Using Lemma 1 and equations (12)-(13), it is clear that:

$$e(k+i) = \gamma e_{ss}(k+i), \quad \forall i \geq 0$$

or,

$$R - (hv_k + \mathbf{P}\underset{\leftarrow}{\mathbf{v}}_{k-1} + \mathbf{Q}\underset{\leftarrow}{\hat{\mathbf{y}}}_k + d_k) =$$
$$\gamma\left[R - (hv_{ss} + \mathbf{P}\underset{\leftarrow}{\mathbf{v}}_{k-1} + \mathbf{Q}\underset{\leftarrow}{\hat{\mathbf{y}}}_k + d_k)\right]$$

which simplifies to the control law (15). □

Theorem 1. *The closed-loop performance can be tuned with the parameter θ via plant control u_k given by* (9).

Proof. Assuming zero initial conditions and no uncertainty, it is clear from (9) that after pre-stabilisation the initial plant control is $u_k = B_0 v_k$. If $v_k = \theta v_{ss}$ then $u_k = \theta(B_0 v_{ss}) = \theta u_{ss}$. Hence, the initial u_k will be θ times the one obtained via mean-level PFC, and therefore will tune the closed-loop performance accordingly. □

Algorithm 1 discusses parameter selection for the desired closed-loop performance.

Algorithm 1 Selecting parameter θ.

- $0 < \theta < 1$ reduces input activity resulting in a slower closed-loop performance. For example, $\theta = 0.5$ uses an initial input half as active as the mean-level benchmark to produce a relatively slower response.
- $\theta = 1$ is equivalent to the mean-level (open-loop) tuning.
- $\theta > 1$ increases input activity with a faster performance. For example, $\theta = 2$ uses an initial input twice as aggressive as the mean-level benchmark to produce a comparatively faster response.

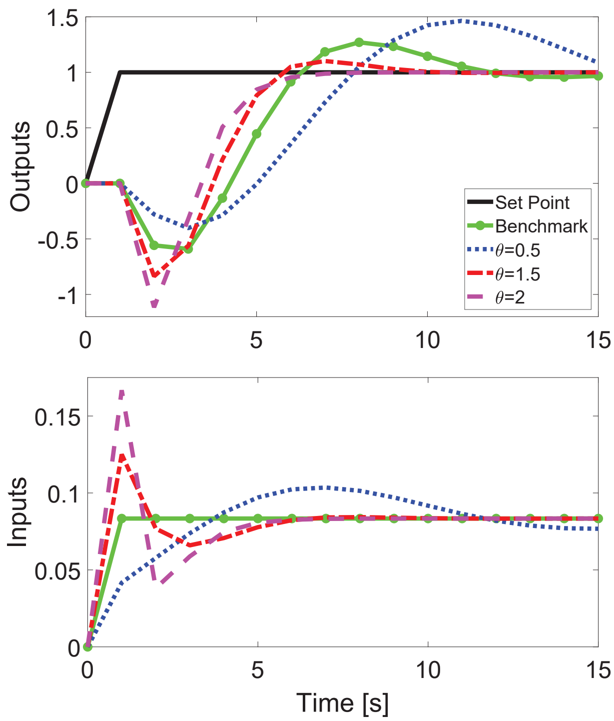

Fig. 2. Tuning efficacy of RPFC for open-loop G_1 in nominal conditions.

Remark 3. *It is advised not to select too large θ or the initial input could be too aggressive to achieve practically. Generally a commendable performance is attainable with θ up to 2-3, given a satisfactory open-loop dynamic behaviour.*

To sum up, the main benefit of the proposal is obvious: it reduces performance tuning to simply one statement, that is how fast or slow one wants the closed-loop system to respond. Of course, a well-behaved (implicitly stable) prediction model is necessary for implementation, which is achievable via pre-stabilisation of difficult dynamics if required. This is unlike the standard procedure generally implemented in PFC, which requires tedious offline analysis of open-loop step response overlaying multiple target trajectories to find the appropriate (ρ, n_y) pair [7]. A similar argument holds with PID for which selecting parameters K_p, K_i and K_d is arguably less intuitive than the proposed tuning algorithm discussed above.

