Received: 6 September 2021 | Revised: 4 January 2022 | Accepted: 11 March 2022

DOI: 10.1002/adc2.102

ORIGINAL ARTICLE

WILEY

Predictive functional control for challenging dynamic processes using a simple prestabilization strategy

Muhammad Saleheen Aftab | **John Anthony Rossiter**

Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, UK

Correspondence
Muhammad Saleheen Aftab, Department of Automatic Control and Systems Engineering, University of Sheffield, Mappin Street, Sheffield S1 3JD, UK.
Email: msaftab1@sheffield.ac.uk

Funding information
University of Sheffield

Abstract

Predictive functional control (PFC) is a straightforward and cheap model-based technique for systematic control of well-damped open-loop processes. Nevertheless, its oversimplified design characteristics are often the cause of diminished efficacy in more challenging applications; processes involving lightly damped and/or unstable dynamics have been particularly difficult to control with PFC. This paper presents a more sustainable solution for such applications by integrating the concept of prestabilization within the predictive functional control formulation. This is essentially a two-stage synthesis wherein the undesirable open-loop dynamics are first compensated, using a well-understood classical approach such as proportional integral derivative (PID), before implementing predictive control in a cascade structure. The proposal, although comes with significant implications for tuning and constraint handling, is, nonetheless, straightforward and provides improved closed-loop control in the presence of external perturbations compared to the standard PFC and the PID algorithms, as demonstrated with two industrial case studies.

KEYWORDS

coincidence horizon, constraint handling, modeling uncertainty, prestabilization, predictive functional control

1 | INTRODUCTION

In process industries, a sustainable feedback control loop needs to be one that is easy to maintain and retune using local staff rather than consultants. Hence, it is advantageous when components of the design are based on simple classical approaches such as proportional integral derivative (PID) which are well understood. This paper considers how predictive functional control (PFC), a low-cost approach to model predictive control (MPC), can exploit simple classical designs within the overall approach and use simple intuitive tuning of the predictive control aspects thereafter.

PFC, since its introduction in the 1970s,[1] has emerged as a strong competitor to the widely popular PID algorithm, especially for single-input-single-output industrial process control loops. The advantages of PFC significantly outweigh those of PID in that it systematically handles process dead-times and constraints with an equivalent cost and complexity threshold, but for which PID requires additional complexity such as Smith predictors[2] and anti-windup algorithms.[3]

Moreover, controller tuning in PFC is intuitive, relating to the physical characteristic of system time constant, which, in principle, makes the tuning process relatively meaningful.[4] Consequently numerous successful PFC implementations have been reported in the literature.[5,6]

Being a model-based approach, PFC has inherited fundamental attributes from the mainstream MPC family,[7] yet it differs significantly from other predictive controllers in the parameterization of the input trajectory and the associated *optimization*. In PFC, the manipulated variable is predefined as a linear combination of polynomial basis functions, whose order depends upon the shape and characteristic of the target.[5,8] For a constant set-point, the predicted input parameterises to just one degree-of-freedom, eliminating the need for the complex optimisation routines generally associated with high-end MPC (e.g. DMC,[9] GPC[10]). This on one hand simplifies computation, but on the other hand necessitates heuristics for constrained predictive control problems.[11] Unlike advanced approaches, simple clipping, saturation, or simplified back calculation have been the commonly deployed constraint management protocols in PFC.

The basic PFC algorithm operates by matching output predictions with a desired first-order response at only one future point, known as the coincidence point, and with a fixed control action.[5-7] Intuitively this approach is effective as long as the model behavior is smooth and monotonically convergent after immediate transients.[12] A prime example is stable first-order plants for which PFC is proven to drive the controlled variable to any desirable target trajectory provided *coincidence* occurs exactly one-step ahead.[7] Similar closed-loop performance could be expected with well-damped higher-order dynamics although a coincidence point of one may not suffice due to the initial lag in the response. Nevertheless, parameter tuning for such simple systems is well understood in literature.[12]

However, what happens when the dynamic behavior is oscillatory, nonminimum phase or, in the worst scenario, completely divergent? Simply put, PFC loses efficacy in these difficult situations. The reason is inconsistency within the implied long-range predictions that deviate from the assumed ideal behaviour after coincidence. Researchers argue that a constant future input may not be sufficient as this lacks enough degrees-of-freedom to tackle difficult dynamics.[4,6,13,14] Although the conventional PFC may still work in some cases due to the application of receding horizon,[7,12] the decision-making is unreliable and prone to failure, especially with tight constraints and/or uncertainties. To overcome difficulties associated with challenging dynamics, a manual for PFC practitioners[6] suggests a variety of possible modifications on a case-to-case basis. Nevertheless, these solutions lack an over-arching systematic design procedure, and unsurprisingly have a very limited applicability.

For challenging applications, various modified PFC algorithms implementing different parameterisations of the decision variable have also been investigated. One proposal[13] recommends altering the input by separating and subsequently cancelling the unwanted dynamics to obtain convergent predictions. This method provides many-fold performance improvements while retaining the basic PFC characteristics but lacks practicality as the proposed minimum moves shaping may produce aggressive input activity and could be quite sensitive to parameterisation errors. Another input shaping proposal[4] ensures relatively less aggressive control moves by allowing predictions to converge over many more samples. This method, tested on numerous simulation models and hardware application, outperforms the predecessor but relies on rather less-intuitive offline computations. Yet another proposal[15] suggests decomposing the higher-order model into multiple first-order subsystems to benefit from simple tuning procedure. But such decomposition for oscillatory dynamics embeds complex number algebra into the computations which may not work easily with general purpose industrial programmable logic controllers (PLCs).[14]

A more recently proposed alternative, building on common practice in the more mainstream MPC literature,[16,17] is to prestabilize the undesirable open-loop predictions using an internal feedback compensation loop. While this concept within PFC has largely been limited to first-order unstable[18] or integrator dynamics[19,20] and very simple prestabilization compensators, one recent study[21] has extended its scope to higher-order unstable dynamics using some more involved inner compensation schemes, resulting in promising performance attributes but at the cost of increased constraint handling complexity. Another recent study[22] has suggested an improved and more meaningful parameter tuning after prestabilization;[22] a benefit that significantly outweighs the slight intricacy in constraint handling that may arise due to the use of complicated internal loops. Nevertheless, the need for a more thorough investigation in this context is evident; an objective which the current paper aims to accomplish. Therefore, the primary contribution, building on the recent proposal,[21] is the development of a systematic but simple PFC design framework for underdamped and unstable dynamic processes, integrating an intuitive tuning algorithm along with straightforward guidelines to perform efficient constraint management.

The remainder of this paper is organised as follows: Section 2 defines the problem and sets control objectives. The conventional PFC is briefly reviewed in Section 3, followed by a detailed discussion on the prestabilized PFC framework in Section 4. Two feedback compensation proposals are discussed next in Section 5 before discussing the proposal for constraint management with prestabilization in Section 6. The simulation case studies follow in Section 7 which presents performance comparisons with standard PFC and PID controllers. Finally, the paper concludes in Section 8.

2 | PROBLEM STATEMENT

Consider a dynamic process characterised by an nth-order strictly proper transfer function model $G(z)$ such that:

$$G(z) = z^{-n_d} G_0(z); \quad G_0(z) = \frac{b(z)}{a(z)}, \tag{1}$$

where $a(z)$ and $b(z)$ are coprime with $a(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n}$, $b(z) = b_1 z^{-1} + b_2 z^{-2} + \cdots + b_n z^{-n}$, respectively, and n_d is the process deadtime. It is assumed that the delay-free model $G_0(z)$ exhibits oscillatory or divergent dynamic behaviour demonstrated by complex or unstable open-loop poles. The process may also be subject to the following limits:

$$\underline{u} \le u_k \le \bar{u}, \quad \Delta\underline{u} \le \Delta u_k \le \Delta\bar{u}, \quad \underline{y} \le y_k \le \bar{y}, \tag{2}$$

where $\Delta = 1 - z^{-1}$. The problem addressed in this paper deals with the design of PFC of the process modeled as $G(z)$, by stabilising and/or conditioning the difficult open-loop dynamics using a simple classical feedback compensation approach. The controller is expected to exhibit some degree of robustness against modelling uncertainty and/or external perturbations.

3 | REVIEW OF PFC

This section briefly reviews the basic characteristics of a conventional predictive functional controller along with its fundamental weaknesses in handling difficult open-loop dynamics, followed by a detailed analysis of the prestabilized PFC (PPFC) algorithm proposed for such applications in the subsequent sections.