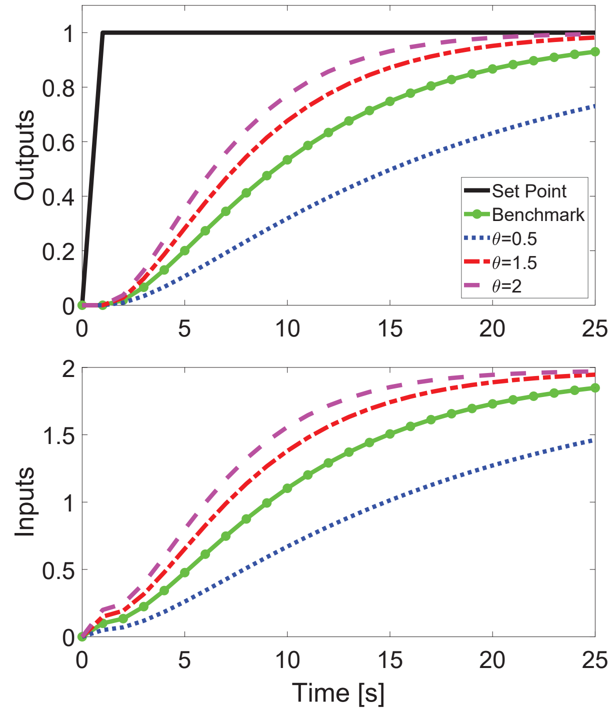

Fig. 3. Tuning efficacy of RPFC for pre-stabilised G_2 in nominal conditions.

V. Simulation Studies

In this section, the tuning efficacy and closed-loop performance of the proposal will be evaluated with three difficult open-loop systems. The process G_1 exhibits slightly underdamped but significantly non-minimum phase characteristics [20], G_2 is the representative second-order model of thermoacoustic oscillations in mechanical engines [21], and G_3 represents a second-order unstable model of a continuous stirred tank reactor [10]. These models are given as follows:

$$G_1 = \frac{-6.69z^3 + 7.86z^2 + 2.39^z + 0.002}{z^4 - 1.23z^3 + 0.54z^2 - 0.006z},$$

$$G_2 = \frac{0.19z + 0.18}{z^2 - 1.23z + 0.96}, \text{ and } G_3 = \frac{2.102z + 0.401}{z^2 - 1.465z + 0.058}$$

To highlight the benefits of the proposed RPFC algorithm, the closed-loop performance will be evaluated in real world scenarios against conventional PFC (CPFC), PID and pre-stabilised conventional PFC (PCPFC) for G_2 and G_3 which require pre-stabilisation as discussed in Section III.

A. Pre-stabilisation of difficult open-loop dynamics

Clearly the open-loop predictions obtained with G_1 will be convergent albiet with an initial lag due to non-minimum phase characteristic, therefore can be used without pre-compensation. On the other hand, both G_2 (poorly damped) and G_3 (unstable) exhibit challenging behaviour that must be pre-stabilised for a well-posed decision making with PFC.

A simple proportional compensator $C_2 = -1.88$ sufficiently filters out the unwanted oscillations in the open-loop step response of G_2, providing the pre-stabilised prediction model

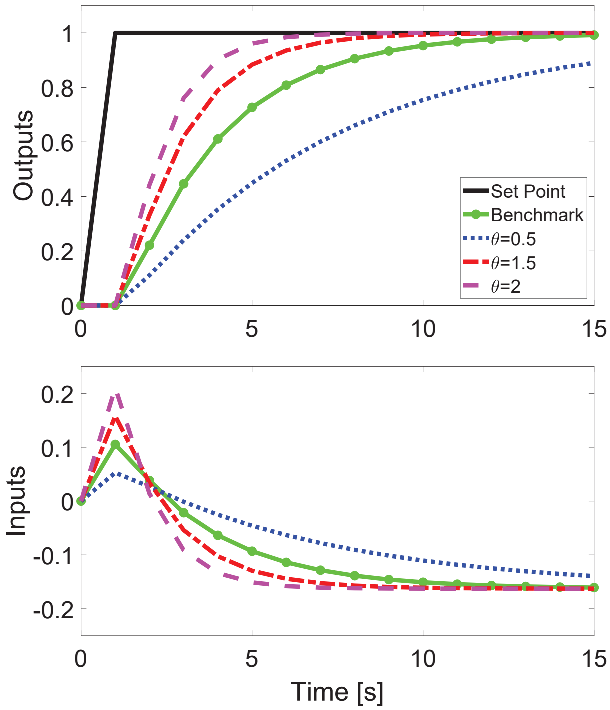

Fig. 4. Tuning efficacy of RPFC for pre-stabilised G_3 in nominal conditions.

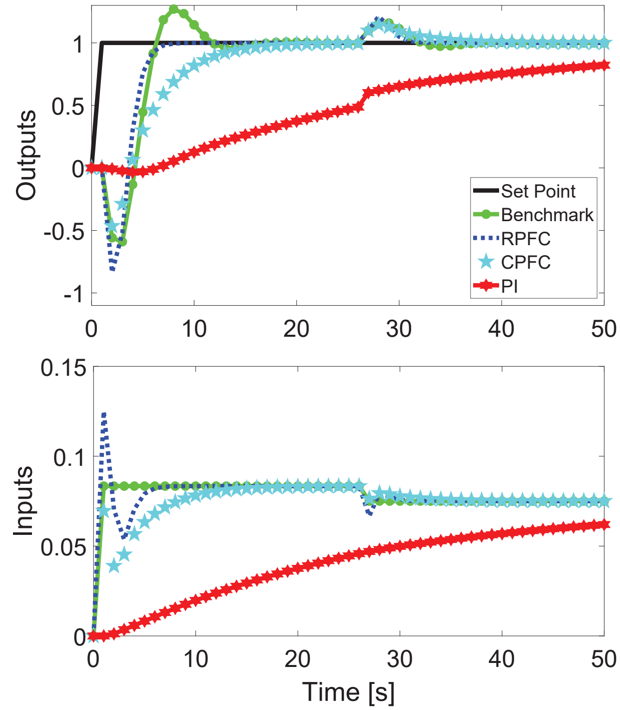

Fig. 5. Comparison of closed-loop performance for G_1 subject to $|\Delta u_k| \leq 0.125$ and 10% output disturbance introduced at 25th second between RPFC ($\theta = 2$), CPFC ($\rho = 0.75$, $n_y = 5$) and PI ($K_p = 0.0012$, $K_i = 0.0023$).

$G_{s,2} = \dfrac{0.19z + 0.18}{z^2 - 1.58z + 0.61}$ with overdamped poles at $z = 0.88, 0.70$. For G_3, a P(D) compensator fails to satisfactorily stabilise the dynamics, therefore a pole placement controller $C_3 = \dfrac{0.303z - 0.012}{z + 0.085}$ was designed ([10]) resulting in the pre-compensated model $G_{s,3} = \dfrac{2.102z^2 + 0.580z + 0.034}{z^3 - 0.743z^2 + 0.028z}$ and stable poles at $z = 0, 0.04, 0.7$.

B. Analysis of tuning efficacy with RPFC

The tuning efficacy of the proposed RPFC algorithm for the open-loop G_1 and the pre-stabilised G_2 and G_3 has been analysed in Figs. 2-4 respectively. It is clear that the parameter θ is succesful in slowing down (with $\theta = 0.5$) or speeding up (with $\theta = 1.5, 2$) the closed-loop response by correspondingly changing the initial input as compared to the mean-level benchmark ($\theta = 1$). Clearly performance tuning with θ in the proposal is far more straightforward and meaningful than finding ρ and n_y in the conventional PFC, or indeed K_p, K_i and K_d in the standard PID algorithms even when presented with difficult open-loop dynamic behaviour.

C. Comparison of closed-loop performance with constraints and uncertainties

We compare and analyse the closed-loop performances for G_1, G_2 and G_3 as shown in Figs. 5-7. Notably, the proposed RPFC in each case outperforms the conventional PFC, pre-stabilised or not, and the PID controllers in the presence of constraints and uncertainties. The key observations are:

- With $\theta = 2$, the initial RPFC control input is slightly less than $2v_{ss}$ due to the effect of constraints (except for G_3). Yet, the achieved closed-loop performance is faster than every alternative, with smooth and quicker disturbance rejection in each case, and especially for G_3 in the presence of unmodelled dynamics.
- The CPFC for G_1 although appears satisfactory albeit with significantly slower transient performance, it fails completely for both G_2 and G_3 with uncertainties. While the pre-stabilised CPFC considerably improves performance, it is still slower than RPFC with relatively sluggish disturbance rejection.
- The PI(D) controller, tuned using MATLAB's robust PID tuner [22], exhibits the poorest closed-loop performance, clearly signifying the importance of using (pre-stabilsed) prediction dynamics in the decision making.