3.1 | Conventional PFC algorithm

For a well-damped open-loop process, the conventional PFC works as follows: at every time sample k, the current control input u_k is used to enforce a match between the predicted plant output y_k and a predefined reference trajectory r_k at a coincidence point n_y samples ahead in the future. 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 reference trajectory represents an ideal first-order response, initiated on the current output given by (for a system with deadtime of n_d samples):

$$r_{k+n_d+i} = R - (R - E[y_{k+n_d|k}])\rho^i, \quad i = 1, 2, \ldots \tag{3}$$

where R is the set-point, $E[y_{k+n_d|k}]$ is the current estimate/prediction of the delayed output and ρ is the target pole ($0 < \rho < 1$), defined as $\rho = e^{-T_s/\tau}$ with T_s and τ being the sampling time and the target time constant, respectively. Therefore, at the point of coincidence n_y, by definition, one obtains:

$$y_{k+n_y|k} = R - (R - E[y_{k+n_d|k}])\rho^{n_y} = r_{k+n_d+n_y}, \tag{4}$$

where the notation $k + x|k$ means the x-step ahead prediction made at the current sample k. The standard practice, as shown in Figure 1, is to simulate a delay-free independent model $G_0(z)$ in parallel with the plant using the same input u_k (a formulation similar to Smith predictor) which in essence provides n_d samples out of synchronization output prediction

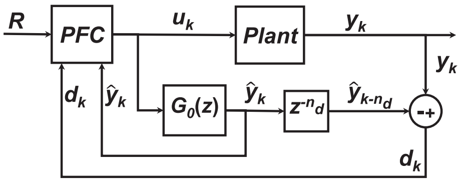


FIGURE 1 The standard predictive functional control control architecture with independent internal model

at the current k. Furthermore, the independent structure tends to induce prediction bias due to uncertainties, causing an offset in the steady-state. For bias-free predictions, a correction term d_k must be included in algebra such that:

$$\{d_k = y_k - \hat{y}_{k-n_d}\} \quad \Rightarrow \quad \{E[y_{k+n_d|k}] = \hat{y}_k + d_k\}, \tag{5}$$

where $\hat{y}_k$ is the independent model output. The output prediction at the coincidence point n_y is recursively obtained using the structure $a(z)\hat{y}_k = b(z)u_k$ such that[7]

$$\hat{y}_{k+n_y|k} = \mathbf{H}\underset{\rightarrow}{\mathbf{u}}_k + \mathbf{P}\underset{\leftarrow}{\mathbf{u}}_{k-1} + \mathbf{Q}\underset{\leftarrow}{\hat{\mathbf{y}}}_k, \tag{6}$$

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

$$\underset{\rightarrow}{\mathbf{u}}_k = \begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+n_y} \end{bmatrix}; \quad \underset{\leftarrow}{\mathbf{u}}_{k-1} = \begin{bmatrix} u_{k-1} \\ u_{k-2} \\ \vdots \\ u_{k-n+1} \end{bmatrix}; \quad \underset{\leftarrow}{\hat{\mathbf{y}}}_k = \begin{bmatrix} \hat{y}_k \\ \hat{y}_{k-1} \\ \vdots \\ \hat{y}_{k-n+1} \end{bmatrix}. \tag{7}$$

As $u_{k+i} = u_k, \forall i > 0$, combining (4)–(6) results in the following PFC control law:

$$u_k = \frac{R - (R - E[y_{k+n_d|k}])\rho^{n_y} - (\mathbf{P}\underset{\leftarrow}{\mathbf{u}}_{k-1} + \mathbf{Q}\underset{\leftarrow}{\hat{\mathbf{y}}}_k + d_k)}{h_{n_y}}, \tag{8}$$

where $h_{n_y} = \sum_{j=1}^{n_y} H(j)$ and $H(j)$ is the jth element of $\mathbf{H}$.

One of the core advantages of PFC over some of the similarly placed techniques, such as the PID, is its ability to integrate constraints within the design instead of treating them as an afterthought.[23] Owing to the assumption of constant future input, it is possible to implement a simple saturation policy to predict and validate the input constraint adherence using just the following four inequalities at each k:

$$\begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} u_k \leq \begin{bmatrix} \bar{u} \\ -\underline{u} \\ \Delta\bar{u} + u_{k-1} \\ -\Delta\bar{u} - u_{k-1} \end{bmatrix}. \tag{9}$$

Output/state constraints, if present, can also be implemented efficiently using model predictions,[7] such as (6), over a large validation horizon n_c, with $n_c >> n_y$, so that future violations (in nominal conditions) could be prevented. Given $\underline{y} \leq y_k \leq \bar{y}$, the following inequalities must be validated at each sample k with an input u_k selected closest to the one obtained via (8), such that:

$$\underline{y} \leq h_i u_k + \mathbf{P}_i \underset{\leftarrow}{\mathbf{u}}_{k-1} + \mathbf{Q}_i \underset{\leftarrow}{\hat{\mathbf{y}}}_k + d_k \leq \bar{y}, \tag{10}$$

where $i = 1, 2, \ldots, n_c$.

Remark 1. The process of constraint validation based on (9) and (10) guarantees nominal recursive feasibility (no change in the steady-state target and/or the disturbance), provided the open-loop system has stable and monotonically convergent dynamic behaviour.[7]

3.2 | Selecting parameters ρ and n_y

The primary tuning parameter ρ represents the ideal (first-order exponential) speed of convergence of the tracking error, that is, how fast or slow the predicted response approaches the set-point. Assuming $n_d = 0$ for simplicity, it is clear from (4) that the predicted n_y-step ahead tracking error $e_{k+n_y|k}$ is equal to ρ^{n_y} times the current error e_k, where $e_k = R - y_k$. While the significance of ρ is obvious, its efficacy is highly dependent on the judicious selection of n_y. In general, as n_y gets larger, the closed-loop performance tends to the open-loop behavior, albeit with zer steady-state offset, irrespective of the chosen target pole.[12] Clearly ρ has the maximum influence when $n_y = 1$, but in practice enforcing one-step ahead coincidence may not always be a good choice,[7] especially if the predicted response exhibits significant initial lag, as is the case with overdamped or nonminimum phase dynamics.

Notably one-step ahead coincidence is mostly effective, providing 100% target tracking in nominal conditions, for processes with dominant first-order behavior.[6,7] However, implementations with heavily damped dynamics generally necessitate coincidence further away in future; a requirement that reduces the efficacy of ρ to some extent. Nevertheless, to achieve a performance closer to the desired one, coincidence should be enforced as early as possible. In this context, one suggestion is to use the point of inflection, that is, the point of maximum gradient on the open-loop step response curve, as the coincidence point.[5] However, it is argued that tuning on this criterion alone may be flawed, especially if the dynamics in question are nonminimum phase.[12] Instead, a more sensible n_y lies within the time window when the step response rises from 40% to 80% of its steady-state value with significant gradient, and the first-order reference that coincides within this time window is a suitable target trajectory.[7]

3.3 | Performance limitations with challenging applications

It has traditionally been difficult to synthesize an effective control law for unstable and/or poorly damped dynamic processes using a low cost approach, such as PID.[24] The simplistic design attributes mean that conventional PFC too struggles and performs rather poorly in these applications as reported in many recent studies.[4,7,12,13] Researchers mainly link this inefficacy to the constant future input assumption[4,7,12] which, although works well when the open-loop predictions are stable and monotonically convergent to the implied steady state, is clearly inappropriate in view of the challenging dynamic characteristics. This results in a large inconsistency between the predicted and the actual behavior, embedding unreliability in the decision-making. It is further noted that:

- With difficult dynamics, the selection of tuning parameters ρ and n_y is far less clear cut, since the available guidelines mainly rely on the analysis of open-loop step response which clearly becomes meaningless in the presence of large oscillations/divergence.
- Recursive feasibility under constraints cannot be guaranteed even nominally, as the continued use of previous input inevitably leads to constraint violation due to oscillations/divergence.

Although the design may still work in some cases due to the receding horizon,[12] it is indeed unreliable and prone to failure especially with uncertainties or tight actuation limits. To tackle this deficiency arising due to the use of constant future input within predictions, an obvious solution is to implement a more flexible parametrization of the input function (see for instance References 4,13). In the current proposal, reparametrization of the degree-of-freedom is achieved via prestabilization of the difficult open-loop dynamics, which is a well-established concept adapted from the mainstream MPC literature.[16,17] The following sections present the proposal in detail.

4 | PRESTABILIZED PFC FRAMEWORK

This section presents the concept of pre-stabilisation in the context of PFC and proposes a systematic design framework, based on the initial proposal,[21] to cater for a variety of difficult open-loop dynamics.

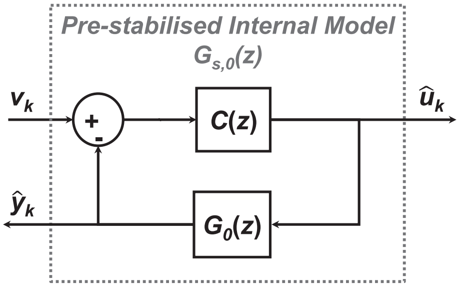


FIGURE 2 Precompensation of internal prediction model $G_0(z)$

4.1 | Establishing the PPFC control law

The fundamental idea behind PPFC is to first stabilize the undesirable open-loop dynamics, using a simple and well understood classical approach, and then implement PFC in the standard way, as an outer loop, for improving performance, and managing constraints and deadtimes. The precompensation loop is generally implemented on the internal model (e.g., see References 18-20) thereby utilizing the internal input as the main decision variable for plant control. This approach, however, is not recommended, especially with open-loop unstable dynamics, because closed-loop stability cannot be ensured as even the smallest amount of numerical precision error would trigger a divergent response from the unstable plant. A particular novelty of this work is separately closing the loop on the plant and the model so implicitly they do not share the same input signal.