To sum up, these examples have clearly highlighted the benefits of RPFC in difficult applications where both the conventional PFC and PID fail to perform.

VI. Conclusions

This paper has addressed the tuning deficiency of PFC, especially associated with difficult open-loop dynamics, by proposing a relative predictive functional control algorithm that simplifies performance tuning to trivial selection of one parameter that speeds up or slows down the transient response as compared to an open-loop benchmark. This implementation implicitly assumes availability of a smooth and well-damped prediction behaviour, which in turn necessitates preconditioning of difficult open-loop systems, for instance, using

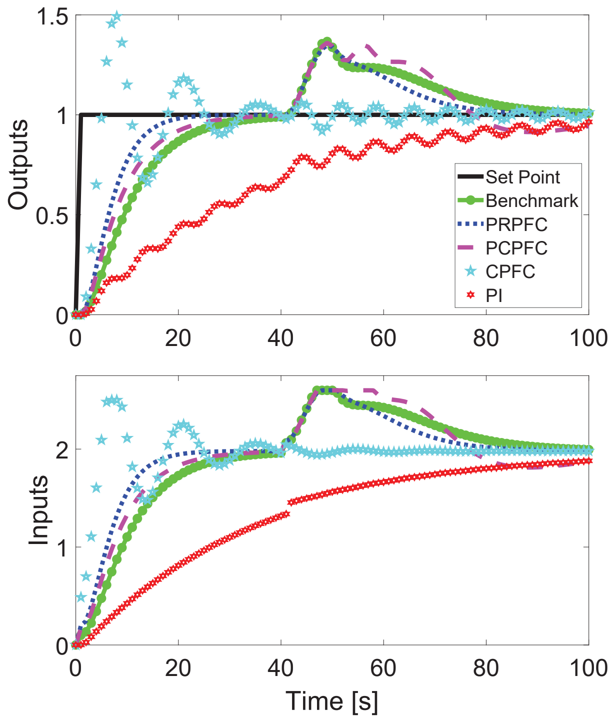

Fig. 6. Comparison of closed-loop performance for G_2 subject to $|u_k| \leq 2.5$ and 10% input disturbance introduced at 40th second between PRPFC ($\theta = 2$), PCPFC/CPFC ($\rho = 0.86$, $n_y = 4$) and PI ($K_p = 0.028$, $K_i = 0.055$).

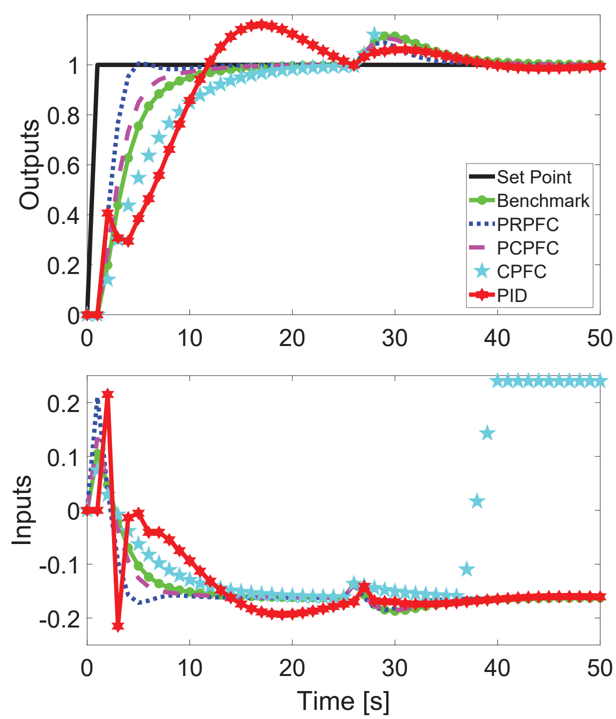

Fig. 7. Comparison of closed-loop performance for G_3 subject to $|u_k| \leq 0.21$ with unmodelled pole at $z = 0.1$ and 15% input disturbance introduced at 25th second between PRPFC ($\theta = 2$), PCPFC/CPFC ($\rho = 0.6$, $n_y = 3$) and PID ($K_p = 0.321$, $K_i = 0.038$, $K_d = 0.323$).

classical feedback compensation. The techniques to do so are, nonetheless, straightforward and trivial enough to be implemented easily without expert intervention. The numerical examples have clearly demonstrated the superiority of the proposal in real world scenarios where the standard PID and PFC algorithms have displayed a rather below par control performance. Although these results are promising, as a future work, the authors plan to extend the scope of validation to realtime experiments in a range of difficult industrial processes.