In the current proposal, the delay-free prediction model $G_0(z)$ is prestabilized using a classical controller $C(z)$ in the feedback configuration shown in Figure 2, where $C(z) = q(z)/p(z)$ with $p(z) = 1 + p_1 z^{-1} + \cdots + p_m z^{-m}$ and $q(z) = q_0 + q_1 z^{-1} + \cdots + q_m z^{-m}$. It means that a compensated prediction model $G_{s,0}(z)$, with stable and monotonically convergent dynamics, given by:

$$G_{s,0}(z) = \frac{C(z)G_0(z)}{1 + C(z)G_0(z)} = \frac{\beta(z)}{\alpha(z)}, \tag{11}$$

is now implemented for decision-making, where $\beta(z) = \beta_1 z^{-1} + \beta_2 z^{-2} + \cdots + \beta_l z^{-l}$, $\alpha(z) = \alpha_0 + \alpha_1 z^{-1} + \alpha_2 z^{-2} + \cdots + \alpha_l z^{-l}$, and $l = m + n$. The PPFC control law is derived in the conventional way, albeit using the closed-loop prediction model $\alpha(z)\hat{y}_k = \beta(z)v_k$, as follows:

$$y_{k+n_y|k} = \mathbf{H}\underset{\rightarrow k}{\mathbf{v}} + \mathbf{P}\underset{\leftarrow k-1}{\mathbf{v}} + \mathbf{Q}\underset{\leftarrow k}{\hat{\mathbf{y}}} + d_k, \tag{12}$$

where $\mathbf{P}$, $\mathbf{Q}$, and $\mathbf{H}$ depend upon the parameters of the prestabilized model (11). The control law takes the form:

$$v_k = \frac{R - (R - E[y_{k+n_d|k}])\rho^{n_y} - (\mathbf{P}\underset{\leftarrow k-1}{\mathbf{v}} + \mathbf{Q}\underset{\leftarrow k}{\hat{\mathbf{y}}} + d_k)}{h_{n_y}}. \tag{13}$$

This, however, also transforms the decision variable from u_k to v_k, with direct implications for parameter tuning and constraint handling.

4.2 | Evaluating the main decision variable u_k

Although the PPFC computes v_k at each sample, evaluating u_k is necessary for plant actuation. However, the implied relationship between u_k and v_k is not straightforward owing to the separate closure of plant and model loops. The inner model input $\hat{u}_k$, nonetheless, is directly linked to v_k, independent of the fine details pertaining to the internal feedback loop design.

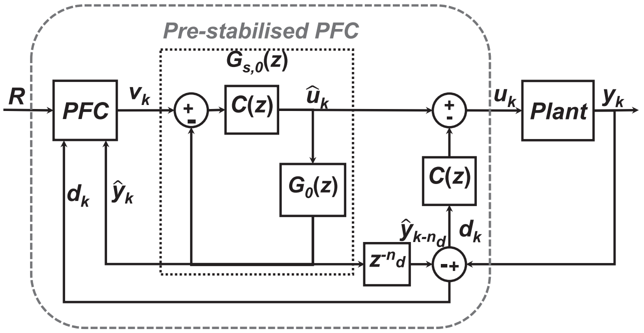


FIGURE 3 Proposed prestabilized predictive functional control control architecture

Lemma 1. *The control variables $\hat{u}_k$ and v_k after pre-stabilisation are related as follows:*

$$\hat{u}_k = q(z).\frac{a(z)}{\alpha(z)}v_k. \tag{14}$$

Proof. This is obvious from the expressions $\hat{y}_k = G_0(z)\hat{u}_k = G_{s,0}(z)v_k$. Eliminating $\hat{y}(k)$ results in:

$$\frac{b(z)}{a(z)}\hat{u}_k = \frac{\beta(z)}{\alpha(z)}v_k = \frac{q(z)b(z)}{\alpha(z)}v_k,$$

which simplifies to (14). ∎

Remark 2. The reader is reminded that the inner loop with G_0 is a simulation or internal model and thus the algebra is exact and with no uncertainty.

The next step is to compute u_k, which in principle could be obtained directly from the loop structure if prestabilization were hardwired. Given that $C(z)$ is designed using the delay-free model $G_0(z)$, utilising it in conjunction with the time-delayed plant may not yield desirable performance. A unique contribution of the proposal is summarized by Theorem 1, which establishes a key relationship between already known quantities to obtain u_k indirectly without hardwiring the prestabilising compensator.

Theorem 1. *Prestabilizing the plant separately with $C(z)$, in addition to the model $G_0(z)$, is equivalent to computing the control input u_k using the following expression:*

$$u_k = \hat{u}_k - C(z)d_k. \tag{15}$$

Proof. Assuming $C(z)$ stabilises both the plant and the model separately, one gets $u_k = C(z)[v_k - (\hat{y}_k + d_k)]$ and $\hat{u}_k = C(z)[v_k - \hat{y}_k]$ for both pre-stabilisation loops, respectively. Eliminating v_k then provides:

$$u_k + C(z)\hat{y}_k + C(z)d_k = \hat{u}_k + C(z)\hat{y}_k,$$

which simplifies to (15). Since both $\hat{u}_k$ and d_k are known, u_k can be computed in effect without hardwiring the compensator in practice. The resulting control architecture is depicted by the block diagram in Figure 3. ∎

Remark 3. In nominal conditions, that is, without modeling mismatch and external disturbances, $u_k = \hat{u}_k$.

Corollary 1. *The obvious corollary of Lemma 1 and Theorem 1 is that the decision variable$\hat{u}_k$ is redundant after pre-stabilisation and can be omitted from computation, which means the model is excited with v_k whereas the plant with u_k.*

Hence, replacing $\hat{u}_k$ in (15) from (14) results in a direct relationship between the decision variables of interest:

$$u_k = q(z).\frac{a(z)}{\alpha(z)}v_k - C(z)d_k, \tag{16}$$

which can be rewritten as:

$$A(z)u_k = B(z)v_k + E(z)d_k, \tag{17}$$

with the polynomials $A(z)$, $B(z)$, and $E(z)$ defined as follows:

$$\begin{aligned} A(z) &= \alpha(z)p(z) = 1 + A_1z^{-1} + A_2z^{-2} + \cdots \\ B(z) &= q(z)a(z)p(z) = B_0 + B_1z^{-1} + B_2z^{-2} + \cdots \\ E(z) &= -\alpha(z)q(z) = E_0 + E_1z^{-1} + E_2z^{-2} + \cdots \ . \end{aligned} \tag{18}$$

At each time sample, the plant input u_k can be computed directly from v_k and vice versa using:

$$u_k = B_0v_k + f_k; \quad f_k = -\mathbf{A}\underleftarrow{\mathbf{u}}_{k-1} + \mathbf{B}\underleftarrow{\mathbf{v}}_{k-1} + \mathbf{E}\underleftarrow{\mathbf{d}}_{k}, \tag{19}$$

where vectors **A**, **B**, and **E** contain appropriate coefficients of the respective polynomials. The main advantage of the proposal is visible clearly since u_k is now reparamerized as a linear combination of a constant term v_k (obtained from the outer PFC loop) and a time-varying term f_k (obtained from the inner loop configuration), which can now handle nonsimple dynamics with ease and efficiency.

Remark 4. The computational requirement of (19) is similar to the open-loop control law (8), but owing to reparametrization of u_k, constraint handling is now expected to be slightly more onerous. Nevertheless, the underlying coding is still elementary; for instance, vector multiplication can be programmed in few lines with the basic loop instruction.

4.3 | Analysing the initial input activity

The dynamics of the initial input produced by the controller is an important metric to assess the expected closed-loop performance, as it provides valuable insights about the implied transient behavior of the controlled system. Assuming zero initial conditions and no uncertainty for simplicity, it is straightforward to show using (13) and (19) that for a change in R:

$$u_{1,n_y} = \frac{B_0R}{h_{n_y}}(1 - \rho^{n_y}), \tag{20}$$

where u_{1,n_y} is the initial input for the chosen n_y. It is noted that:

- The initial input is directly proportional to the magnitude of the desired set-point. This is expected since tracking a large target change usually requires a correspondingly aggressive control action.
- h_{n_y}, which is computed from the model parameters based on the selected coincidence horizon, inversely affects u_{1,n_y}.
- For smaller values of n_y, the initial input is inversely related to the term ρ^{n_y}, which means a faster target pole (smaller ρ) requires an aggressive initial control and vice versa. Note that large n_y values make ρ^{n_y} insignificant.

Two instances of particular interest are when either one-step ahead coincidence ($n_y = 1$) is enforced or when n_y is chosen so large (theoretically approaching ∞) that $\rho^{n_y} \to 0$; knowing the initial input activity for both cases can provide a better understanding of the expected closed-loop performance for various possible choices of ρ and n_y.

Theorem 2. *For a given set-point R and a target pole ρ, the initial control for $n_y = 1$ and $n_y \to \infty$ is given by:*

$$u_{1,n_y} = \begin{cases} \dfrac{B_0R}{\beta_1}(1 - \rho); & n_y = 1 \\ \dfrac{B_0R}{G_{s,0}(1)}; & n_y \to \infty, \end{cases}$$

where β_1 is the lead coefficient of $\beta(z)$, and $G_{s,0}(1)$ is the steady-state gain of the pre-stabilised system.

Algorithm 1. Selecting ρ and n_y

With multiple target poles such that $0<\rho_i<\rho_{i-1}<\cdots<\rho_1\leq z_s$, where z_s is the slowest (dominant) pole of the pre-stabilised prediction model, plot (20) over a long enough range of n_y, preferably up to one time constant (i.e. the time required to reach approximately 63% of the implied steady-state response). Select a combination of ρ and n_y which gives $u_{1,n_y}\approx\theta u_{1,\infty}$, where θ is the amplification factor roughly chosen within $2\leq\theta\leq 5$.

Proof. The one-step ahead prediction ($n_y=1$) obtained from the prestabilized model $\alpha(z)\hat{y}_k=\beta(z)v_k$ can be written as:

$$\hat{y}_{k+1}+\alpha_1\hat{y}_k+\alpha_2\hat{y}_{k-1}+\alpha_3\hat{y}_{k-2}+\cdots=\beta_1 v_k+\beta_2 v_{k-1}+\beta_3 v_{k-2}+\ \ldots$$

which can be rearranged in the vector form:

$$\hat{y}_{k+1}=\beta_1 v_k+[\beta_2\quad \beta_3\ \ldots]\underset{\leftarrow k-1}{\mathbf{v}}+[-\alpha_1\quad -\alpha_2\quad -\alpha_3\ \ldots]\underset{\leftarrow k}{\hat{\mathbf{y}}},$$

from which it is clear that $h_1=\beta_1$. Hence, using (20):

$$u_{1,1}=\frac{B_0R}{\beta_1}(1-\rho);\quad n_y=1. \tag{21}$$

When $n_y\rightarrow\infty$, it is known from a previous study[12] that h_{n_y} approaches the static gain of the system. Therefore, (20) reduces to:

$$u_{1,\infty}=\frac{B_0R}{G_{s,0}(1)};\quad n_y\rightarrow\infty. \tag{22}$$

■

Note that (22) represents a special approach implementing the so called *mean-level* tuning in which one implicitly accepts the open-loop (in this case the prestabilized) transient behavior in the closed-loop performance.[7] In practice, this can be achieved by selecting the degree-of-freedom $v_k=v_{ss}=\dfrac{R}{G_{s,0}(1)}$, where v_{ss} denotes the expected stead-state input. Notwithstanding the lack of mathematical rigour, a sensible choice of parameters could then be the one that simply amplifies $u_{1,\infty}$ by a reasonable amount, such that the resulting initial control is not too aggressive, that is, practically achievable.

Remark 5. Although prestabilization allows one to implement conventional tuning methods discussed in Section 3.2, see for instance References 21,22, a key contribution here is the development of Algorithm 1, which utilizes transient input activity for a more meaningful and performance oriented selection of ρ and n_y (as shown in Figure 5). However, direct implementation of (20) with the complicated open-loop dynamics should be avoided as parameter tuning based on unreliable, that is, numerically infeasible, computations of h_{n_y} could possibly lead to ill-posed decision-making.[7]

4.4 | Summary

To sum up, the concept of PPFC works *systematically* in three simple steps: forming stable and well-damped closed-loop predictions using a classical feedback compensator, implementing PFC using the prestabilized model, and analyzing the predicted initial input for a meaningful parameter selection. The proposal is independent of the underlying open-loop characteristics, and therefore could be applied to a variety of processes including those exhibiting instability and/or poor damping. The next section will discuss two simple methods to design the inner controller for such applications, followed by a brief analysis of the impact of prestabilization on constraint management.

5 | DESIGN OF PRESTABILIZING COMPENSATOR

So far we have examined the impact of prestabilization on the core functionality of PFC by assuming a suitable compensator that stabilizes the undesirable open-loop dynamics for consistent prediction behaviour. In this section, two common methods of classical feedback control are proposed for this purpose, namely: proportional (plus derivative), and pole placement designs. It is noted that both alternatives are well-understood and easily implementable with basic technical know-how. Hence, the proposal assures a cheap and sustainable loop design combined with the fundamental benefits of predictive control.

5.1 | Compensation via P/PD controller

The proportional plus integral plus derivative (PID) compensation is arguably the most popular industrial process controller owing to cheap and straightforward implementation and maintenance thereafter. Hence, it makes intuitive sense to utilize the benefits of such a universal technique to further enhance the capabilities of PFC, which was originally developed to compete with PID in cost and performance. The idea here is to tune the proportional (plus derivative) part only, using any standard time-domain or frequency-domain PID tuning method, to prestabilize the difficult dynamics before implementing PFC. It is noted that for a majority of first and second order processes, a simple P- or PD-type controller can satisfactorily prestabilize the undesirable dynamics. Nevertheless, there are instances like poorly damped or difficult higher-order poles which may require a slightly more sophisticated approach such as the one based on pole placement.

5.2 | Compensation via pole placement

The main idea behind pole placement is to design the controller by specifying the desired prestabilized pole configuration. It is noted that pole placement generally results in higher-order controllers, which in the context of PPFC may slightly increase the burden of constraint management, but this is an inevitable consequence when simpler alternatives are no longer effective.

The current pole placement proposal is based on the analytical approach of feedback compensation presented in reference.[21] Assume that a $(n-1)$th-order bi-proper compensator $C(z)$ is used to modify the open-loop model $G_0(z)$, as shown in Figure 3, resulting in the prestabilized transfer function $G_{s,0}(z)$, with a smooth and monotonically convergent prediction behavior. Then one may write:

$$G_{s,0}(z) = \frac{\beta(z)}{\alpha(z)} = \frac{q(z)b(z)}{p(z)a(z) + q(z)b(z)}, \tag{23}$$

where $\alpha(z)$ is the $(2n-1)$th-order prestabilized pole polynomial, and the underlying relationship,

$$p(z)a(z) + q(z)b(z) = \alpha(z), \tag{24}$$

is called the *Diophantine Equation*. In order to design the $C(z)$, one must define the desired pre-stabilised characteristic polynomial $\alpha(z)$ and then utilize linear algebra to obtain the coefficients of $p(z)$ and $q(z)$ with,

$$\mathbf{M} = \mathbf{S}^{-1}\mathbf{D}, \tag{25}$$

where $\mathbf{M} = [p_{n-1} \cdots p_0 \quad q_{n-1} \cdots q_0]^T$, $\mathbf{D} = [\alpha_{2n-1} \cdots \alpha_0]^T$ and $\mathbf{S}$ is the *Sylvester Matrix*[25] given by:

$$\mathbf{S} = \begin{bmatrix} a_n & 0 & \dots & 0 & b_n & 0 & \dots & 0 \\ a_{n-1} & a_n & \dots & 0 & b_{n-1} & b_n & \dots & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & a_1 & \dots & a_{n-1} & 0 & b_1 & \dots & b_{n-1} \\ 0 & 1 & \dots & a_{n-2} & 0 & 0 & \dots & b_{n-2} \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & a_1 & 0 & 0 & \dots & b_1 \\ 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 \end{bmatrix}. \tag{26}$$

Algorithm 2. Unconstrained PPFC

```
I.   First stabilize the open-loop dynamics using, for example, the P(D) or Pole Placement methods discussed above.
II.  Select appropriate tuning parameters $\rho$ and $n_y$ using the the proposed Algorithm 1, or indeed the standard tuning
     guidelines, discussed in Section 3.2.
III. At each sample $k$, compute the unconstrained values of $v_k$ using (13).
IV.  Finally, compute the unconstrained value of $u_k$ with (19), and update the plant and the model.
```

Note that $\alpha(z)$ is factorized as:

$$\alpha(z) = o(z)a^{-}(z)\alpha^{+}(z), \tag{27}$$

where $o(z)$ is the $(n-1)$th-order observer generally selected as $o(z) = z^{n-1}$, $a^{-}(z)$ factors the stable open-loop poles and $\alpha^{+}(z)$ represents the p_u prestabilized poles. If $a^{+}(z) = \prod_{i=1}^{p_u}(z - z_{p,i})$ then: *Proposal for unstable poles*. With $z_{p,i} > 1$, design $\alpha^{+}(z) = \prod_{i=1}^{p_u}(z - 1/z_{p,i})$. In case an integrator factor $(z-1)$ is present, then one may simply replace it with $(z-0.5)$.[4] *Proposal for complex poles*. With $z_{p,i} \in \mathbb{C}$, place the prestabilized poles at the real part of the complex open-loop poles, that is, $\alpha^{+}(z) = \prod_{i=1}^{p_u}(z - \Re(z_{p,i}))$. This will effectively filter out the undesirable oscillations but without compromising the convergence speed.

This completes the internal feedback loop design via pole placement.

5.3 | Summary

This section has proposed two very simple and straightforward approaches of prestabilization. While the standard P/PD controllers are generally sufficient, one may also utilize pole placement for more involved open-loop dynamics, for which the proposed design steps are fairly elementary.

We are now in a position to sum up the discussion of unconstrained PPFC with the following algorithm (Algorithm 2):

6 | CONSTRAINT HANDLING WITH PRESTABILIZED DYNAMICS

For completeness, this section summarizes how constraint handling can be done in a very efficient manner for PFC where there is only a single degree-of-freedom.