References

[1] R. Haber, J. A. Rossiter, and K. Zabet, "An alternative for pid control: Predictive functional control-a tutorial," in *2016 American Control Conference (ACC)*. IEEE, 2016, pp. 6935–6940.

[2] C. G. S. Skogestad, "Should we forget the smith predictor?" *IFAC-PapersOnLine*, vol. 51, no. 4, pp. 769–774, 2018.

[3] A. Visioli, *Practical PID Control*. Springer London, 2006.

[4] J. Richalet and D. O'Donovan, "Elementary predictive functional control: A tutorial," in *2011 International Symposium on Advanced Control of Industrial Processes (ADCONIP)*, May 2011, pp. 306–313.

[5] J. Richalet and D. O'Donovan, *Predictive functional control: principles and industrial applications*. Springer Science & Business Media, 2009.

[6] J. Rossiter, *A first course in predictive control*. CRC Press, 2018.

[7] J. Rossiter and R. Haber, "The effect of coincidence horizon on predictive functional control," *Processes*, vol. 3, no. 1, pp. 25–45, 2015.

[8] J. A. Rossiter and M. S. Aftab, "A comparison of tuning methods for predictive functional control," *Processes*, vol. 9, no. 7, p. 1140, jun 2021.

[9] M. Abdullah and J. A. Rossiter, "Using laguerre functions to improve the tuning and performance of predictive functional control," *International Journal of Control*, vol. 94, no. 1, pp. 202–214, mar 2019.

[10] M. S. Aftab and J. A. Rossiter, "Pre-stabilised predictive functional control for open-loop unstable dynamic systems," in *Proceedings of 7th IFAC Conference on Nonlinear Model Predictive Control*, 2021.

[11] J. Rossiter, "Input shaping for pfc: how and why?" *Journal of control and decision*, vol. 3, no. 2, pp. 105–118, 2016.

[12] J. A. Rossiter and M. Abdullah, "Improving the use of feedforward in predictive functional control to improve the impact of tuning," *International Journal of Control*, pp. 1–12, nov 2020.

[13] M. S. Aftab and J. A. Rossiter, "Predictive functional control with explicit pre-conditioning for oscillatory dynamic systems," in *Proceedings of 2021 European Control Conference*, 2021.

[14] M. S. Aftab, J. A. Rossiter, and Z. Zhang, "Predictive Functional Control for Unstable First-Order Dynamic Systems," in *CONTROLO 2020*, ser. Lecture Notes in Electrical Engineering, J. A. Gonçalves, M. Braz-César, and J. P. Coelho, Eds., vol. 695. Cham: Springer International Publishing, 2021, pp. 12–22.

[15] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[16] J. A. Rossiter, B. Kouvaritakis, and M. Rice, "A numerically robust state-space approach to stable-predictive control strategies," *Automatica*, vol. 34, no. 1, pp. 65–73, 1998.

[17] M. Abdullah and J. Rossiter, "Input shaping predictive functional control for different types of challenging dynamics processes," *Processes*, vol. 6, no. 8, p. 118, 2018.

[18] N. S. Nise, *Control Systems Engineering, Sixth Edition*. John Wiley & Sons, 2007.

[19] M. Abdullah, J. Rossiter, and R. Haber, "Development of constrained predictive functional control using laguerre function based prediction," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10 705–10 710, 2017.

[20] Q. Zhu, J. Qiu, I. Delshad, M. Nibouche, and Y. Yao, "U-model enhanced control of non-minimum phase systems," vol. 51, no. 15, pp. 3146–3162, aug 2020.

[21] A. M. Annaswamy and S. Hong, *The Control Handbook*. CRC Press, 2011, ch. Control of Unstable Oscillations in Flows.

[22] MathWorks, "Pid tuning algorithm," 2021. [Online]. Available: https://www.mathworks.com/help/control/getstart/pid-tuning-algorithm.html