The addition of an internal feedback loop reparameterizes the input function which implies that u_k is no longer constant within the prediction horizon. This directly affects the way input and rate constraints are handled, as one now has to observe constraint adherence at each future sample over a validation window extending well beyond the coincidence point. This is crucial because any unobserved input violation could eventually lead to infeasibility, invalidating the current optimization. Interestingly though, transforming the degree-of-freedom does not change the procedure to verify output/state constraints. Specifically, the standard methods, such as the one discussed in Section 3.1, remain valid, the only change being the use of prestabilized model predictions in the corresponding inequality (10). Taking all this into account, each row of the following vector inequalities restricts the ith prediction such that:

$$\begin{aligned} \mathbf{L}\underline{u} &\leq \underset{\rightarrow k}{\mathbf{u}} \leq \mathbf{L}\bar{u} \\ \mathbf{L}\Delta\underline{u} &\leq \Delta\underset{\rightarrow k}{\mathbf{u}} \leq \mathbf{L}\Delta\bar{u} \\ \mathbf{L}\underline{y} &\leq \underset{\rightarrow k+1}{\mathbf{y}} \leq \mathbf{L}\bar{y}, \end{aligned} \tag{28}$$

where $i = 0, 1, \dots, n_c$ and $\mathbf{L} = [1 \quad 1 \ \dots]^T$. Ideally, the validation horizon n_c should cover the settling period of $G_{s,0}(z)$; for example, the time to reach and stay within about 95% of the implied steady-state is often sufficient. It is more convenient to represent the constraint inequalities in terms of v_k as this value remains constant along n_c, by noting that $\underset{\rightarrow k}{\mathbf{u}} = B_0\mathbf{L}v_k + \underset{\rightarrow k}{\mathbf{f}}$,

Algorithm 3. Constrained PPFC

At each sample k, execute Step III of Algorithm 2 and update $\mathbf{f}_{\rightarrow k}$. Verify each row of (29), enforcing saturation at $v_k = Y^j/X^j$ for every violation in the jth row. Finally, compute the constraint adhering value of u_k using (19).

$\Delta\mathbf{u}_{\rightarrow k} = \mathbf{C}_{1/\Delta}^{-1}(\mathbf{u}_{\rightarrow k} - \mathbf{L}u_{k-1})$, and $\mathbf{y}_{\rightarrow k+1} = h_i\mathbf{L}v_k + \mathbf{P}\mathbf{v}_{\leftarrow k-1} + \mathbf{Q}\hat{\mathbf{y}}_{\leftarrow k} + \mathbf{L}d_k$:[7]

$$\underbrace{\begin{bmatrix} B_0\mathbf{L} \\ -B_0\mathbf{L} \\ B_0\mathbf{C}_{1/\Delta}^{-1}\mathbf{L} \\ -B_0\mathbf{C}_{1/\Delta}^{-1}\mathbf{L} \\ h_i\mathbf{L} \\ -h_i\mathbf{L} \end{bmatrix}}_{\mathbf{X}} v_k \leq \underbrace{\begin{bmatrix} \mathbf{L}\bar{u} - \mathbf{f}_{\rightarrow k} \\ -\mathbf{L}\underline{u} + \mathbf{f}_{\rightarrow k} \\ \mathbf{L}\Delta\bar{u} - \mathbf{C}_{1/\Delta}^{-1}\mathbf{f}_{\rightarrow k} + \mathbf{C}_{1/\Delta}^{-1}\mathbf{L}u_{k-1} \\ -\mathbf{L}\Delta\underline{u} + \mathbf{C}_{1/\Delta}^{-1}\mathbf{f}_{\rightarrow k} - \mathbf{C}_{1/\Delta}^{-1}\mathbf{L}u_{k-1} \\ \mathbf{L}\bar{y} - \mathbf{P}\mathbf{v}_{\leftarrow k-1} - \mathbf{Q}\hat{\mathbf{y}}_{\leftarrow k} - \mathbf{L}d_k \\ -\mathbf{L}\underline{y} + \mathbf{P}\mathbf{v}_{\leftarrow k-1} + \mathbf{Q}\hat{\mathbf{y}}_{\leftarrow k} + \mathbf{L}d_k \end{bmatrix}}_{\mathbf{Y}}, \tag{29}$$

where $\mathbf{C}_{1/\Delta}$ is a lower triangular matrix defined as follows (Algorithm 3)[7]

$$\mathbf{C}_{1/\Delta} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix}. \tag{30}$$

Theorem 3. *Algorithm 3 guarantees recursive feasibility in the presence of constraints, provided the target set-point and disturbance remain unchanged.*

Proof. First it is noted that the long-range predictions after pre-stabilisation will be stable and convergent with a constant input $v_{k+i} = v_k,\ \forall i > 0$. Next, if one assumes feasibility at the start (i.e., with a reasonable set-point and initial conditions[7]), then at every subsequent sample, the choice $v_k = v_{k-1}$ will always satisfy constraints and hence will always be feasible. ■

Conversely it is worth emphasising that feasibility cannot be guaranteed with the direct implementation of open-loop dynamics, as the recursive use of a previous input would eventually result in oscillations/divergence and therefore unavoidable constraint violations.

Remark 6. Although recursive feasibility is established in principle for the nominal case, the underlying assumption, that is, a constant target and/or disturbance, is indeed somewhat conservative. For example, only small target/disturbance changes may be permissible in practice, since a large change is highly likely to cause infeasibility. A common approach adopted in the mainstream MPC literature to furnish rigorous recursive feasibility properties in more realistic scenarios is to employ some relatively costly computations involving, for instance, reference governing,[26] min-max synthesis,[27] or tubes,[28] which if utilized in conjunction with a technique as inexpensive as PFC would not only undermine its simplicity but also escalate its price range considerably. Arguably, the lack of concrete feasibility results could be mitigated to some extent by following sensible guidelines, such as using large enough validation horizons, specifying attainable control objectives etc., as is usually the case with many industrial process control algorithms incorporating constraints.[7]

7 | SIMULATION CASE STUDIES

In this section, we investigate the efficacy of the proposed PPFC algorithm alongside the conventional PFC and the PI(D) controllers in practical scenarios with two real world case studies. The first system G_1 is underdamped, whereas the second model G_2 is an open-loop unstable process. Detailed discussion is presented in the following sections.

7.1 | Description of case studies

7.1.1 | Thermoacoustic oscillations in a combustion process

A typical continuous combustion process in gas turbines or high-speed propulsion engines involves burning a fuel–air mixture for thrust production. Under the right conditions, the process also generates audible pressure waves, which are potentially hazardous for structures and personnel.[29] The underlying thermoacoustic phenomenon is complex and nonlinear; nevertheless, a simplified laboratory apparatus, known as the Rijke Tube, which demonstrates similar dynamic characteristics, is generally used for the design and analysis of feedback controllers in a straightforward manner.[30] Figure 4A shows a Rijke tube combustion apparatus consisting of a glass cavity with burner, pressure sensor, and diaphragm actuator. In this setup, the actuator movement produces additional waves that interact with the thermoacoustics to damp down the audible oscillations. The linearized second-order model, ignoring the sensor and actuator dynamics, is given by:

$$G_1 = \frac{y(z)}{u(z)} = \frac{10.66z + 10.54}{z^2 - 1.543z + 0.9671}, \tag{31}$$

where y is the measured pressure (Pa) and u is the diaphragm velocity (m/s), subject to physical limits: $|\Delta u| \leq 0.015$ m/s and $|y| \leq 4.5$ Pa. In the open-loop configuration, the primary pressure wave oscillates at 142 Hz with an exponentially decaying humming sound, at the steady-state operating point $y_{ss} = 50$ Pa and $u_{ss} = 1$ m/s.

7.1.2 | Temperature control in Jacketed continuous stirred tank reactor

The Continuous Stirred Tank Reactor (CSTR) is a common industrial unit widely employed in different chemical manufacturing processes. The reaction dynamics converting component A into component B in an ideal CSTR has a nonlinear first-order behavior. Nevertheless, many chemical reactions require a specific temperature to be maintained within the tank for flawless yield. Therefore, the tank is generally equipped with an outer jacket in which

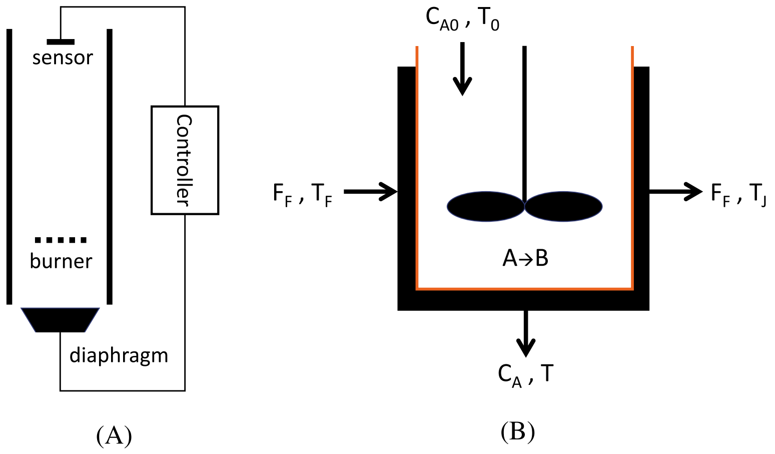


FIGURE 4 (A) Rijke tube apparatus, (B) Jacketed CSTR process

the temperature of a flowing fluid T_J is used to regulate the inside reaction temperature T, as shown in Figure 4B. The overall coupled model has two-state nonlinear dynamics with potential for exotic behavior owing to multiple steady-states.[31] In this study, the linearized model around the operating point depicts unstable second-order dynamics given by[32]

$$G_2 = \frac{T(z)}{T_J(z)} = \frac{0.00895z - 0.008249}{z^2 - 1.972z + 0.9719} z^{-25}, \tag{32}$$

subject to $|T_J| \leq 2.1°F$. Note that both T and T_J are deviation variables around the steady-state values $T_{ss} = 560.8°F$ and $T_{J,ss} = 2637.9°F$.

7.2 | Preconditioning of open-loop dynamics

We will first prestabilize the prediction models G_1 and G_2 using the two proposed methods, namely Proportional (plus Derivative) and Pole Placement, respectively. A P(D) compensator can be tuned easily with the standard tuning methods. Here, the robust PID tuning algorithm available in the MATLAB environment (see Reference 33 for details) will be utilized.

For G_1, a simple proportional gain, with or without the derivative action, fails to sufficiently damp the output oscillations. Consequently a pole placement compensator will be designed by placing the desired poles at the real part of the open-loop complex poles ($z = 0.7715$). The resulting compensator

$$C_1^{PP} = \frac{-0.00937z - 0.00972}{z + 0.106}, \tag{33}$$

therefore provides the following prestabilized transfer function model

$$G_{s_1}^{PP} = \frac{-0.106z^2 - 0.2084z - 0.1025}{z(z^2 - 1.543z + 0.5951)}, \tag{34}$$

with the now stable poles residing at 0, 0.7715, 0.7715. Note that the additional pole at $z = 0$ here represents the minimum-order observer dynamics (refer to Section 5.2 for the detailed design steps). For G_2, a P compensator can comfortably stabilize the open-loop dynamics, with $C_2^P = 0.502$ providing

$$G_{s_2}^{P} = \frac{0.004482z^2 - 0.004137}{z^2 - 1.968z + 0.9678}, \tag{35}$$

having poles at 0.9784, 0.9892. This completes the offline prestabilization step in a straightforward manner.

7.3 | Analysis of tuning efficacy

This section demonstrates the power of the proposed approach in this paper. Because the inner loop has better conditioned behaviour, now an intuitive PFC tuning procedure is straightforward, which is not the case with the original dynamics. Using Algorithm 1, Figure 5 analyzes the initial input as a function of n_y for both G_1 (Figure 5A) and G_2 (Figure 5B) for various possible choices of the target pole. It is evident that:

- Depending on the prediction dynamics, $n_y = 1$ may or may not be a suitable choice. For example, it may work with G_2 but for G_1 it would produce an overactive control, even with the slowest target pole.
- The target pole ρ loses efficacy beyond the system's time constant (approximately after 8 and 130 samples for both G_1 and G_2, respectively), with the initial input nearly approaching $u_{1,\infty}$.
- It is possible to obtain similar initial control with different pairings of (ρ, n_y). Faster target poles, however, tend to intercept the $\theta u_{1,\infty}$ horizontal line at longer coincidence points, suggesting a weaker link between the target and the actual response.

In order to assess the tuning efficacy, we select two distinct parameter pairs from Figure 5 which provide similar initial inputs. For G_1: $\rho_1 = 0.7215$, $n_{y_1} = 4$ and $\rho_2 = 0.6715$, $n_{y_2} = 5$, and for G_2: $\rho_1 = 0.9767$, $n_{y_1} = 27$ and $\rho_2 = 0.9517$, $n_{y_2} = 47$. The results are shown in Figure 6. Evidently tuning with faster pole but longer coincidence generally provides comparatively quicker transition to the set point than using a slower target pole with smaller n_y, despite a similar initial control effort. Table 1 tabulates the resulting RMS error values with the selected parameter choices. Expectedly the true performance with large coincidence points converges quickly to the set point (smaller rms$[R - y_k]$), but weakly linked to the associated reference trajectory (bigger rms$[r_k - y_k]$). This, in practice, should not be an issue as long as a sensible n_y is selected, that is, the one that does not undermine the desirable effect of the faster target pole.

7.4 | Effect of uncertainties on the expected closed-loop performance

We analyze the tuning efficacy in the presence of external disturbances, measurement noise and modeling mismatches. For the underdamped process G_1, a −5% constant disturbance is introduced at the process output around 65 ms, whereas

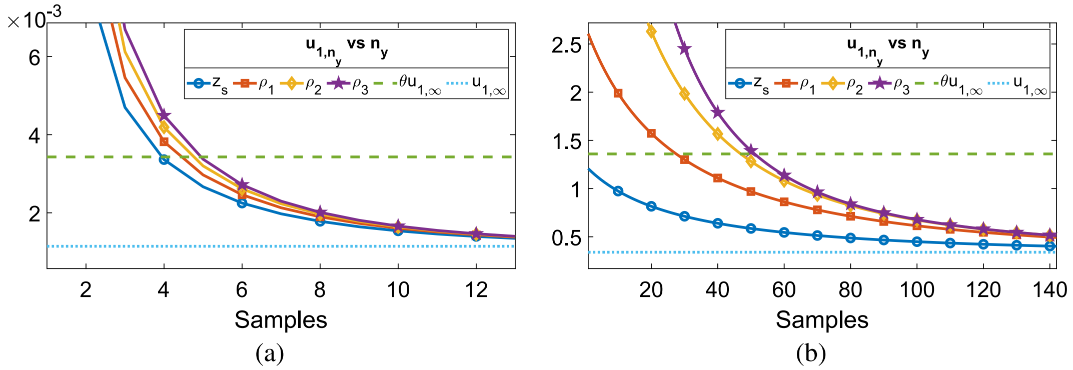


FIGURE 5 Initial input activity for (A) G_1 with $z_s = 0.7715$, $\rho_1 = 0.7215$, $\rho_2 = 0.6715$, $\rho_3 = 0.6215$, $u_{1,\infty} = 0.00124$, $\theta = 3$ and $R = 1$ (B) G_2 with $z_s = 0.9892$, $\rho_1 = 0.9767$, $\rho_2 = 0.9517$, $\rho_3 = 0.9017$, $u_{1,\infty} = 0.3398$, $\theta = 4$ and $R = 1$

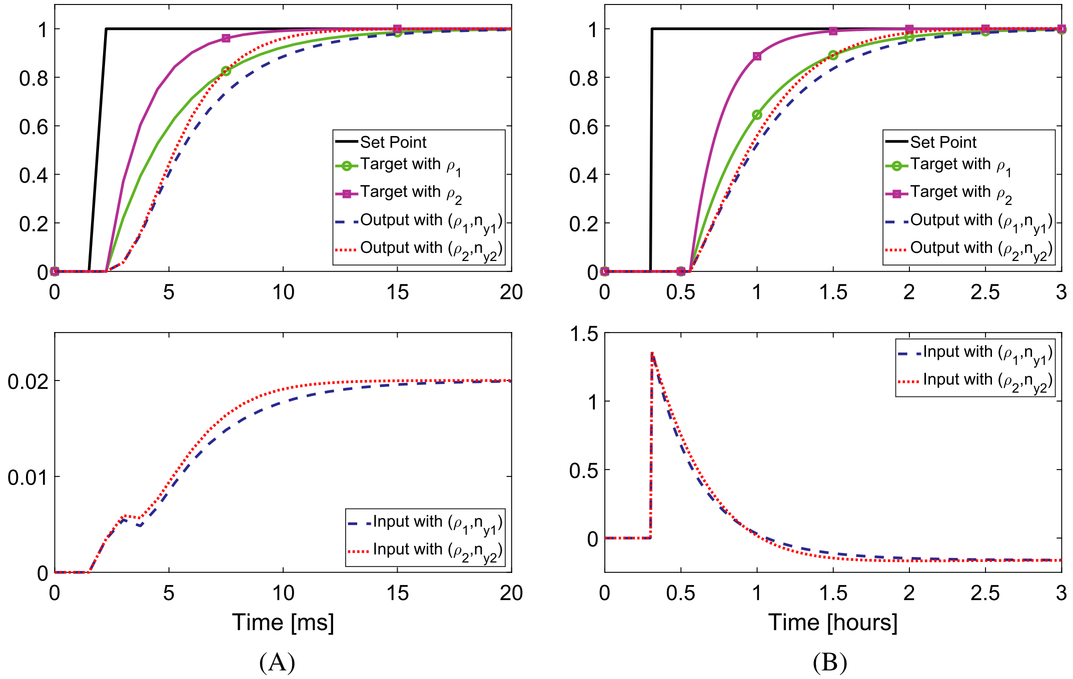
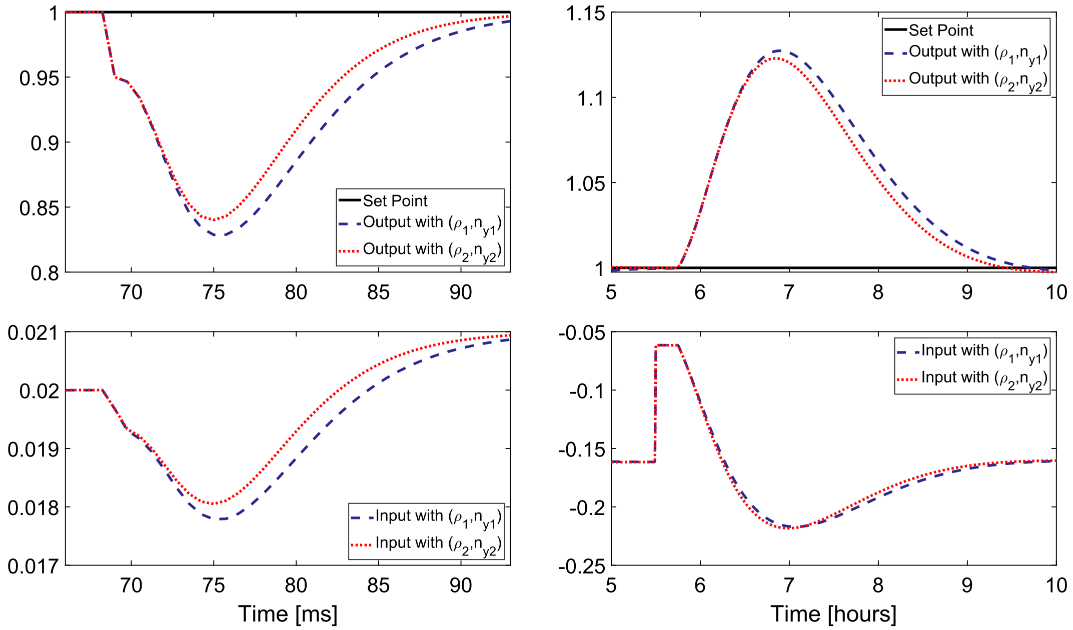


FIGURE 6 Analysis of tuning efficacy with the chosen (ρ, n_y) pairs for (A) G_1 with (0.7215,4) and (0.6715,5), (B) G_2 with (0.9767,27) and (0.9517,47)

TABLE 1 RMS of $(r_k - y_k)$ and $(R - y_k)$ for G_1 and G_2 with the selected tuning parameters

			$\mathbf{rms}[r_k - y_k]$	$\mathbf{rms}[R - y_k]$
G_1	$\rho_1 = 0.7215$	$n_{y_1} = 4$	0.0527	0.2194
	$\rho_2 = 0.6715$	$n_{y_2} = 5$	0.0958	0.2103
G_2	$\rho_1 = 0.9767$	$n_{y_1} = 27$	0.0374	0.2792
	$\rho_2 = 0.9517$	$n_{y_2} = 47$	0.0857	0.2695



(A) (B)

FIGURE 7 Comparison of disturbance rejection with both tuning choices for (A) G_1 with −5% output disturbance, and (B) G_2 with 10% input disturbance

for G_2 a 10% constant input disturbance is introduced around the mid of the fifth hour. The results, shown in Figure 7A,B respectively, suggest a comparatively quicker disturbance rejection with the faster target pole in both examples. Similarly, as shown in Figure 8A,B, the closed-loop performances with the selection (ρ_2, n_{y_2}) appears to be slightly more affected by the modeling errors (unmodeled pole at $z = 0.25$ for G_1, and approximately 10% multiplicative uncertainty for G_2). Interestingly, both performances appear indistinguishable (Figure 8) with respect to the measurement noise.

7.5 | Comparison of constrained closed-loop performance against CPFC and PID

Finally, a comparative analysis of the constrained closed-loop performance against the conventional PFC (CPFC) and PID algorithms is presented. The PPFC controller, in both examples, is tuned with the faster pole selection (ρ_2, n_{y_2}). For a fair comparison, the CPFC controller also uses these parameters, albeit with the difficult open-loop prediction dynamics given in (31) and (32), respectively. Furthermore, the PI(D) controller is synthesized using the robust PID tuning algorithm available in MATLAB.[33] The actual nonlinear models of G_1 and G_2 act as the *plant* for a more realistic evaluation, with the results shown in Figure 9. Here, PPFC-P and PPFC-PP refer to proportional and pole placement precompensation, respectively.

Figure 9A depicts the scenario for the poorly damped process, where a set point change of 5 Pa from the initial steady-state is introduced. As evident, the PI controller fails completely, destabilising under constraints. The CPFC,

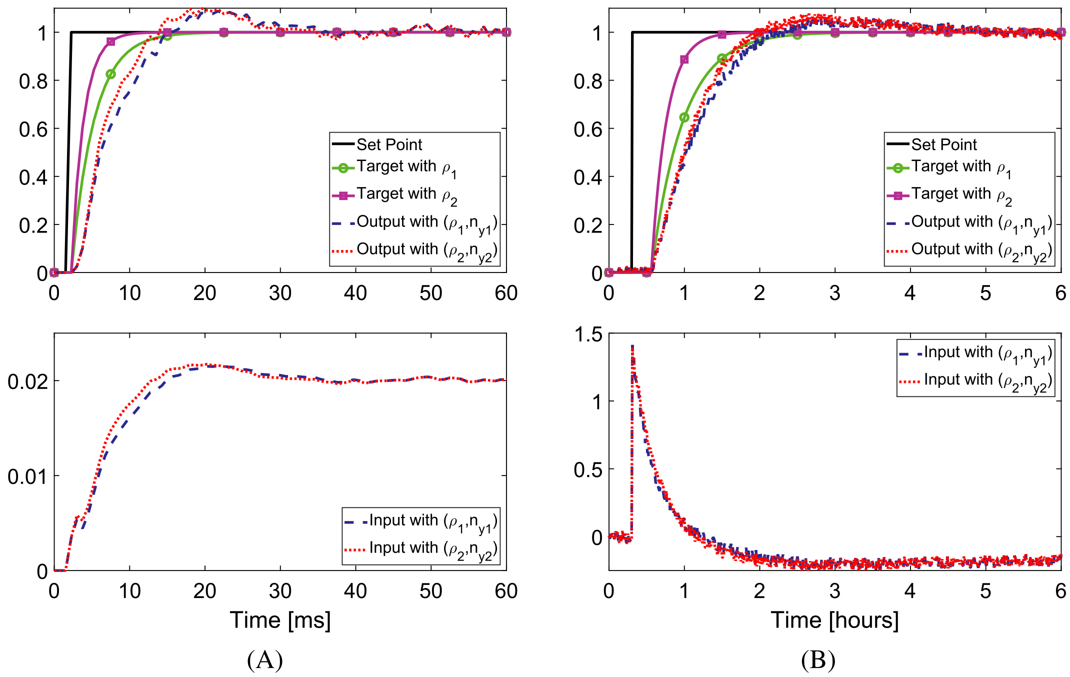


FIGURE 8 Comparison of noise sensitivity (Guassian white measurement noise with $\mu = 0.05$) and modeling mismatches with both tuning choices for (A) G_1 with unmodeled pole at $z = 0.25$, and (B) G_2 with 10% multiplicative uncertainty

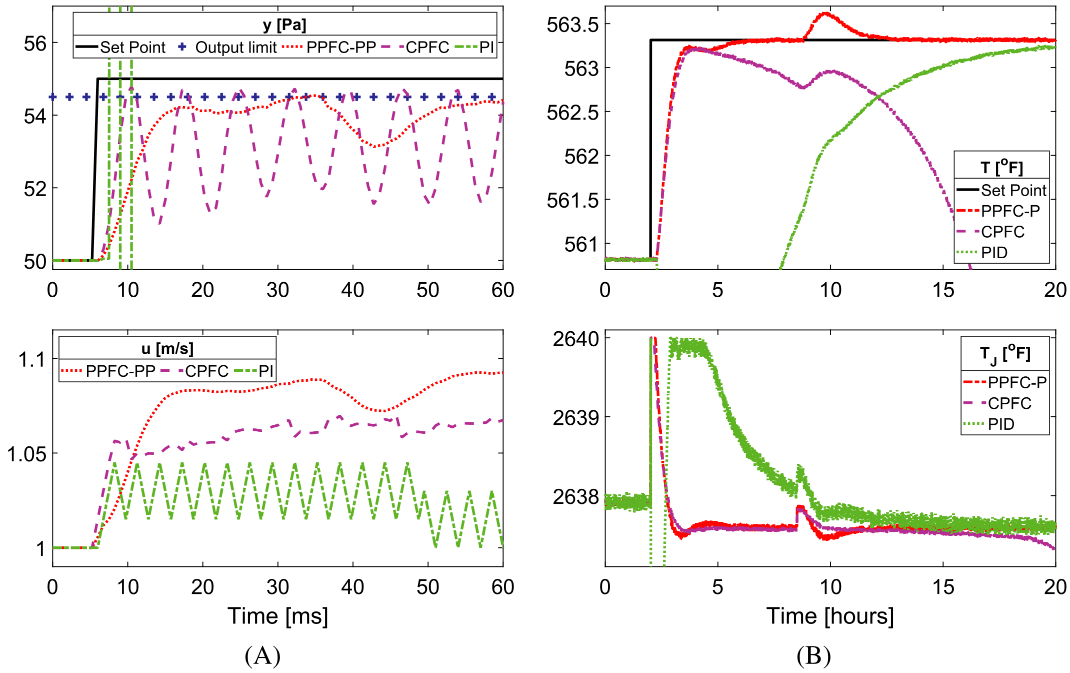


FIGURE 9 Comparison of the constrained closed-loop performance in the presence of external disturbances and measurement noise for (A) the process G_1 subject to $|\Delta u| \leq 0.015$ m/s and $|y| \leq 4.5$ Pa, and (B) the process G_2 subject to $|T_J| \leq 2.1$°F

although does not destabilise, clearly fails to damp down the audible oscillations. On the other hand, the proposed PPFC-PP not only successfully filters out the acoustic signal, but does so by maintaining feasibility despite a large change in both the set point and the disturbance. Notably the output never reaches the new target due to the restriction imposed on the process variable.

For the unstable process, the closed-loop performance is displayed in Figure 9B. A step change of $2.5°F$ drives the process away from the nominal operating point causing large uncertainty, which along side the imposed actuation limit proves too demanding for both the CPFC and the PID. The resulting instability with the standard PFC becomes apparent only after some time owing to the use of numerically infeasible open-loop predictions in the decision-making. The PI controller too fails to accommodate the effect of constraints and uncertainty. In comparison, the proposed algorithm depicts superior performance with highly commendable characteristics despite facing the challenges.

In conclusion, these examples have clearly validated the rationale behind using pre-stabilised predictions in a PFC law for a reliable closed-loop performance.

8 | CONCLUSIONS

A systematic design framework for PFC using prestabilization is presented to overcome the fundamental weaknesses of the standard PFC algorithm with oscillatory and unstable dynamic systems. The proposal employs well-understood classical feedback control mechanisms to modify the difficult open-loop behaviour, thereafter deploying a cascade structure for a reliable PFC implementation, with improvements observed on two main fronts. Firstly, the controller tuning after prestabilization becomes far more consistent and meaningful, with a stronger influence on the closed-loop performance. Secondly, the availability of stable and convergent predictions allows nominal recursive feasibility results under constrained operation, which is generally not the case with difficult open-loop dynamics. An inevitable consequence of prestabilization, however, is a slightly more involved constraint validation process, as reparameterizing the main decision variable renders the simple saturation policy less straightforward to implement.

As for stabilising the open-loop dynamics, two simple and intuitive proposals are discussed. In most cases, the simple proportional plus derivative compensation proves sufficient. This is fairly generic and based on the fact that the majority of real-world processes can be adequately represented as dominant second-order dynamics, for which simple tailored solutions are well understood. Where P(D) alone is insufficient (for instance poorly damped dynamics), pole placement schemes can be quite effective at preconditioning. Two real-world case studies have been used to analyze and validate the closed-loop performance of the PPFC in a variety of practical scenarios. In general, the proposed PPFC operates more efficiently with external disturbances, sensor noise and uncertainties as opposed to the standard PFC and the PID controllers.

Future work will focus more formally on frequency domain robustness studies to gain clearer understanding of the pros and cons of different internal feedback designs. Moreover, extending the scope of validation across a range of industrial case studies and real-time experiments is also under consideration.

ACKNOWLEDGMENTS

The first author would like to acknowledge the University of Sheffield for his PhD scholarship which is funding his studies.

CONFLICT OF INTEREST

The authors declare no potential conflict of interests.

AUTHOR CONTRIBUTIONS

This paper is a collaborative work between both authors. John Anthony Rossiter provided initial proposals and accurate communication of the concepts employed in previous MPC and PFC control laws while reviewing the whole project. Muhammad Saleheen Aftab proposed the framework, developed the code and analyzed the concepts in various challenging case studies.

DATA AVAILABILITY STATEMENT

Data available on request from the authors

ORCID

Muhammad Saleheen Aftab https://orcid.org/0000-0002-1195-7145

REFERENCES

1. Richalet J, Rault A, Testud JL, Papon J. Model predictive heuristic control. Automatica. 1978;14(5):413-428.
2. Skogestad CGS. Should we forget the smith predictor? IFAC-PapersOnLine. 2018;51(4):769-774. doi:10.1016/j.ifacol.2018.06.203
3. Visioli A. Practical PID Control. Springer; 2006.
4. Abdullah M, Rossiter JA. Input shaping predictive functional control for different types of challenging dynamics processes. Processes. 2018;6(8):118.
5. Richalet J, O'Donovan D. Elementary predictive functional control: a tutorial. Proceedings of the 2011 International Symposium on Advanced Control of Industrial Processes (ADCONIP); 2011:306-313.
6. Richalet J, O'Donovan D. Predictive Functional Control: Principles and Industrial Applications. Springer Science & Business Media; 2009.
7. Rossiter JA. A First Course in Predictive Control. CRC Press; 2018.
8. 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(2):229-239. doi:10.2478/v10006-008-0021-z
9. Cutler CR, Ramaker BL. Dynamic matrix control??a computer control algorithm. Joint Automatic Control Conference; Vol. 17, 1980:72. 10.1109/JACC.1980.4232009
10. Clarke DW, Mohtadi C. Properties of generalized predictive control. Automatica. 1989;25(6):859-875. doi:10.1016/0005-1098(89)90053-8
11. Maciejowski JM. Predictive Control: With Constraints. Pearson Education; 2002.
12. Rossite JA, Haber R. The effect of coincidence horizon on predictive functional control. Processes. 2015;3(1):25-45.
13. Rossiter JA. Input shaping for PFC: how and why? J Control Decis. 2016;3(2):105-118.
14. Zabet K, Rossiter JA, Haber R, Abdullah M. Pole-placement predictive functional control for under-damped systems with real numbers algebra. ISA Trans. 2017;71:403-414.
15. Rossiter JA, Haber R, Zabet K. Pole-placement predictive functional control for over-damped systems with real poles. ISA Trans. 2016;61:229-239.
16. Mayne DQ, Rawlings JB, Rao CV, Scokaert POM. Constrained model predictive control: stability and optimality. Automatica. 2000;36(6):789-814.
17. Anthony RJ, Basil K, Rice MJ. A numerically robust state-space approach to stable-predictive control strategies. Automatica. 1998;34(1):65-73.
18. Aftab MS, Rossiter JA, Zhang Z. Predictive functional control for unstable first-order dynamic systems. Proceedings of the CONTROLO 2020; Vol. 695, 2021:12-22; Cham, Springer International Publishing.
19. Abdullah M, Rossiter JA. Alternative method for predictive functional control to handle an integrating process. Proceedings of the 2018 UKACC 12th International Conference on Control (CONTROL); 2018:26-31; IEEE.
20. Zhang Z, Rossiter JA, Xie L, Su H. Predictive functional control for integral systems. Proceedings of the International Symposium on Process System Engineering; 2018.
21. Aftab MS, Rossiter JA. Pre-stabilised predictive functional control for open-loop unstable dynamic systems. IFAC-PapersOnLine. 2021;54(6):147-152. 7th IFAC Conference on Nonlinear Model Predictive Control NMPC; 2021. doi:10.1016/j.ifacol.2021.08.537
22. Rossiter JA, Aftab MS. A comparison of tuning methods for predictive functional control. Processes. 2021;9(7). doi:10.3390/pr9071140
23. Haber R, Rossiter JA, Zabet K. An alternative for PID control: predictive functional control-a tutorial. Proceedings of the 2016 American Control Conference (ACC); 2016:6935-6940; IEEE.
24. Åström KJ, Hägglund T. PID Controllers: Theory, Design, and Tuning. Instrument society of America Research; 1995.
25. Goodwin G. Control System Design. Prentice Hall; 2001.
26. Gilbert EG, Kolmanovsky I, Tan KT. Discrete-time reference governors and the nonlinear control of systems with state and control constraints. Int J Robust Nonlinear Control. 1995;5(5):487-504. doi:10.1002/rnc.4590050508
27. Hu J, Ding B. An efficient offline implementation for output feedback min-max MPC. Int J Robust Nonlinear Control. 2018;29(2):492-506. doi:10.1002/rnc.4401
28. Raković SV, Kouvaritakis B, Cannon M, Panos C. Fully parameterized tube model predictive control. Int J Robust Nonlinear Control. 2012;22(12):1330-1361. doi:10.1002/rnc.2825
29. Annaswamy AM, Hong S. Control of unstable oscillations in flows. In: Levine WS, ed. The Control Handbook. CRC Press; 2011.
30. Bhattacharya C, Mondal S, Ray A, Mukhopadhyay A. Reduced-order modelling of thermoacoustic instabilities in a two-heater Rijke tube. Comb Theory Model. 2020;24(3):530-548. doi:10.1080/13647830.2020.1714080
31. Bequette BW. Behavior of a CSTR with a recirculating jacket heat transfer system. Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301); 2002; IEEE.
32. Rao AS, Chidambaram M. Analytical design of modified Smith predictor in a two-degrees-of-freedom control scheme for second order unstable processes with time delay. ISA Trans. 2008;47(4):407-419. doi:10.1016/j.isatra.2008.06.005
33. MathWorks. PID tuning algorithm; 2021. https://www.mathworks.com/help/control/getstart/pid-tuning-algorithm.html

AUTHOR BIOGRAPHIES

Muhammad Saleheen Aftab. Muhammad Saleheen Aftab completed BE in Electronics Engineering from NEDUET, Pakistan in 2009. He has also completed MSc in Electrical Engineering with specialization in Electronic Instrumentation & Control Systems from SQU, Oman in 2015. Currently he is pursuing PhD at the Dept. of Automatic Control and Systems Engineering, University of Sheffield, UK. His research interests include Model-based Predictive Control, Convex Optimization, and Intelligent Control Systems. He has published numerous research articles in prestigious academic conferences and scientific journals.

John Anthony Rossiter. After studying his first degree in Engineering Science and DPhil (both at Oxford), Dr Rossiter has been an academic at Loughborough (1992–2001) and now at Sheffield. He has always maintained strong interests in both technical research and education. His technical research has predominantly been based around the area of predictive control and more specifically with a focus on insight and concepts, modifying the basic algorithm to optimize computational efficiency and/or simplicity with minimal sacrifice to the expected performance. Within education, his interests are varied and he has also served as chairs of the IFAC and IEEE technical committees on control education. He has played a major role in improving mathematics support for engineers and also tries to enthuse colleagues to consider the potential of new technology for improving the learning experience available to students.

How to cite this article: Aftab MS, Rossiter JA. Predictive functional control for challenging dynamic processes using a simple prestabilization strategy. *Adv Control Appl*. 2022;e102. doi: 10.1002/adc2.102